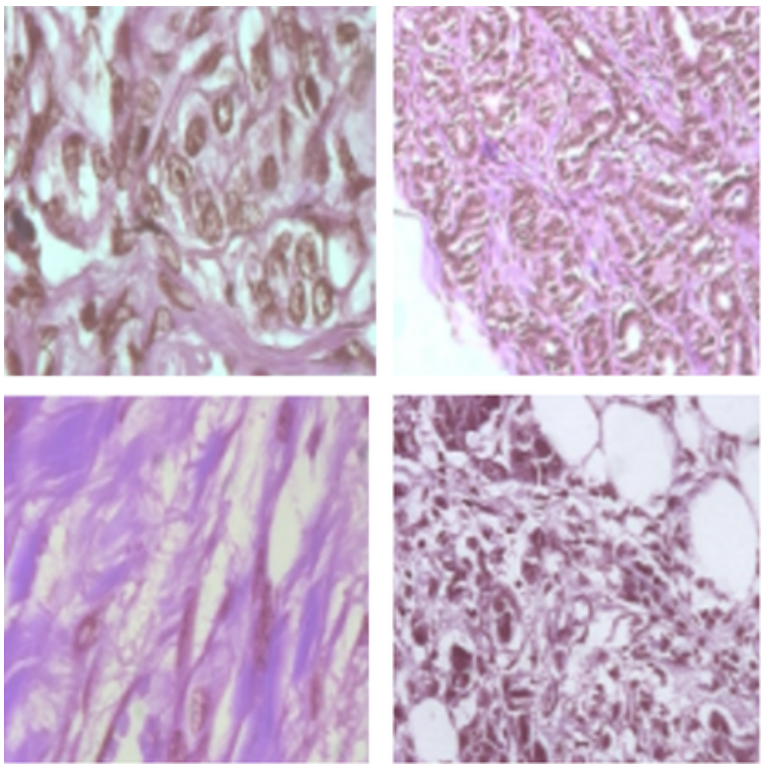
Bachelor's Degree in Data Science 24/25

Deep Learning Project

Breast Cancer Prediction Models



Group 2:
Inês Mendes – 20211624
Miguel Matos – 20221925
Patrícia Bezerra – 20221907
Rita Silva – 20221920
Vasco Capão – 20221906

December, 2024

Abstract

Early breast cancer diagnosis is fundamental to ensure a timely and effective treatment. This project uses deep-learning techniques to classify high-resolution microscopic images of breast tissue from the BreakHis dataset. The present report is structured into two different stages: binary classification, where the model implemented distinguishes between benign and malignant tumours, and multi-class classification, where the model can identify specific tumour types.

During the development of this project, various techniques for pre-processing, transfer learning with pre-trained models, transformations and hyperparameter optimization were used in order to improve the models' performances. Despite difficulties such as class imbalance, the binary classification model achieved an F1 score of 0.91 for malignant cases, with good recall values to reduce false negatives. For multiclass classification, the HyperBand optimized model achieved an overall accuracy of 61% and a weighted average F1 score of 58%, indicating potential but emphasising the need for further improvement.

1 Introduction

To improve treatment outcomes and patient survival rates it is essential an early diagnosis of breast cancer. With advancements in medical imaging, high-resolution microscopic images have become an important tool for detecting the existence and kind of breast tumours. However, pathologists must manually analyse these photos, which can be time-consuming and unpredictable.

This project will employ deep learning approaches to create and test models for classifying breast tissue images from the BreakHis dataset. The work was divided into two stages: a binary classification task to distinguish between benign and malignant tumours, and a multi-class classification problem to determine the precise tumour type. The research aims to address difficulties such as class imbalance and computational restrictions through the use of deep learning techniques such as transfer learning, hyperparameter tuning, and performance optimization.

This work investigates the potential of deep learning in medical imaging, emphasising its strengths and limitations. The report also summarizes techniques, findings, and future developments, illustrating the promise of AI to improve breast cancer diagnosis.

