UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA ELÉTRICA E BIOMÉDICA
CURSO DE ENGENHARIA BIOMÉDICA

SOFIA PINHEIRO KLAUTAU

MELANOMA CLASSIFICATION WITH NEURAL NETWORKS USING AN UNBALANCED DATASET OF SKIN LESION IMAGES

BELÉM-PA
2023

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA ELÉTRICA E BIOMÉDICA
CURSO DE ENGENHARIA BIOMÉDICA

SOFIA PINHEIRO KLAUTAU

MELANOMA CLASSIFICATION WITH NEURAL NETWORKS USING AN UNBALANCED DATASET OF SKIN LESION IMAGES

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia Életrica e Biomédica da Universidade Federal do Pará, como requisito parcial para a obtenção do Grau de Bacharel em Engenharia Biomédica.

Orientador: Prof. Dr. Leonardo Lira Ramalho
Coorientador: Prof[a]. Dr[a]. Ana Carolina Quintão Siravenha Müller

BELÉM-PA
2023

Sofia Pinheiro Klautau

Melanoma Classification with Neural Networks Using an Unbalanced Dataset of Skin Lesion Images

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia Életrica e Biomédica da Universidade Federal do Pará, como requisito parcial para a obtenção do Grau de Bacharel em Engenharia Biomédica.

Data da Defesa: 17 de julho de 2023

BANCA EXAMINADORA

Prof. Dr. Leonardo Lira Ramalho

(ORIENTADOR)

Profª. Drª. Ana Carolina Quintão Siravenha Müller

(COORIENTADOR)

Profª. Drª. Adriana Rosa Garcez Castro

(Membro – FEEB/ITEC/UFPA)

Prof. Dr. Ilan Sousa Correa

(Membro – FCT/ITEC/UFPA)

Profª. Drª. Carminda Célia Moura de Moura Carvalho

(Diretora da FEEB/ITEC/UFPA)

This work is dedicated to my parents and siblings, my biggest supporters.

ACKNOWLEDGEMENTS

I would first like to thank my advisors, Professor Leonardo Ramalho and Professor Ana Carolina Müller, for their invaluable guidance throughout the entire process of this work. Their patience and expertise have been instrumental in supporting my research journey, and I am very grateful for the opportunity to have been advised by them and for all their help.

My journey in Engineering has been shaped by many people, in different places and moments, and I am so grateful to have met so many incredible people.

To my friends from the 2018.2 Biomedical Engineering class at UFPA, thank you for all these years of companionship and unity. It was a pleasure studying with all of you.

Thank you to all my colleagues at LASSE, where I had the pleasure of working for two years, and learned so many invaluable things.

To all the professors who contributed to my education at UFPA, my sincerest thank you.

I thank my cherished friends from school whose friendships are still a part of my daily life, I thank the friends that I made while abroad who became my family, and the ones I made in and outside of classes. You have all been an anchor and instrumental in making daily life memorable.

I would like to also thank my dear family: my uncle Avertano Klautau who gave me my first Engineering textbook for Calculus I, my cousin Kayan Coutinho who gifted me his calculator from his own Engineering studies that I used for all of mine, my aunt Silvia Klautau who welcomed me into her home during my first semester while I living by myself, and all my cousins and grandparents who always cheered me on.

Lastly, to my sister Amanda Klautau and my brother Aldebaro Klautau Neto, thank you for always being helpful and for being my friends. To my beloved mother Sonja Klautau, so kind and unwaveringly supportive at all times, and to my father Aldebaro Klautau Junior, my favorite Engineer, great source of inspiration and the reason why I wanted to study at UFPA since I was a child, thank you so much. I would not have made it this far without you, and I will be always grateful for everything you have provided me, for believing in me and always encouraging me to follow my dreams.

