Breast Cancer Classification - Deep Learning Project

Group 4

Adriana Pinto | 20221921; David Duarte | 20221899; Maria Teresa Silva | 20221821
Marta Alves | 20221890; Miguel Nascimento | 20221876

Introduction

Cancer is one of the most common types of cancer worldwide and, according to the World Health Organization[1], it is the leading cause of cancer related deaths among women. Approximately two out of eight women die when diagnosed with this disease[2]. Breast cancer can be categorized into invasive cancer and in situ cancer according to whether it spreads out or not, and invasive cancer is further divided into four stages (i.e., I, II, III, or IV) based on the spreading severity. Early detection and efficient systemic therapies are essential in reducing the mortality rate of breast cancer. Deep learning has made remarkable progress over the past decade, demonstrating great efficacy in extracting representations from high-dimensional data. Consequently, it has been widely studied for analysing medical images, reshaping cancer research and personalized clinical care. Networks such as AlexNet, VGG, ResNet, DenseNet and Inception have all been successfully applied in breast cancer imaging. Recently, Vision Transformers, a type of deep neural networks that are mostly based on attention mechanisms, have shown great potential in image processing[3].

The goal of this project is to develop a deep learning model to classify breast tissue images for cancer diagnosis. The model will first identify if the cancer is benign or malignant through binary classification and then extend to multiclass classification to predict specific tumor types.

Methodology

For this project, different metrics were used to assess the performance of each problem. In the binary problem, as we are working with imbalanced target classes, the main metric used to assess our models was F1 Score. However, as missing a malignant cancer diagnosis (false negatives) can have severe consequences, it is important to focus
on recall alongside the F1 score to ensure that the model not only predicts correctly benign and
malignant cells but minimizes the risk of overlooking actual cancer cases. For the multiclass problem, the macro F1 score was chosen as the evaluation metric, ensuring that all classes were treated equally. Since Keras does not natively support this metric, it was evaluated on the validation dataset for comparison.

Regarding the optimizer chosen, Adam was used as a baseline for comparing models due to its faster convergence. Later, other optimizers were explored to further improve the models, such as Rmsprop and AdamW. Moreover, Adopt[4] [5] optimizer was considered, but were encountered many issues during its implementation, mainly due to TensorFlow version dependecies incompatibilities, and was not possible to fully try it. The loss function used in the binary problem was *binary_crossentropy* and for the multiclass problem was *categorical_crossentropy*. In terms of transfer learning, DenseNet121[6] [7] was majorly used as it performs better with 224x224 images, as this is the resolution it was originally trained on. Several research papers focusing on cancer imaging applications were reviewed, and the majority utilized DenseNet121[8], highlighting its effectiveness in this domain. ResNet50 was also used for comparison purposes, as was equally referred in some research papers.

Initial Steps

Before proceeding with the detailed steps, it is important to mention the modifications made to the folder structure and corresponding images paths, ensuring there were no empty folders. As part of this process, the image paths were replaced in the *image_data.csv* file. After performing an initial exploration with *.info()* and *.describe()* methods, missing values were identified, and treated by extracting information from the image paths, which contained details for the missing columns. A new column was added to label each image as either 'train' or 'test,' based on a stratified split by cancer type. Next, separate folders were created for the training and testing sets and the images were copied into their respective folders to align with the new structure.

Binary Problem

Maintaining image quality is critical when working with cancer images. A key consideration is how to import those images. As the original resolution of the images are rectangles, should downscaling be performed to squares, accepting some distortion since CNNs tend to perform better with square images? Or should the original aspect ratio be preserved to avoid more distortion?

To address this, the images were preprocessed by transforming them into arrays and dividing by 255 and tested two methods: resizing directly to squares and resizing while maintaining the aspect ratio, then padding[9] the remaining space to form a square.

Experiments were conducted using image dimensions of 224x224 and 128x128, revealing that both approaches performed similarly. Although processing 128x128 images is faster, due to the critical nature of working with cancer images the priority was to achieve the best performance possible. Higher-resolution images provide more detailed features, which are essential for complex models to perform effectively. Based on our findings and prior experience in class, we chose to resize the images to 224x224, ensuring the resolution aligns with the capabilities of advanced models for improved accuracy.