To accomplish our objective, we created three notebooks:

1. *01_df_cleaningi* - notebook focusing on data preparation and addressing the missing values.
2. *02_Beningn_Malign* - notebook to complete the binary classification problem.
3. *03_CancerType* - notebook to solve the multi-class classification task.

Additionally, we created a *.py* file named *functions*, which includes all imports and functions used in the three notebooks for better organization.

2 Experimental Setup

In this section, we list the key Python libraries employed in our project and their use:

- os: Used to manage directories and file paths.
- numpy: Used for creating arrays to store image data.
- pandas: Used to load, manipulate, and analyze structured data.
- matplotlib.pyplot: Used to create visualizations such as loss/accuracy curves.
- cv2: Used for image processing tasks, such as resizing and normalizing.

sklearn.metrics:

- confusion_matrix: Computes the confusion matrix for detailed error analysis of model predictions.
- ConfusionMatrixDisplay: Used to generate and visualize the confusion matrix to evaluate classification results.
- classification_report: Provides a detailed summary of model performance, including precision, recall, F1 score, and accuracy.

sklearn.model_selection:

- train_test_split: Splits datasets into training and testing/validation sets.

sklearn.utils:

- compute_class_weight: Calculate the weights for each class to handle unbalanced datasets during model training.

tensorflow.keras:

- Provides tools to build, train, and evaluate deep learning models, including Sequential, load_model, Conv2D, MaxPooling2D, Flatten, Dense, Dropout, Resizing, GlobalAveragePooling2D, ImageDataGenerator, EarlyStopping, Adam, MobileNetV2, NASNetMobile, and InceptionV3.

keras_tuner:

- Used to automate the tuning of hyperparameters, such as the learning rate and the number of layers, to optimize model performance, utilizing HyperBand Search.

3 Model Approaches

3.1 Pre-processing

In the pre-processing stage of our project, we started by importing the dataset containing the information about the images. We then checked the existence of duplicate and missing values, encountering some missing entries in the 'Benign or Malignant', 'Cancer Type', and 'Magnification' columns. Considering that the variables with missing data are linked to the image paths, we addressed the problem by extracting the missing information directly from the path using a function created to extract that. To conclude this stage we exported the preprocessed dataframe.

3.2 Binary Classification

We started by importing the dataset and keeping the images and their respective labels into separate variables.

Initially, the dataset was split into 80% for training and 20% for testing. Later, the 80% allocated for training was divided into 80% for actual training and 20% for validation. This resulted in approximately 64% of the dataset for training, 16% for validation and 20% for test.

Given the class distribution across the training, validation, and test sets, we conclude that there is an imbalance between the malignant and benign classes, with a higher number of malignant samples. Due to this imbalance, we decided to use the F1 score as our evaluation metric on the test set, because this metric considers both precision and recall, providing a more balanced measure of model performance. This is helpful to ensure that both false positives and false negatives are given proper weight in the evaluation.

Following that we imported the images using a composed function that included the preprocessing steps. These steps included the resizing of the images' dimensions and normalizing the pixel values. To manage the computational costs, the image sizes were reduced to 150x150, as the computer could not handle larger sizes.

Lastly, we decided to map the categorical labels, in our case "Benign" and "Malign", to numeric values, 0 and 1 respectively.

3.2.1 Model Implemented

Before reaching the best model we experimented with several ones. Firstly, we tested a small and simple model and a more complex one. This approach allows us to compare the performance of both extremes and determine the best starting point.

Following, we implemented an intermediate model, however, its performance was not satisfactory, so we included dropout layers, but despite the result being a little bit better it was still a poor result.

Data augmentation is a way to increase the size of the training dataset by making small changes to the existing data, and for this, we also tried a model. Different pre-trained models were tested and despite one achieving the best result, we still tried undersampling in a naive way.

After experimenting with a wide range of models, it was decided to utilize the pre-trained model, "NasNetMobile", and it was considered the best one. As this model was chosen to extract features of our dataset, its top layers were frozen.