Sofia Klautau
July 2023

“I am not afraid of storms, for I am learning how to sail my ship.”
(Louisa May Alcott)

RESUMO

As aplicações de Inteligência Artificial (IA) em vários campos são extensas e têm o potencial de revolucionar vários aspectos da saúde moderna, por exemplo, demonstrando avanços promissores na melhoria da precisão e eficiência da detecção e classificação do câncer de pele. Esta área de estudo é de grande importância, pois busca melhorar a identificação e o diagnóstico precoce do câncer de pele, impactando positivamente os resultados dos pacientes e as estratégias de tratamento. Este trabalho descreve um estudo realizado sobre o uso de conjuntos de dados desbalanceados para a classificação de imagens de lesões de câncer de pele usando Redes Neurais Artificiais, mais especificamente, um conjunto de dados que possui mais de 98% de amostras pertencentes à classe negativa. Três estratégias foram aplicadas para tentar mitigar as dificuldades causadas pela grande diferença no número de imagens em cada classe, no caso, lesões que são melanoma e lesões que não são melanoma: reduzir o número de amostras no conjunto de dados para equilibrá-lo , aplicando aumento de dados e aplicando pesos de classe. Além disso, métodos para otimizar o processo de treinamento de uma Rede Neural Convolucional são aplicados com sucesso para automatizar o processo de seleção de hiperparâmetros e o tempo de treinamento de modelos que usam grandes redes neurais como extratores de características é reduzido por causa disso. O aumento de dados e pesos de classe adotados neste trabalho ajudaram no procedimento de treinamento, mas não foram suficientes para produzir uma grande melhoria no desempenho, porém o último método foi aplicado no melhor resultado obtido.

Palavras-chave: Machine Learning, câncer de pele, melanoma, otimização, classificação de imagens, dataset desbalanceado.

ABSTRACT

The applications of Artificial Intelligence (AI) in various fields are extensive and have the potential to revolutionize various aspects of modern healthcare, for example, demonstrating promising advances in improving the accuracy and efficiency of skin cancer detection and classification. This area of study is of significant importance as it seeks to improve early identification and diagnosis of skin cancer, positively impacting patient outcomes and treatment strategies. This work describes a study carried out on the use of unbalanced datasets for the classification of images of skin cancer lesions using Artificial Neural Networks, more specifically, a dataset that has over 98% of samples belonging to the negative class. Three strategies were applied to try to mitigate the difficulties caused by the large difference in the number of images in each class, in this case, lesions that are melanoma and lesions that are not melanoma: reducing the number of samples in the dataset to balance it, applying data augmentation and applying class weights. In addition, methods for optimizing the training process of a Convolutional Neural Network are successfully applied to automate the hyperparameters selection process and the training time of models that use large neural networks as feature extractors is reduced because of it. The data augmentation and class weights adopted in this work helped the training procedure but were not enough to produce a large improvement in performance, but the latter method was applied in the best result obtained.

Keywords: Machine Learning, skin cancer lesion, melanoma, optimization, image classification, unbalanced dataset.

LIST OF FIGURES

Figure 1 – Examples of skin lesions that are melanoma and lesions that are not melanoma. 14
Figure 2 – Examples of low-quality and noisy skin lesion images. 25
Figure 3 – General architecture of the neural networks applied. 28
Figure 4 – Examples of original and augmented images. 33
Figure 5 – Intermediate Values Plot for the first model in the initial experiments. 36
Figure 6 – Optimization History Plot for the first model in the initial experiments. . . . 36
Figure 7 – Hyperparameter Importances Plot for the first model in the initial experiments. 37
Figure 8 – Intermediate Values Plot for the second model in the initial experiments. . . 37
Figure 9 – Optimization History Plot for the second model in the initial experiments. . 38
Figure 10 – Comparison of good and bad results on the second model. 39
Figure 11 – Hyperparameter Importances Plot for the second model in the initial experiments. 40
Figure 12 – Comparison between the best and pruned model in the fourth initial model. . 41
Figure 13 – Comparison between the best and a bad result in the fourth initial model. . . 42
Figure 14 – Plots comparing the best and last models in the third and fourth experiments. 43
Figure 15 – Confusion matrices for two balanced data subsets. 45
Figure 16 – Sample Complexity Curves. 46
Figure 17 – Optimization History Plots for two studies using class weights. 49
Figure 18 – Optimization History and Parameter Importances with data augmentation for Study 35. 51
Figure 19 – Optimization History and Parameter Importances with data augmentation for Study 40. 52
Figure 20 – Best model in contour plot between the number of neurons in layer 1 versus the dropout rate. 54
Figure 21 – Metrics results for the accuracy and AUC for the Studies. 55