Duplicate images were checked, as this can lead to data leakage and contamination of the validation and test sets with images that were already used to train the models. To address this, the train and test sets were combined, identifying 125 duplicates, which is not significant considering the initial number of images used. As duplicated images were present, a stratified train test split was conducted again, to create new unique stratified sets. The split was stratified based on both binary and multiclass labels to maintain label distribution. Then, a checkpoint was created by saving the arrays, eliminating the need to rerun the previous code. Next, the train data was splitted into train and validation, as keras *validation_split* in the first tests with image resizing and padding led to abnormal high *val_precision*.

The modelling process began by testing different architectures and parameters. Following the approach taught in class, models were designed to represent extremes: one with minimum complexity and another with maximum complexity. The minimal model converged quickly but underperformed, while the maximum model required significantly more time to converge due to its complexity. To balance these extremes, a model with moderate complexity was tested, which achieved better scores but exhibited more fluctuations. So, this model was chosen as our base comparison. After, in hopes of minimizing these fluctuations and improving scores, image transformations were tested. The first transformation was Histogram equalization[10], which improves global contrast by redistributing an image's histogram[11]. The intuition behind this process is that large peaks correspond to images with low contrast where the background and foreground are both dark or both light. Hence histogram equalization stretches the peak across the whole range of values leading to an improvement in the global contrast of an image. For this reason, it's commonly used in medical and satellite imagery although it can produce unrealistic effects. In this case, it didn’t help improve the score nor the fluctuations. Contrast Limited Adaptive Histogram Equalization (CLAHE[12] [13]) was also tested, it enhances image contrast by applying histogram equalization to small regions (tiles) within an image, rather than globally. By limiting the contrast amplification, CLAHE avoids over-enhancement and excessive noise amplification, producing more natural results[14].

Although it got better results than the other image transformation, it still got worse fluctuations problems.

The main problem in the base model still remained, so different poolings were experimented after some guidance from theoretical classes. Batch normalization was also tested to try to normalize our results. The experiments showed that *avg_batch* and *max_batch* have similar results, therefore *max_batch* was chosen. It was observed that the model without batch normalization doesn't tend to overfit in later epochs but tends to have lower scores[15]. Therefore, a more complex model was used to address overfit while maintaining good scores.

The first model achieved scores close to 0.90 in precision and recall, however a slight overfit was detected. By adding Dropout along with Callbacks such as *early_stopping* and *model_checkpoint* with the intention of saving the best model it was possible to solve the problem and achieve promising results. It was observed that good results had been achieved without relying on grid search or extensive data preprocessing. Even so, the goal was to explore how much further it could be improved with minimal changes to preprocessing.

To start, a basic analysis was conducted of our model's performance. The model classified 78(4,9%) False Positives (FP) and 74(4,7%) False Negatives (FN). Curious whether these errors occurred within the same cancer type, the performance was analysed further revealing that while the model struggled with a few samples in each class, it performed consistently across all cancer types[6], with no bias toward any specific class. Therefore, it becomes more challenging to pinpoint areas for improvement.

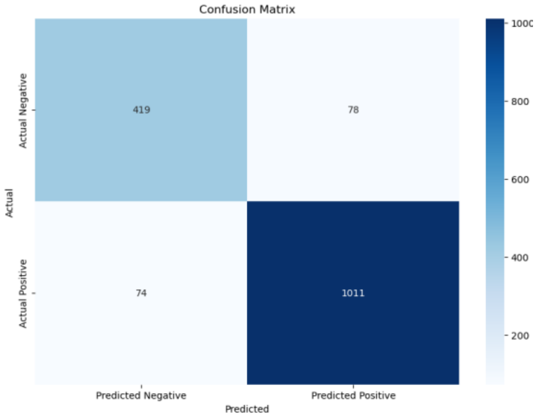


Figure 1 - Confusion Matrix

Attempts were made to visualize correct and incorrect predictions, but without medical expertise, identifying patterns was not possible. As a result, data augmentation and image transformations were based on practical experimentation rather than domain knowledge. Given the strong baseline and limited resources, the focus was placed on simple augmentation techniques while minimizing distortion.