Considering this, we added a single dense layer composed of 128 neurons, followed by a dropout layer with a rate of 0.3, and a final dense layer for output.

Due to an expensive computation cost, we opted to utilize a small batch size of 32 samples. Indeed, larger batch sizes were tested, but these caused problems, particularly within the Visual Studio Code, without yielding significant performance improvements compared to the smaller batch size.

3.2.2 Evaluation

As previously mentioned, the F1 Score was the metric used to calculate the model's performance, considering that the dataset is imbalanced.

For the malignant class, a score of 0.91 was obtained on the training set after testing the model. We consider that the most serious case related to our work is that the model does not detect the existence of malignant cancer, that is, a false negative. However, our model achieves a recall value of 0.93, indicating that the problem described above is unlikely to occur, with malignant cases being mostly identified. Related to the benign cases, the F1 score obtained is 0.80, which shows a decent balance. Although this is already a very high value in most cases, with the context of it being a medical issue it requires even higher values with a better balance. Further improvement is possible, mainly by reducing the number of false positive cases.

By analyzing the confusion matrix, Figure 1 in the appendix, it is possible to conclude that concerning benign cases, the final model managed to identify positively 383 cases, however, it incorrectly classified 113 cases as being malignant, false positives, leading to 0.77 being the value obtained in our model's recall. In the case of malignant tumours, the model positively identified 1005 cases, however, 81 malignant cases were identified as benign, a false negative, which is a very worrying error. Finally, in this case of malignant tumours, our model achieved a recall value of 0.93, which means that the model is very effective in identifying the majority of malignant cases.

3.3 Multi-class Classification

The initial steps of the multiclass classification stage were similar to those of the binary classification. After dividing the dataset in the same way, we verified whether the data was balanced. By analyzing the distribution of values across labels in each set, it was possible to verify class imbalance. Classes such as Phyllodes Tumour and Adenosis had much fewer samples than the most represented class, Ductal Carcinoma. Similar to the binary classification problem, we will address this imbalance by using the F1 score as the primary evaluation metric for the test set.

Regarding the types of breast cancer, we transformed the names of the classes to numerical labels, which means that to the names of the different types of cancer, a number is assigned.

3.3.1 Model Implemented

For this task, we started to test a simple model, with and without dropout layers. Next, we changed the optimizer and added complexity to both convolutional and dense layers. Using the best model architecture achieved, we tested the model with ImageDataGenerator. Once more, several pre-trained models were tested, but satisfactory results were not achieved. Then we performed transformations on the images, including adjustments to brightness, contrast, saturation and some noise addition. Additionally, we addressed class imbalance by assigning weights to each class, however, the results remain similar.

Finally, we performed a HyperBand search to select the optimal values for the model's hyperparameters. We concluded that the best performance was achieved using the model optimized through this search.

The exploration process determined that the model performs best with 3 convolutional layers, with the values [32, 96, 128] for filters and [3, 5, 5] for kernel sizes, which is possible to verify in Figure 2 presented in the appendix. Additionally, the HyperBand Search concluded that the dense layer should have 160 neurons to optimize the model's generalization capacity and, during the model compilation process, a learning rate of 0.0001 was identified as the ideal value to use.

Lastly, after testing our model with and without dropout layers, we concluded that using the dropout layers would lead to a better result.

3.3.2 Evaluation

Given the imbalance in our dataset, the F1 Score is once more the performance metric that will be used, to ensure a balanced evaluation of precision and recall.

The HyperBand Search and the saturation-adjusted approach were compared by their performance, revealing some differences in their effectiveness. The HyperBand Search model demonstrated superior performance across multiple metrics and classes when compared to the other approach.

The saturation-adjusted model achieved only 42% accuracy, struggling significantly with most classes, as it is possible to check by a weighted average F1 score of 28%. Classes such as Adenosis and Fibroadenoma, achieved an F1 score of 0%, revealing that this model failed when handling some classes.