LIST OF TABLES

Table 2 – Sample of the dataset in CSV format. 24
Table 3 – Example of possible Optuna numerical hyperparameter options. 29
Table 4 – Example of possible Optuna categorical hyperparameter options. 29
Table 5 – Total number of images for each class in the training, validation and test sets. 31
Table 6 – Data Augmentation Transformations Used for Training (C1 and C3). 32
Table 7 – Number of images selected for each class in the validation and test sets. . . . 34
Table 8 – Fixed hyperparameters for the third initial model selection study. 39
Table 9 – Hyperparameters for the best trial (number 14) in the third initial model selection study. 40
Table 10 – Fixed hyperparameters for the conclusions regarding the first initial model. . 43
Table 11 – Fixed hyperparameters to study the sample complexity of the data. 44
Table 12 – Hyperparameters for the best trial (number 72) in Study 26. 47
Table 13 – Results of the metrics for val, train and test sets in trial 72 of Study 26. 47
Table 14 – Hyperparameters for the best trial (number 155) in Study 32. 48
Table 15 – Hyperparameters for the best trial (number 559) in Study 35. 50
Table 16 – Hyperparameters for the best trial (number 538) in Study 39. 50
Table 17 – Hyperparameters for the best trial (number 38) in Study 40. 53
Table 18 – Results of the metrics for validation, train and test sets in trial 38 of Study 40. 53
Table 19 – Overview of the characteristics of each main Study. 53
Table 20 – Overview of the metrics results in each main Study. 54

LIST OF ACRONYMS

AI	Artificial Intelligence.
AUC	Area Under The Curve.
CNN	Convolutional Neural Network.
CNNs	Convolutional Neural Networks.
CSV	Comma-Separated Values.
CV	Computer Vision.
DICOM	Digital Image Communication In Medicine.
DL	Deep Learning.
EffNet	EfficientNet.
GPU	Graphics Processing Unit.
GPUs	Graphics Processing Units.
ISIC	International Skin Imaging Collaboration.
JPG	Joint Photographic Experts Group.
ML	Machine Learning.
ROC	Receiver Operating Characteristic.
SIIM	Society For Imaging Informatics In Medicine.
TL	Transfer Learning.
TNR	True Negative Rate.
TPR	True Positive Rate.

CONTENTS

1 INTRODUCTION 13
1.1 Objectives 15
1.2 Related Works 15
1.3 Work Structure 18

2 BASIC CONCEPTS 19
2.1 Classification in Machine Learning 19
2.2 Object Detection 20
2.3 Data Augmentation 20
2.4 Convolutional Neural Networks 21
2.5 Skin Cancer Detection 21

3 METHODOLOGY 23
3.1 Database 23
3.2 Neural Networks and Evaluation Metrics 24
3.2.1 Evaluation Metrics 24
3.2.2 EfficientNet Models and Transfer Learning 26
3.2.3 Optuna for Hyperparameter Optimization 28
3.2.4 Types of Results in Optuna 30
3.3 Data Usage and Processing 30
3.4 Strategies for Unbalanced Datasets 31
3.4.1 Data Balancing 31
3.4.2 Data Augmentation 31
3.4.3 Weights 32