To balance the dataset, new images were generated using data augmentation (ImageDataGenerator), while minimizing distortion[16]. Knowing that distortion is a critical concern for medical images, hyperparameters were carefully adjusted to ensure the augmented images introduced only subtle changes, maintaining minimal distortion. Unfortunately, the results did not improve. Additionally, Synthetic Minority Over-sampling Technique (SMOTE[17]), a data augmentation technique used to address class imbalance in datasets, was used. It works by generating synthetic samples for the minority class rather than simply duplicating existing data. This is done by interpolating between existing minority class samples to create new, realistic data points. When applied to raw image data, SMOTE works in a different manner compared to ImageDataGenerator. While ImageDataGenerator creates augmented images by applying transformations like rotations, flips or scaling, SMOTE generates entirely new images by interpolating pixel values between existing minority samples[1]. However, this approach failed to improve the scores.

To finalize the work on the binary classification problem, the DenseNet121 model, commonly used in medical image classification tasks, particularly for datasets like BreakHis, was fine-tuned. Because the dataset used in this project differs significantly from ImageNet, it was decided to unfreeze the last 12 layers to allow the model to learn patterns specific to the data, adapting the pre-trained network to better fit the medical images in use. The model obtained similar results to the best model until now, however, the other model was preferred due to its faster training time.

Multiclass Problem

As in the previous problem, the same image resizing approaches were evaluated, and it was concluded that, similar to binary classification, it yielded identical results. Consequently, a simple 224x224 resize was initially adopted to prioritize performance, models' precision and compatibility to transfer learners.

Next, the Best Binary Model, a Lighter Architecture and a Extreme Maximum Architecture Models were tested for this problem. The Lighter model demonstrated less overfitting, with recall and precision exhibiting low overfit despite significant overfitting in loss. Efforts to reduce overfitting in both the Best Binary and Lighter models included techniques such as batch normalization, dropout, L2 regularization, increased batch size, and the AdamW optimizer. Unfortunately, these methods did not improve the Best Binary model. In contrast, the Lighter model, which was adopted as the base model, showed a reduction in overfitting with the use of the AdamW optimizer and increased batch size, though it still experienced significant overfitting in loss.

Unsatisfied with the scores, alternative approaches were explored. First, incorporating class weights allowed assigning different importance to each class in the loss function. This helped the model focus more on underrepresented classes, potentially improving overall performance, particularly for the minority classes. This resulted in poorer macro F1 scores. Then, the image enhancement techniques previously tested on the binary problem were also applied but led to even worse results in both macro F1 and weighted F1 scores.

Data augmentation was then applied. However, due to computational constraints, it was impossible to balance all classes with 1811 samples each. Instead, the majority class was slightly downsampled, and data augmentation was applied to the other classes using SMOTE, resulting in a balanced dataset with 600 images per class. Unfortunately, the scores didn't improve. Additionally, with a resolution of 128x128, SMOTE could be performed without downsampling the majority class. While this approach significantly increased the number of generated images, it led to an improvement in the macro average F1 score.

Next, Multi Input Model was tested. While exploring this approach, determining an optimal architecture proved to be challenging, as the objective was to showcase multi model potential. First, the Lighter and Best Binary models architectures were tested. The latter was selected due to its lower training loss, indicating that more complex models might achieve better results. To further enhance its performance, a basic grid search was conducted. Although the grid search tested only four hyperparameter configurations, one promising configuration was achieved, resulting in reduced loss and improved model performance.

After this discovery, the question of whether adding the image magnification could further improve results was raised. To incorporate magnification information, the preprocessing was redone. However, the results were similar to those obtained in the first approach.