On the other hand, the HyperBand Search achieved an overall accuracy of 61% and a weighted average F1 score of 58%. Performance across classes was more balanced, and for the ones mentioned before, Adenosis and Fibroadenoma, an F1 score of 57% and 55% were obtained respectively. However, for minority classes like Papillary Carcinoma and Phyllodes Tumor, the F1 score continues to achieve very low results.

Despite the advancements brought by the HyperBand Search model, neither approach achieves a level of performance suitable for real-world deployment. Both models are limited by their overall results, inconsistent performance across classes, and inability to address minority class predictions.

4 Error Analysis

During the training and evaluation of our model, several challenges were encountered that impacted the models' performances. Overfitting was a common issue, where the model performed well on the training data but struggled to generalize to new data, as seen from discrepancies between training and validation losses, as well as accuracies, in Figure 3 found in the appendix.

A limitation encountered during this project was the batch size. Initially, larger batch sizes were tested, but these frequently caused crashes due to memory constraints, particularly in RAM. To fix this, we chose to perform our models with a smaller batch size, despite affecting the performance.

Throughout the project, we faced some problems with rerunning models due to errors identified during the process. Each rerun often produced different results, requiring us to adjust parameters and comments.

We also attempted to use Google Colab, but we faced limitations as the image import process could not be completed due to time and memory constraints.

5 Future Work

Similarly to all projects, there is always potential for improvement. Given more time, various aspects of this project could have been improved or new techniques explored.

Since pre-trained models are trained to capture specific features of images, we would have experimented with the models used by adding more layers, adjusting the dimensions of existing layers, or testing alternative models. These enhancements would have been possible with access to better computational resources.

Additionally, given the existence of various callbacks and the fact that only one was used in our project, we could have experimented using different callbacks or even combinations of them.

In terms of oversampling techniques, only Data Augmentation was used. However, with more time to explore methods like SMOTE or GANs and adapt them to our problem, we could perhaps have improved the overall performance of our models.

Although we have used different model architectures, we also considered using a different one called Inception Recurrent Residual Convolutional Neural Network (IRRCNN). It is a hybrid model that combines the capabilities of inception modules, recurrent layers, and residual connections to identify spatial and sequential data patterns. However, due to time constraints and the complexity of understanding this architecture, it was not implemented.

Finally, regarding the HyperBand Search conducted during the multiclass classification stage, only a limited number of combinations were tested. With more time, we could have explored a wider range of combinations.

6 Conclusion

This project demonstrated deep learning's potential for improving breast cancer diagnosis by classifying histopathology images.

The binary classification model presented an overall good performance, achieving an F1 score of 0.91 and a recall of 0.93 for malignant cases, minimizing false negatives, an important factor in medical applications. However, the results decreased slightly for benign classifications, which resulted in an F1 score of 0.80 and a higher rate of false positives. Regarding the multi-class classification problem, the final model, optimized by the HyperBand search, achieved a moderate overall accuracy of 61% and a weighted F1 score of 58%. This model showed an improved performance for some minority classes, such as Adenosis (57%) and Fibroadenoma (55%).

Future improvements could involve experimenting with different architectures such as IRRCNN, using robust oversampling approaches like SMOTE or GANs, and performing deeper hyperparameter searches. Improved processing capabilities would also allow for greater image dimensions and batch sizes, potentially increasing performance.

In conclusion, while the findings demonstrate the promise of deep learning in medical imaging, significant modifications are required to obtain the consistency and precision required for real-world clinical applications.

7 References

- Breast cancer classification from histopathological ... (n.d.-a). https://arxiv.org/pdf/1811.04241
- Classification of breast cancer based on histology images using convolutional neural networks — IEEE Journals Magazine — IEEE Xplore. (n.d.). https://ieeexplore.ieee.org/document/8353225/
- Nawaz, Majid A., Adel Hassan, Taysir. (2018). Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network. International Journal of Advanced Computer Science and Applications. 9. 10.14569/IJACSA.2018.090645.