4 RESULTS 34
4.1 Initial Model Selection Experiments for Optuna Investigation 34
4.1.1 First Initial Model Selection 35
4.1.2 Second Initial Model Selection 35
4.1.3 Third Initial Model Selection 38
4.1.4 Fourth Initial Model Selection 40
4.1.5 Discrepancies Between Last and Best Models 41
4.1.6 Conclusions Regarding the Initial Model Selection 42
4.2 Studying the Sample Complexity of Unbalanced Sets 43
4.3 Model Selection with Pre-calculated Backend Outputs 46
4.4 Unbalanced Datasets and Training with Class Weights 48
4.5 Data Augmentation 48

5 CONCLUSION AND FUTURE WORK 56

BIBLIOGRAPHY 59

1 INTRODUCTION

The advent of Artificial Intelligence (AI) and Machine Learning (ML) has indisputably caused a great impact on the healthcare field, for example , in medical diagnosis, personalized medicine, telemedicine and remote monitoring, drug discovery and development (KAUL et al., 2020). Also, the progressive growth of AI has sparked debate about the ethical considerations that must be analyzed before implementation (CHAR et al., 2020), such as algorithm transparency, data privacy, and the need for human oversight in critical decision-making processes.

Nevertheless, the potential applications for the role of AI are very vast and have the potential to transform many aspects of how modern medicine is applied to patients today. For instance, it has shown promise in improving the accuracy and efficiency of skin cancer detection (SEBASTIAN; PETER, 2022) with non-invasive techniques, the subject of study of this work.

Cancer is a disease with a high mortality rate that is best defined by the disorganized or uncontrollable growth of body cells (National Cancer Institute, 2021) that may form masses called tumors, and spread to other tissues, a phenomenon called metastasis. Tumors can be classified as benign, which means they will not spread and may be removed, and as malignant when they are cancerous and may both grow and spread (Cancer.Net, 2022).

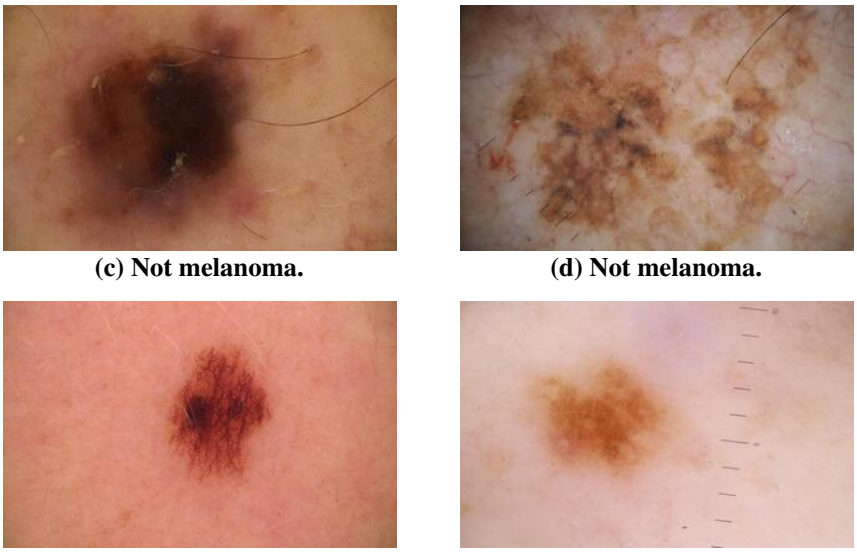
Many types of tumors can grow on the skin. The human skin consists of three main layers, the epidermis, which is the outer and visible layer, the dermis, which is the middle one, and the deepest layer, the hypodermis, or subcutaneous tissue (PUBLISHING, 2021). Skin cancer is among the most common types of cancer in humans, and it has many types, the primary ones being:

- The Basal Cell Carcinoma (BCC): a common type of cancer caused by the basal cells found in the lower epidermis, are usually located in areas that are more exposed to the sun (American Cancer Society, 2023), and rarely spreads to other body parts.
- Squamous Cell Carcinoma (SCC): another common type of cancer caused by the squamous cells in the epidermis, and it can be diagnosed on any part of the skin, although they are more likely to be also found in areas that have more sun exposure (American Cancer Society, 2023).
- Melanoma: the most dangerous type. It is caused by the melanocytes, cells that produce the melanin pigment that colors the skin, hair, and eyes and are located in the epidermis. Also, it only accounts for around 2% of all skin cancers, but causes the most skin cancer deaths (LINARES et al., 2015). Its biggest warning sign is a mole or spot that changes over time or suddenly appears anywhere on the body (American Cancer Society, 2023).

Most melanomas are found on the skin surface and tend to have specific characteristics of damage that medical professionals look for in order to diagnose and classify them, such as

Figure 1 – Examples of skin lesions that are melanoma and lesions that are not melanoma.

(a) Melanoma. **(b) Melanoma.**



(c) Not melanoma. **(d) Not melanoma.**

Source: the 2020 SIIM-ISIC competition dataset.

asymmetry, border format, color of lesion, diameter, and evolution (LINARES et al., 2015). Thus, they are more rapidly detected (CUMMINS et al., 2006), in comparison to internal tumors. Some examples are shown in Figure 1.

Therefore, images of skin lesions are very important when studying melanomas, because when diagnosed early, the chances of being cured with the least amount of treatment are higher (UConn Today, 2021).

It is important to note that currently, this process requires the expertise of dermatologists who are specialists in skin cancer, which makes it sometimes an expensive procedure and not easily available and affordable in developing countries (MELARKODE et al., 2023).

In this sense, the constant need for cancer specialists has elevated the demand for the development of automated diagnosis systems that may be capable of helping doctors detect skin cancer early (MELARKODE et al., 2023). It goes without saying that these systems should not be designed as replacements for doctors but be made in a way that dermatologists can apply them in order to optimize their patient care. It has been proven that it is important to contemplate human factors, such as personality, experience as a clinician, cognitive style, and preferences, and that they should be treated with more importance when designing and employing AI diagnostic tools (FELMINGHAM et al., 2021a). Furthermore, clinicians will be able to use the possible end product of an AI medical system with more efficiency and safety if the knowledge taken from human factors is considered.

Besides, another important question in AI for diagnostics is the fact that although there are many databases widely available that contain high-quality dermoscopic images, which are photos taken with the use of a non-invasive tool similar to a microscope that is applied by

dermatologists to diagnose skin lesions with better visual detail, they can still have significant difficulties, since in nature, the population of people without skin cancer is significantly larger than the population of individuals who have been in fact diagnosed with skin cancer.

This means that when creating databases, they are highly imbalanced, where examples of some lesion classes are incredibly outnumbered by other lesion types (ALAM et al., 2022). This, along with the different levels of image quality, makes it harder to achieve efficient results in classification tasks, seeing as the system will not have as many cancerous examples to learn from compared with the non-cancerous examples.

The importance of AI classification for skin cancer lies in its potential to improve early detection, enhance accuracy (MAGALHAES et al., 2019), and assist dermatologists in making more informed decisions. For this reason, this work applies a database that contains thousands of varied dermoscopic images and studies the different effects of methodologies used to lessen the disparities of the classes, in order to compare what strategies yield the best results for classification between a non-melanoma lesion and a melanoma.

1.1 Objectives

The primary objective of this study is to address the challenge of imbalanced datasets in the Machine Learning (ML) classification of skin cancer lesions, specifically focusing on an extensive database of images containing melanoma or non-melanoma skin lesions.

In this setting, this work aims to study the effects of two strategies to reduce class imbalance: data augmentation and the weighting of classes. These approaches will be thoroughly discussed in Chapter 3. Additionally, this research explores the impact of training Convolutional Neural Networks (CNNs) with the available data, while also leveraging Transfer Learning (TL) techniques. Also, an optimization tool to enhance performance and minimize computational costs is explored. It is called Optuna, an open-source hyperparameter optimizer for automatically fine-tuning hyperparameters such as the dropout rate, batch size, significantly reducing the training time required.