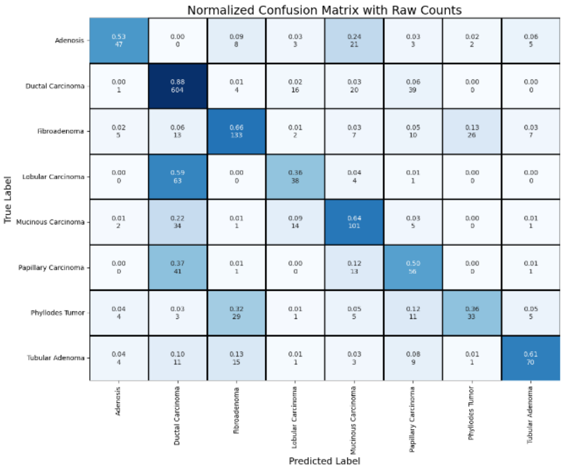
Up to this point, various methods have been tested, with the most promising being the first approach of Multi Input modelling. While the scores were satisfactory, further improvement was desired. Therefore, transfer learning was explored to determine if it can enhance the results. Without extensive experimentation, unfreezing layers already resulted in improved performance with only the image input. This finding led to considering the use of DenseNet121 in the Multi Input model for potentially even better results.

The model showed an increase in F1 macro, though it also showed a tendency to overpredict Ductal Carcinoma. This is evident in its misclassification of Lobular Carcinoma, Mucinous Carcinoma, and Papillary Carcinoma as Ductal Carcinoma[18].

This indicated the need to improve differentiation among these four classes. The issue likely arises from the limited number of samples in the minority classes. Addressing this could involve upsampling the underrepresented classes or applying image transformations to enhance distinguishing features.

It was also decided to test ResNet50, which produced the worst results, likely due to low number of epochs. Since training ResNet50 took approximately 11 minutes per epoch, further tuning was deprioritized. DenseNet121, with the same architecture, achieved better results and was more efficient in this context. Later, to improve performance, different hyperparameters for learning rate, unfrozen layers, and optimizers were tested using Hyperband tuning. Out of the 8 model combinations tested, the one with the best validation loss matched the previously evaluated configuration. While adding more hyperparameters to the search could lead to better results, the computational cost was significant, as 10 trials required approximately 400 minutes. Given these resource constraints, further tuning was limited to this configuration.

Finally, the multi input model with DenseNet121 was adapted to receive as input just the images, using our final binary model to predict the label Benign/Malignant and returning as output the multiclass prediction. This led to a reduction in the performance of the model but knowing that it has an extra error rate due to the missing prediction of the binary model, its results were satisfactory. Below it can be seen the confusion matrix for this model, that reflects the same overprediction of the previous version of this model.

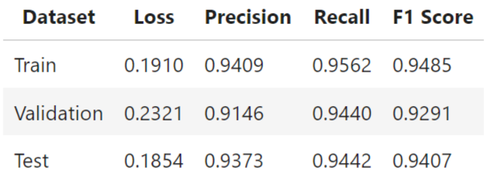


	Adenosis	Ductal Carcinoma	Fibroadenoma	Lobular Carcinoma	Mucinous Carcinoma	Papillary Carcinoma	Phyllodes Tumor	Tubular Adenoma
Adenosis	0.53 47	0.00 0	0.09 8	0.03 3	0.24 21	0.03 3	0.02 2	0.06 5
Ductal Carcinoma	0.00 1	0.88 604	0.01 4	0.02 16	0.03 20	0.06 39	0.00 0	0.00 0
Fibroadenoma	0.02 5	0.06 13	0.66 133	0.01 2	0.03 7	0.05 10	0.13 26	0.03 7
Lobular Carcinoma	0.00 0	0.59 63	0.00 0	0.36 38	0.04 4	0.01 1	0.00 0	0.00 0
Mucinous Carcinoma	0.01 2	0.22 34	0.01 1	0.09 14	0.64 101	0.03 5	0.00 0	0.01 1
Papillary Carcinoma	0.00 0	0.37 41	0.01 1	0.00 0	0.12 13	0.50 56	0.00 0	0.01 1
Phyllodes Tumor	0.04 4	0.03 3	0.32 29	0.01 1	0.05 5	0.12 11	0.36 33	0.05 5
Tubular Adenoma	0.04 4	0.10 11	0.13 15	0.01 1	0.03 3	0.08 9	0.01 1	0.61 70

Figure 2 - Confusion Matrix

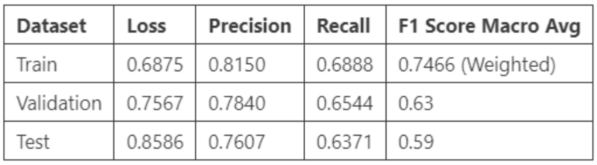
Results

As a result, the best model achieved an F1 score of 0.94 on the test set, for the binary classification[19], and 0.59 of macro F1 score for the multiclass classification[20].



