Image Classification

Breast Cancer Diagnosis

Carolina Almeida (20221855), Duarte Carvalho (20221900), Francisco Gomes (20221810), Margarida Henriques (20221952), Marta Monteiro (20221954)

1. Introduction

Breast cancer is one of the most common and life-threatening cancers worldwide among women, being, currently, the leading cause of cancer death in female patients [1]. Female gender is the strongest breast cancer risk factor. Approximately 99% of breast cancers occur in women and 0.5% to 1% of breast cancers occur in men. According to the World Health Organization, in 2022, there were 2.3 million women diagnosed with breast cancer and 670 000 deaths worldwide. Breast cancer occurs in every country of the world in women at any age after puberty, but at increasing rates later in life [2].

From 2020 to 2040, the Global Breast Cancer Initiative (GBCI) aims to prevent 2.5 million preventable deaths attributed to breast cancer on a global scale. In women under the age of 70, this would result in a 25% reduction in breast cancer mortality by 2030 and a 40% reduction by 2040 [3]. Accurate detection of the disease in its initial stage is crucial for successful treatment and disease management. Deep Learning techniques have shown promising results in breast cancer diagnosis and have the potential to improve the accuracy and efficiency of diagnosis, leading to earlier detection and better patient outcomes.

Convolutional Neural Networks (CNNs) are one of the most widely used deep learning techniques for medical image analysis. They can extract complex features at different levels of abstraction, making them suitable for detecting complex patterns in medical images. CNNs have been used for tasks such as breast mass classification, tumor segmentation, and breast density classification.

The goal of this project is to develop a high-performance model for breast cancer diagnosis by initially training a binary classification model to differentiate between benign and malignant images. Subsequently, the model will then be extended to a multi-class classification framework, in order to predict specific tumor types, in 8 different categories, based on image analysis.

2. Methodology

This project is structured into two main stages. In Stage 1, a **Binary Classification Model** is developed and trained to differentiate between benign and malignant breast cancer images. In Stage 2, a **Multi-Class Classification Model** is created to classify images into multiple tumor types. In both stages, the model is evaluated using original and preprocessed images to assess whether preprocessing enhances its performance.

Before developing the actual models, we identified four rows with missing values, which were manually imputed. Additionally, we discovered 131 groups of duplicate images using perceptual hashing. To prevent future data leakage, we retained only one image from each group. We also created several visualizations, which helped us identify the class imbalance in both the binary and multi-class stages of the project. Furthermore, the analysis revealed that benign and malignant cancers were assigned distinct cancer types.

2.1. Binary Classification Model

Data Preparation

The process of building the binary classification model began with careful preparation of the dataset. The first step involved converting the "Benign" and "Malignant" labels into numerical binary values using the `LabelEncoder` from the Scikit-learn library, making the data compatible with the model. Next, the dataset was split into training and testing sets using the `train_test_split` function from Scikit-learn, ensuring that the class distribution remained consistent across both sets, with 20% of the data allocated to the testing set. The corresponding image paths were then loaded into arrays, ready for training.

To help the model generalize better in real-world scenarios, the training data was augmented. Using the `ImageDataGenerator` from the Keras library, the training images were rotated, zoomed, flipped, and shifted to simulate real-world variability, while the test data was normalized through rescaling. This approach exposed the model to a wider variety of data, helping to reduce the risk of overfitting.

One significant challenge encountered was class imbalance. To address this, the training dataset was balanced to ensure equal representation of both classes. The `RandomOverSampler` from the imbalanced-learn library was used to increase the number of samples from the underrepresented class, thus balancing the dataset.

To further optimize the training process, two mechanisms were employed. Early Stopping was implemented using the Keras library, which monitored the validation loss and stopped training if no improvement was observed after five epochs, thus preserving the best model weights. Additionally, a learning rate scheduler from Keras was used to adjust the learning rate, halving it when progress slowed for three consecutive epochs, ensuring efficient convergence.

After preparing the data, various models were explored for the binary classification task using both the original and preprocessed images. For the original images, custom CNN, VGG16, and ResNet50 models were trained, each offering different approaches to feature extraction and classification. For the preprocessed images, custom CNNs were trained for each preprocessing technique applied, including grayscale conversion, contrast adjustment, and brightness-contrast adjustment, to improve the model's performance by simulating real-world variations in image quality.