1.2 Related Works

The article by (ESTEVA et al., 2017) is a groundbreaking and well-cited study in which researchers developed a deep CNN that demonstrates the capability to classify skin lesions and diagnose skin cancer with a level of competence comparable to dermatologists. The CNN is trained on a dataset of thousands of clinical images, representing a myriad of different diseases, and achieves performance equivalent to that of 21 board-certified dermatologists in distinguishing between various types of skin cancer and benign lesions. The authors highlight the potential

of using deep learning algorithms on mobile devices to provide low-cost universal access to diagnostic care.

The paper written by (PEREZ et al., 2018) explores the use of data augmentation techniques to improve melanoma classification in skin lesion analysis using deep learning models. With deep learning models demanding substantial amounts of annotated data, the limited availability of such data for skin lesions poses a challenge. The study investigates the impact of various data augmentation scenarios on melanoma classification using three different CNN architectures: Inception-v4, ResNet, and DenseNet, showing the significance of data augmentation as a means to enhance melanoma classification and addressing the scarcity of annotated skin lesion images.

The paper by (SALIDO; RUIZ, 2018) presents a deep learning approach for the automatic detection and classification of melanoma in dermoscopy images. The system pre-processes the images to remove artifacts like hair and then uses a Convolutional Neural Network (CNN) based on the AlexNet model for image classification. By fine-tuning the pre-trained AlexNet model and utilizing transfer learning, the approach demonstrates the potential of deep learning in accurately detecting and classifying melanoma.

The paper by (DILDAR et al., 2021) presents a systematic review of deep learning techniques for the early detection of skin cancer. It emphasizes the importance of early diagnosis and discusses various approaches, and highlights the potential of computer-based technologies in improving the speed, accuracy, and non-invasive nature of skin cancer detection. It also identifies areas for further improvement in current diagnostic techniques and emphasizes the need for continued advancements in skin cancer detection systems.

A novel database called the HAM10000 ("Human Against Machine with 10000 training images") (TSCHANDL et al., 2018) groups high-quality dermoscopic photos curated from another dataset. The HAM10000 was designed to mitigate the problem of the small sample size of more common databases and the lack of dermoscopic images other than melanoma or common moles. The PH^2 (MENDONçA et al., 2013) is another example of a dermoscopic image dataset curated for Machine Learning tasks.

The paper by (ALWAKID et al., 2022) proposes a deep learning-based method for diagnosing skin cancer by extracting lesion zones with precision using the HAM10000 dataset. The approach involves enhancing the image quality using Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) and segmenting regions of interest (ROI) from the images. A Convolutional Neural Network (CNN) and a modified version of Resnet-50 are then used to classify skin lesions. The proposed CNN-based model outperforms previous studies with an accuracy of 0.86, precision of 0.84, recall of 0.86, and F-score of 0.86.

The paper by (CHEN et al., 2022) conducts a comprehensive review of medical image augmentation using Generative Adversarial Networks (GANs). It analyzes many related papers

and discusses the advantages of augmentation models, the relationship between augmented models and training set size, and the limitations of current models. It displays the potential of GAN-based augmentation in addressing the limited training sample challenge in medical image diagnosis while classifying GAN-based augmentation models into types and explaining their optimization objectives and mechanisms.

The article written by (FELMINGHAM et al., 2021b) emphasizes the importance of incorporating human factors in the design and implementation of Artificial Intelligence (AI) for skin cancer diagnosis in the real world, ultimately enhancing patient outcomes in skin cancer diagnosis. While AI algorithms have shown impressive accuracy in experimental settings, their integration into clinical practice requires an understanding of many human cognitive factors. Also, cognitive errors and biases that clinicians may experience when using AI diagnostic tools and the potential benefits and challenges of combining human intelligence with AI are discussed.