Dataset	Loss	Precision	Recall	F1 Score
Train	0.1910	0.9409	0.9562	0.9485
Validation	0.2321	0.9146	0.9440	0.9291
Test	0.1854	0.9373	0.9442	0.9407

Figure 3 – Binary Classification Final Scores



Dataset	Loss	Precision	Recall	F1 Score Macro Avg
Train	0.6875	0.8150	0.6888	0.7466 (Weighted)
Validation	0.7567	0.7840	0.6544	0.63
Test	0.8586	0.7607	0.6371	0.59

Figure 4 – Multiclass Classification Final Scores

Future Work

Due to time constraints, several potential improvements were not explored. In the binary classification task, hyperparameter tuning could have been conducted to potentially achieve better results. For multiclass classification, using the data augmentation approach with reduced image size on the best-performing model was identified as a promising direction, given its superior macro F1 score compared to the normal dataset. Additionally, applying Hyperband tuning with a broader range of parameters and experimenting with dynamic class weights, based on model performance or adjusted for classes misclassified as the majority class, were other processes that could have been pursued. Incorporating class weights to improve the performance of classes with the lowest scores in multiclass classification was another potential approach. Furthermore, an error encountered during the grid search process should be addressed.

Conclusion

We believe that this problem allowed us to gain a lot of practical experience on developing a deep learning model, while learning how to work with images as data. Moreover, we could conclude that in simpler problems, where a small architecture can yield good results (such as in binary classification), it may be worth building our own model. However, for more complex problems, using the right transfer learner can provide better results more easily, allowing us to focus on experimenting with data augmentation and other techniques rather than spending time developing architectures that may not deliver the desired performance.