Models for Original Images

Custom CNN We built the binary classification model using Kera's `Hyperband Tuner` to find the best hyperparameters. After tuning, we trained the model in three phases. First, on the original dataset for 50 epochs, monitoring key metrics like recall, precision, and F1 score. In the second phase, data augmentation enhanced generalization by introducing variability. In the final phase, class imbalance was addressed with oversampling to ensure balanced learning.
After each phase, we analyzed and visualized the model's performance, comparing metrics to assess improvements.

VGG16 We then built the model using the `VGG16` architecture. By excluding its top layers and adjusting the input size, we adapted it to our task. The model was trained in three phases: first, on the original dataset for 50 epochs, using callbacks for efficient training. Next, data augmentation was applied to improve generalization, and finally, class imbalance was addressed through oversampling.
After each phase, we evaluated the model's performance using F1 scores and visualized the results to monitor progress.

ResNet 50 Model Later, we used the `ResNet 50` architecture for the binary classification model. After loading the ResNet50 model with `imagenet` weights, we trained it in three phases. First, we trained on the original dataset for 50 epochs, with callbacks to prevent overfitting. Next, data augmentation was applied to diversify the training and validation sets. In the final phase, we oversampled the data to address class imbalance. After each phase, loss and F1 scores were evaluated and visualized to track model performance.

Models for Preprocessed Images

For preprocessed images [4], a custom CNN model was trained for grayscale images, while the other two preprocessing techniques used the model obtained from:

Grayscale Conversion We created a binary classification model using a CNN for grayscale images. The architecture included convolutional layers with increasing filter sizes, followed by max pooling, flattening, and fully connected layers, with dropout for regularization. After preprocessing the images to grayscale, we trained the model for 50 epochs with early stopping and learning rate reduction callbacks. The performance was tracked using recall, precision, and F1 score, which were visualized to monitor progress.

Contrast Adjustment We subsequently applied contrast preprocessing to the training images to enhance features. After visualizing the effect of the contrast adjustment, the images were processed and prepared for training. We then trained the binary classification model for 50 epochs, using early stopping and learning rate reduction to optimize performance. During training, we evaluated the model using recall, precision, and F1 score, with results plotted to visualize how the model improved throughout the process.

Brightness-Contrast Adjustment We applied both brightness and contrast preprocessing to the training images to further enhance their features. The adjusted images were then used to train the binary classification model for 50 epochs, with early stopping and learning rate reduction callbacks to optimize training. Throughout this process, the model's performance was evaluated using recall, precision, and F1 score. The results were visualized through plots to track improvements over time, ensuring that the model was learning effectively from the processed images.

Results

The model trained on preprocessed images did not perform well, and models trained with oversampled data resulted in overfitting. Ultimately, the best approach was to use the tuned model with the original images using the hold-out method.

The classification report shows good performance for Malignant cases (class 1), with precision of 0.86, recall of 0.94, and an F1-score of 0.90. For Benign cases (class 0), precision is 0.83, recall is 0.67, and F1-score is 0.74, indicating the model struggles to identify some Benign instances.
Overall accuracy is 0.85, with stronger performance on Malignant cases, likely due to the class imbalance. The confusion matrix confirms the model is better at predicting Malignant cases, while Benign cases experience more misclassifications.

2.2. Multi-Class Classification Model

The second phase of the project focuses on the development of a multi-class classification model, designed to classify breast cancer images into one of eight tumor types. This model builds on the binary classification model from Stage 1, extending its functionality to handle more complex multi-class tasks.

The methodology in this phase closely follows the steps from Stage 1, including data preparation, label encoding (in this case, one hot encoding was applied, converting the labels into vectors of 8 possible categories), and splitting the dataset into training and testing sets. One challenge addressed was class imbalance, as some tumor types, such as Adenosis, had fewer images compared to others, like Ductal Carcinoma. Additionally, model development followed the same approach as in Stage 1, using both original and preprocessed images for training.

The primary difference in this stage was the use of the `Functional API` in Keras for handling the original images. This approach allowed the construction of a more flexible model architecture, capable of processing multiple inputs and outputs simultaneously.

The Functional API enables the creation of more complex models by combining different inputs. For this project, the model was designed with two inputs: image data and a binary feature. The binary input helps the model distinguish between benign and malignant tumors, while the image data is processed to extract detailed features. These inputs are merged to create a unified representation, enabling the model to classify tumors into eight distinct categories.

Results

Once again, all models trained with oversampled data resulted in overfitting. At this stage, both pre-trained models showed poor performance, and contrary to our expectations, the functional API model performed the worst, with a very high loss. As a result, the best-performing model remained the one that was tuned with the original images using the hold-out method.