Appendix

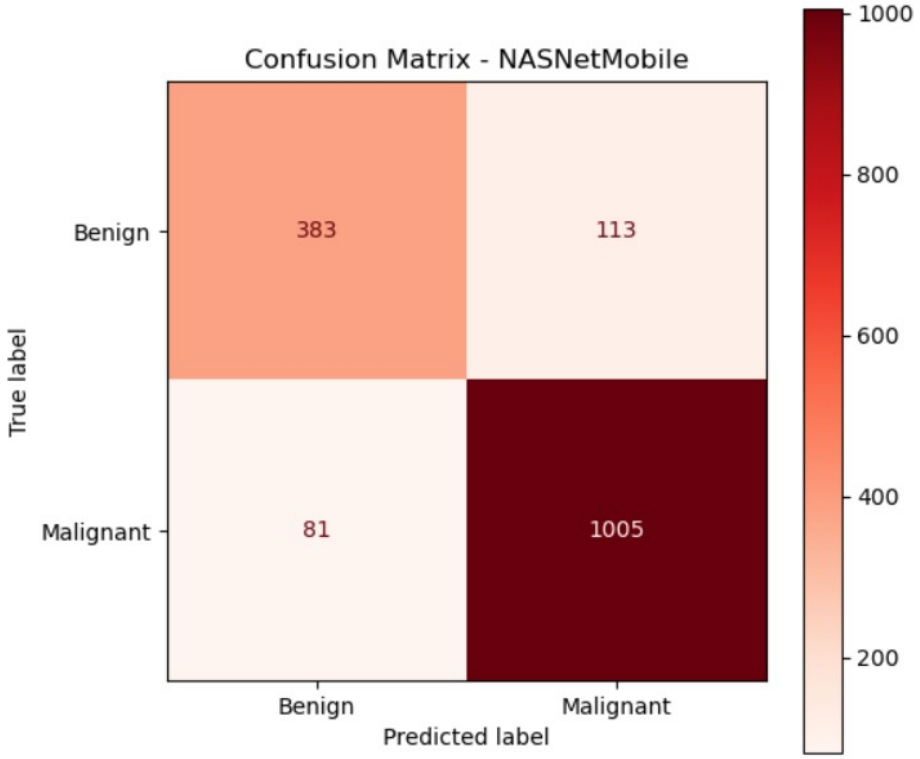


Figure 1: *Confusion Matrix of NASNetMobile Model*

```
Best Hyperparameters:
- Number of Conv Layers: 3
- Filters and Kernel Sizes: [32, 96, 128],
  [3, 5, 5]
- Dense Units: 160
- Learning Rate: 0.0001
```

Figure 2: *HyperBand Search Results*

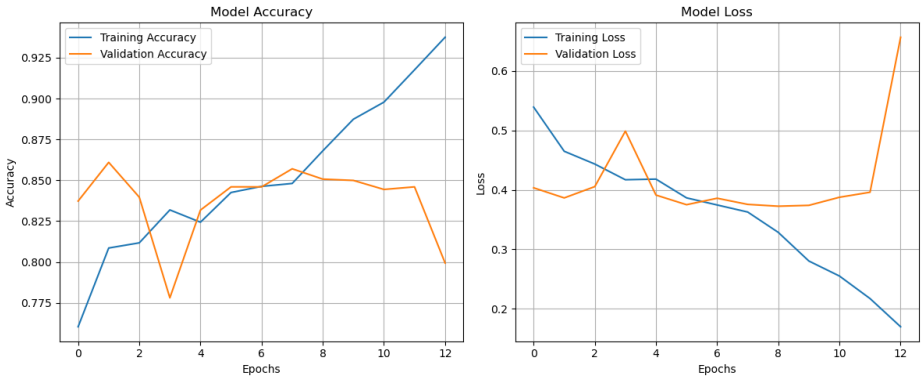


Figure 3: *Overfitting in Binary Classification*