Annexes

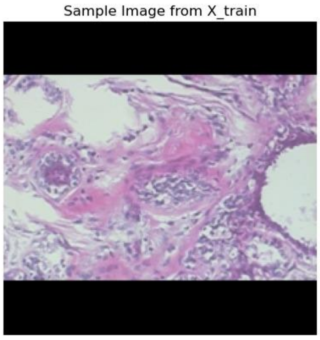


Figure 5 – Padding Image

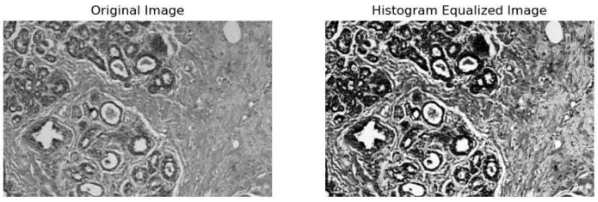


Figure 6 – Histogram Equalized Image

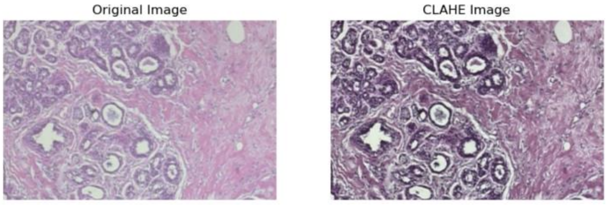


Figure 7 – CLAHE Image

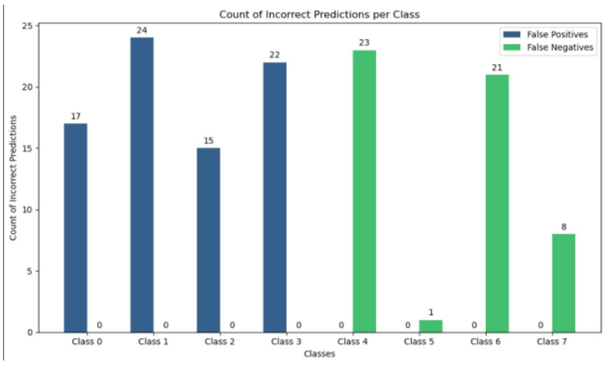


Figure 8 – Count of Incorrect Prediction per Class

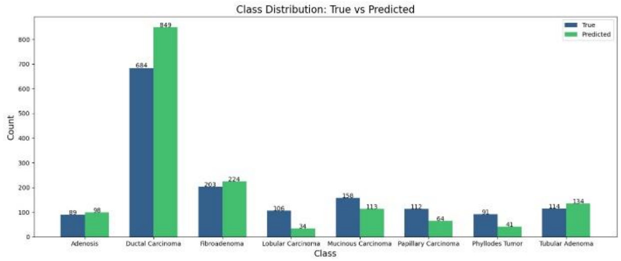


Figure 9 – Count of Incorrect Prediction per Class

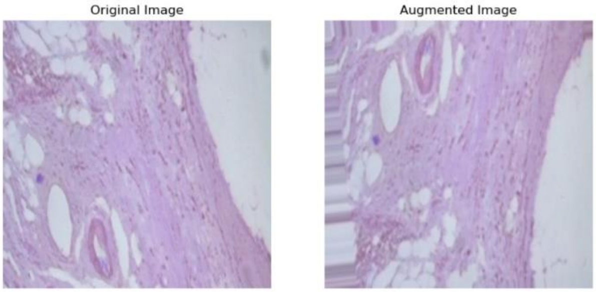


Figure 10 – Image Data Generator

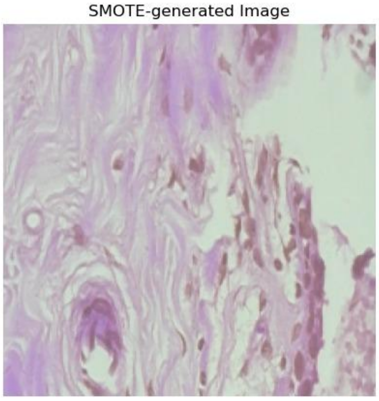


Figure 11 – SMOTE Image

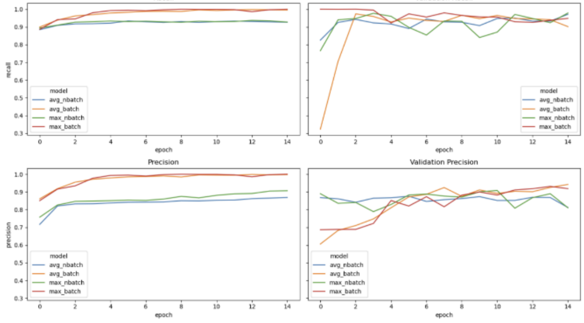


Figure 12 – Binary Scores

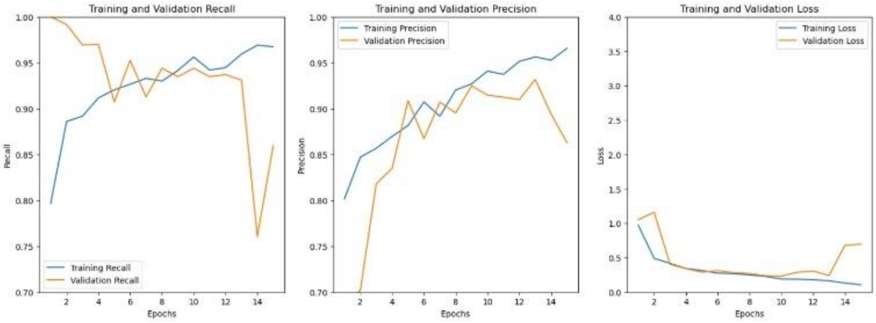


Figure 13 – Final Binary Model

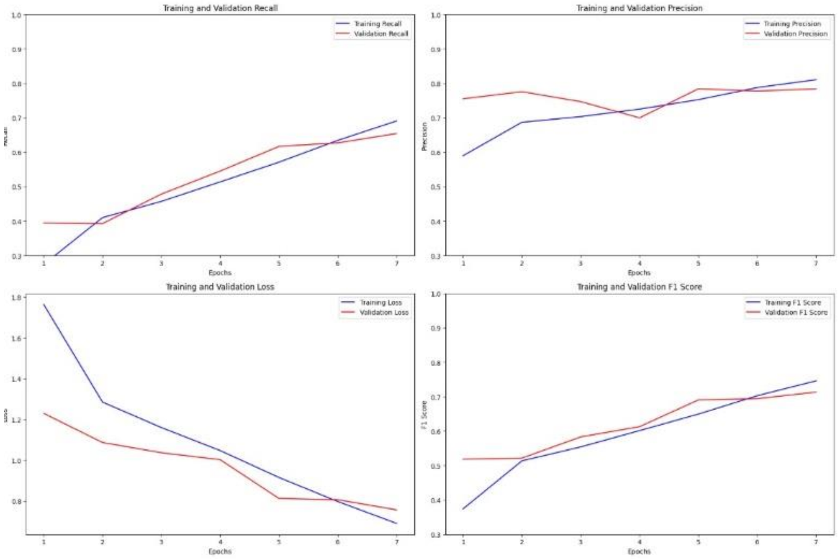


Figure 14 – Final Multiclass Model

References:

1. World Health Organization, "Breast cancer inequities," WHO, [Online]. Available: https://www.who.int/initiatives/global-breast-cancer-initiative/breast-cancer-inequities.
2. L. Luo *et al.*, "Deep Learning in Breast Cancer Imaging: A Decade of Progress and Future Directions," in *IEEE Reviews in Biomedical Engineering*, doi: 10.1109/RBME.2024.3357877.
3. A. Sharma and S. Mittal, "Deep Learning Approach –Improved CNN Model for the Breast Cancer Classification," *2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC)*, Gwalior, India, 2024, pp. 847-853, doi: 10.1109/AIC61668.2024.10730870.
4. S. Taniguchi, " *Adopt: Modified Adam Can Converge with Any β_2 with the Optimal Rate*", 2024. [Online]. Available: https://arxiv.org/abs/2411.02853.
5. iShohei220, "*Adopt: Modified Adam Can Converge with Any β_2 with the Optimal Rate*" GitHub repository. [Online]. Available: https://github.com/iShohei220/adopt.