For some classes, like Ductal Carcinoma, the model performs well, with a precision of 0.59, recall of 0.93, and an F1-score of 0.72. This is likely due to Ductal Carcinoma being the most frequent class in the dataset, allowing the model to learn more effectively from a larger number of samples. The high recall indicates the model is good at identifying most instances of this common cancer type.

However, the model struggles with rarer classes, such as Phyllodes Tumor, where precision, recall, and F1-score are all 0.00. This reflects the model's inability to identify instances of this cancer type, likely due to the limited data available during training. Phyllodes Tumor ranks second-to-last in terms of the number of observations, making it difficult for the model to learn distinguishing features. Similarly, Mucinous Carcinoma and Papillary Carcinoma also show low recall values, indicating the model misses many instances of these rarer cancers.

The macro average F1-score of 0.27 and the weighted average F1-score of 0.45 highlight the challenges posed by class imbalance.

3. Conclusion

Overall, we achieved good results in the first phase of the project, where we focused on binary classification, but the performance in the second phase, which involved multi-class classification, was not as strong. Our approach for both stages struggled to deal with the imbalance, achieving satisfactory performance for the majority class, but poor results for the minority class, which is a common issue in real-world machine learning applications.

In the first phase, we invested a significant amount of time optimizing the model, which unfortunately left us with limited time to fine-tune the multi-class model in the second phase. Despite these time constraints, we successfully achieved all project objectives, meeting the core requirements of both phases.

If we had more time, we would have pursued an approach we believe would yield better results: combining the pre-trained VGG16 and ResNet 50 models [5]. We believe this would have leveraged the strengths of both architectures to improve classification performance. Additionally, we would have trained the models with larger image sizes than (50, 50), as we recognize that using higher resolutions could have led to better feature extraction and, consequently, improved model performance.

In summary, while the results in phase two did not meet our expectations, the project allowed us to explore important aspects of deep learning, such as model tuning, data preprocessing, and handling class imbalance. With more time, we would have further refined our approach, but we are confident in the methodologies used and the progress made toward the project's objectives.

References

[1] Liga Portuguesa Contra o Cancro, "Cancro da mama," 2024. [Online]. Available: https://www.ligacontracancro.pt/cancro-da-mama/

[2] World Health Organization, "Breast cancer," 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/breast-cancer

[3] J. Abdollahi, N. Davari, Y. Panahi, and M. e. a. Gardaneh, "Detection of metastatic breast cancer from whole-slide pathology images using an ensemble deep-learning method," *Archives of Breast Cancer*, 2023. [Online]. Available: https://archbreastcancer.com/index.php/abc/article/view/545

[4] OpenCV, "Opencv documentation," 2024. [Online]. Available: https://docs.opencv.org/4.x/

[5] A. Rugina, A. Negru, F. Andrei *et al.*, "Cancers," *PMC*, 2024. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11201924/#sec1-cancers-16-02222

All other technical approaches implemented, including model architecture, tuning, data augmentation, and other strategies, were derived from the notebooks provided in the practical classes.

Annexes

Figure 1: Training and Validation Loss and F1 Score for the Binary Classification Model

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 48, 48, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 32)	0
conv2d_4 (Conv2D)	(None, 20, 20, 256)	205,056
conv2d_5 (Conv2D)	(None, 18, 18, 128)	295,040
flatten_1 (Flatten)	(None, 41472)	0
dense_3 (Dense)	(None, 512)	21,234,176
dropout_1 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 64)	32,832
dense_5 (Dense)	(None, 1)	65

Total params: 21,768,067 (83.04 MB)

Trainable params: 21,768,065 (83.04 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 2 (12.00 B)

Figure 2: Summary of the Binary Classification Model Architecture

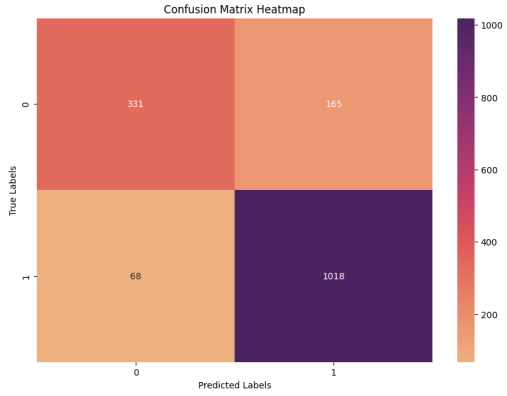

Figure 3: Heatmap of the Binary Classification Model's Performance

	precision	recall	f1-score	support
0	0.83	0.67	0.74	496
1	0.86	0.94	0.90	1086
accuracy			0.85	1582
macro avg	0.85	0.80	0.82	1582
weighted avg	0.85	0.85	0.85	1582