The paper by (KASSEM et al., 2021) presents another systematic review of computer-aided systems for skin lesion diagnosis, focusing on the diagnostic accuracy of traditional machine learning methods and deep learning models. The authors describe challenges such as small datasets and racial bias, and commonly used datasets are discussed, with the importance of high-performance computer-aided systems for accurate and rapid diagnoses being emphasized.

This systematic review by (WEN et al., 2022) aimed to identify and evaluate publicly available skin image datasets used for skin cancer diagnosis. Most of them originated from Europe, North America, and Oceania, and the majority contained dermoscopic images or macroscopic photographs. The limited applicability of these datasets to real-life clinical settings and the need for quality standards in reporting characteristics and metadata for skin image datasets are debated.

The article by (HA et al., 2020) describes the winning solution to the 2020 SIIM-ISIC Melanoma Classification Challenge hosted on the Kaggle platform. The authors used an ensemble of multiple and diverse models, applying data augmentation and using data from past years of the competition to try to reduce the class imbalance.

The present work is aligned in different dimensions with previous efforts to use Deep Learning (DL) in skin cancer diagnosis. For instance, most works in this area rely on convolutional layers. The closest papers are the ones that use the 2020 SIIM-ISIC Melanoma Classification competition dataset or similar, such as the aforementioned paper (HA et al., 2020).

The main similarities between the paper by (HA et al., 2020) and this work is that the authors in (HA et al., 2020) used Transfer Learning (TL) and Convolutional Neural Networks (CNNs), specifically some EfficientNet (EffNet) models. But differently than in this work, in (HA et al., 2020), the results are more deeply evaluated since the classifier input is not only the image, but also the metadata that describes the patient age, gender and lesion site, etc. Another aspect is that while this work focused on designing a single classifier, an ensemble of eighteen classifiers

was adopted in (HA et al., 2020). This was feasible because their results were obtained with the use of up to 8 Graphics Processing Units (GPUs) in parallel. Furthermore, while an Area Under the Curve (AUC) of 0.9490 on the private Kaggle leaderboard was reported by them, this work achieved an AUC of only 0.8698. It is important to note that the training, validation and test sets were different in both works, as it is reported in Chapter 4.

1.3 Work Structure

The remainder of this work is structured as follows:

- **Chapter 2**: Provides an overview of Machine Learning applications, skin cancer detection, and classification.
- **Chapter 3**: Describes the methodology used in this work, the database, and the type of neural networks applied.
- **Chapter 4**: Details the results obtained through the methods applied.
- **Chapter 5**: Concludes the work and presents the final considerations.

2 BASIC CONCEPTS

Over the years, the rapidly evolving technological landscape has largely benefited from the new development of AI (RUSSELL; NORVIG, 2009) and ML (GéRON, 2017) applications in healthcare. In this chapter, the fundamental concepts that form the foundation of this research are presented.

2.1 Classification in Machine Learning

ML is a field of study that focuses on developing algorithms and models capable of learning from data and making predictions, classifications and regressions, generally by coding in programming languages or platforms, such as Python, MATLAB, or Java. Developing effective models and applying them to real-world problems has become a growing trend in modern society (KOUROU et al., 2015).

For instance, supervised learning is a widely used approach where algorithms are trained on labeled data in which each input data has an associated label, and the algorithms learn patterns and relationships by using the labeled examples, enabling them to make accurate predictions or classifications for new and unseen data. Moreover, unsupervised learning deals with unlabeled data (OSISANWO et al., 2017), by aiming to uncover hidden patterns and structures within the data without any specific guidance from the humans that fed it the data.

Classification, a fundamental supervised learning task, involves categorizing or classifying data into predefined classes or categories based on its features that models can analyze. Besides, classification aims to assign categorical labels to input data, while regression focuses on predicting continuous numerical values. Its goal is to learn a decision boundary or mapping function that can accurately classify new instances of data into predefined classes or categories.