6. M. Y. Adam, M. M. Saeed, and A. S. A. Ahmed, "Medical image enhancement application using histogram equalization in computational libraries," *International Journal of Computer Science and Technology*, vol. 6, no. 1, pp. 7–12, Jan.–Mar. 2015. [Online]. Available: https://www.ijcst.org/Volume6/Issue1/p2_6_1.pdf.
7. V. Prasad, "Enhancement of image using histogram manipulation," GitHub repository, 2024. [Online]. Available: https://github.com/vishallprasad/Enhancement-of-Image-using-Histogram-Manipulation.
8. SoumyadeepB, "*SMOTE for Imbalanced Datasets*," GitHub repository, 2024. [Online]. Available: https://github.com/SoumyadeepB/MachineLearning/blob/master/Model%20Improvement%20Techniques/SMOTE%20for%20Imbalanced%20Datasets.ipynb.
9. E. TheLearner, "CLAHE implementation using OpenCV," GitHub repository, 2024. [Online]. Available: https://github.com/EshbanTheLearner/CLAHE-openCV.
10. A. Mekkad, "Image enhancement using CLAHE: Image restoration implementation," GitHub repository, 2024. [Online]. Available: https://github.com/ArunMekkad/ImageEnhancementCLAHE/blob/main/image_restoration.py.
11. V. Patel, V. Chaurasia, R. Mahadeva and S. P. Patole, "GARL-Net: Graph Based Adaptive Regularized Learning Deep Network for Breast Cancer Classification," in *IEEE Access*, vol. 11, pp. 9095-9112, 2023, doi: 10.1109/ACCESS.2023.3239671.
12. D. Upadhyay, M. Manwal, V. Kukreja and R. Sharma, "Deep Learning-based VGGNet, GoogleNet, and DenseNet121 Models for Cervical Cancer Prediction," *2024 5th International Conference for Emerging Technology (INCET)*, Belgaum, India, 2024, pp. 1-6, doi: 10.1109/INCET61516.2024.10593267.
13. "Life | Vol. 13, Issue 9, Article 1945," MDPI. [Online]. Available: https://www.mdpi.com/2075-1729/13/9/1945.
14. Keras Documentation - https://keras.io/api/
15. Scikit-learn Documentation - https://scikit-learn.org/stable/api/index.html