Figure 4: Performance Metrics of the Binary Classification Model

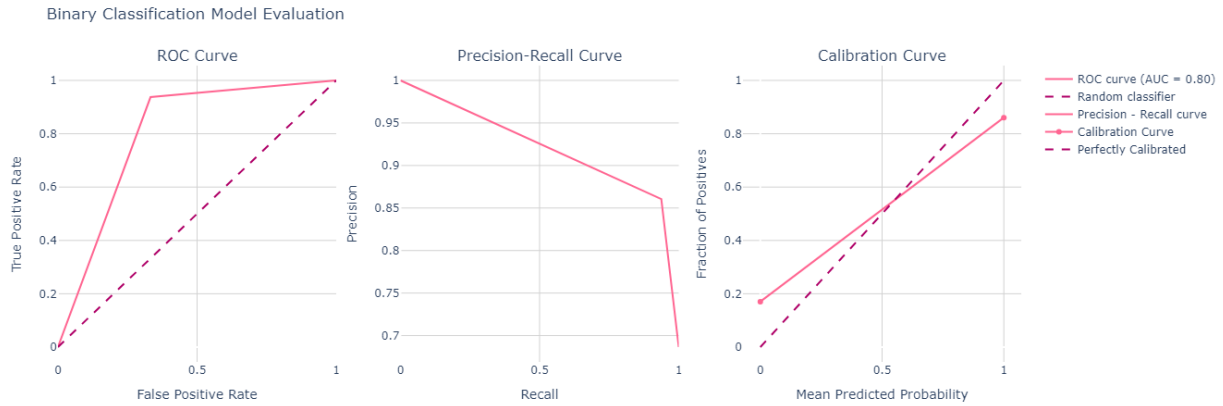

Figure 5: Performance Metrics for the Binary Classification Model – ROC Curve, Precision-Recall Curve, and Calibration Curve

Figure 6: Training and Validation Loss and F1 Score for the Multi-Class Classification Model

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 48, 48, 128)	3,584
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 128)	0
conv2d_4 (Conv2D)	(None, 20, 20, 128)	409,728
conv2d_5 (Conv2D)	(None, 18, 18, 160)	184,480
flatten_1 (Flatten)	(None, 51840)	0
dense_3 (Dense)	(None, 128)	6,635,648
dropout_1 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 112)	14,448
dense_5 (Dense)	(None, 8)	904

Total params: 7,248,794 (27.65 MB)

Trainable params: 7,248,792 (27.65 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 2 (12.00 B)

Figure 7: Summary of the Multi-Class Classification Model Architecture

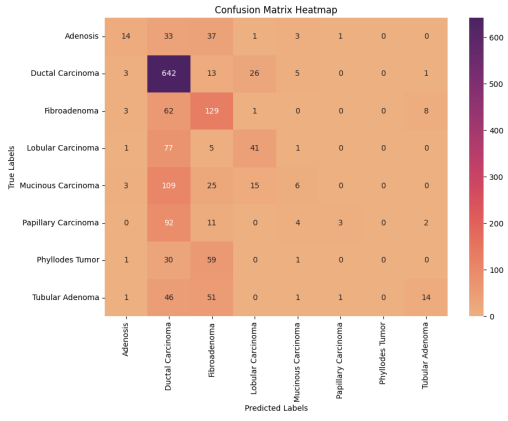

Figure 8: Heatmap of the Multi-Class Classification Model's Performance

	precision	recall	f1-score	support
Adenosis	0.54	0.16	0.24	89
Ductal Carcinoma	0.59	0.93	0.72	690
Fibroadenoma	0.39	0.64	0.48	203
Lobular Carcinoma	0.49	0.33	0.39	125
Mucinous Carcinoma	0.29	0.04	0.07	158
Papillary Carcinoma	0.60	0.03	0.05	112
Phyllodes Tumor	0.00	0.00	0.00	91
Tubular Adenoma	0.56	0.12	0.20	114
accuracy			0.54	1582
macro avg	0.43	0.28	0.27	1582
weighted avg	0.49	0.54	0.45	1582

Figure 9: Performance Metrics of the Multi-Class Classification Model