This common ML application plays a vital role in many practical applications, including spam detection, medical diagnosis (AMRANE et al., 2018), and image recognition. For classification models to be effective, proper training and validation to ensure reliable and accurate predictions are necessary.

Regression is another fundamental task in ML, in which models aim to approximate the relationship between input variables and a continuous target variable, enabling predictions of unknown values within a given range. Regression is widely employed in areas like stock market forecasting, house price prediction, and demand forecasting.

After training, model evaluation and validation are crucial to assess the performance of ML models. After training a model on a labeled dataset, it needs to be tested on unseen data to measure its accuracy, precision, recall, and other metrics. In addition, cross-validation techniques are employed to obtain a more robust estimation of the model's performance, a

technique in which the data is divided into a fixed number of folds or partitions and the analysis is performed on each partition, so that the resulting error estimates are averaged to obtain an overall performance measure (TEAM, 2022).

Also, overfitting and underfitting are common challenges in ML (RUSSELL; NORVIG, 2009). Overfitting occurs when a model becomes too complex and learns the training data too well, leading to poor performance on new and unseen data. Underfitting, on the other hand, happens when a model acts in a way that is too simplistic and fails to capture the underlying patterns in the data. Therefore, finding the right balance is crucial to ensure the model generalizes well to unseen data.

This work deals with a specific classification model, which has only two classes. In this case, metrics such as accuracy can be complemented by others, such as false positives and false negatives (RUSSELL; NORVIG, 2009).

2.2 Object Detection

Object detection is another crucial task in ML that involves not only identifying the presence of objects in an image or video but also localizing their positions. Unlike classification, which assigns a single label to an entire image, object detection aims to detect and recognize multiple objects within an image and provide their precise bounding boxes. This enables practical applications in the industry such as video surveillance and image-based search engines.

One popular approach is the use of methods based in DL, particularly CNNs. They are trained on large datasets and learn to extract meaningful features from images, enabling them to detect objects with high accuracy (DHILLON; VERMA, 2020). The combination of localization and classification allows object detection models to accurately identify and locate multiple objects within an image.

Accordingly, object detection tasks rely on classification to determine the specific class or label associated with the detected objects, involving training the model to recognize and classify them into predefined categories. By integrating classification into the object detection pipeline, the model can provide not only the presence and position of objects but also their corresponding labels.

2.3 Data Augmentation

Data augmentation is a regularization technique to artificially increase the amount of samples in the training set by creating new examples based on the existing ones in order to improve the performance of the model during validation or testing. It involves applying a variety of transformations to existing data samples, resulting in augmented versions of the original dataset. Many of the well-known DL frameworks support it, offering image transformations such as flip-

ping, rotating, cropping, and scaling, or combinations of them (SHORTEN; KHOSHGOFTAAR, 2019).

Data augmentation can also mitigate the risk of overfitting by expanding the size of the training set and reducing the model's dependence on specific features. By incorporating this technique into the training process, it is possible to introduce diversity and variation which helps the model generalize better to unseen data and make accurate predictions in real-world scenarios.

2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm inspired by the anatomy and physiology of the human brain and its neural connections (O'SHEA; NASH, 2015). They are designed to process and analyze visual data, such as images and videos, and are usually structured with layers of connected neurons, where the layers learn features or characteristics about the data, with early layers learning simple features like edges and textures, while deeper layers can learn more complex patterns and objects (MANDAL, 2021). One of the key aspects of these types of networks is the use of convolutional operations, which allow them to efficiently extract and detect features, and then be applied to tasks like classification and object detection.

One of the fundamental components of CNNs is the convolutional layer, which performs the convolution operation by sliding a small filter or kernel across the input image (SINGH, 2020), a value that is usually chosen by the programmer, and these filters have weights that are learned during the training process. Additionally, there may be pooling layers, which are commonly used to reduce the spatial dimensions and create spatial invariance (ZAFAR et al., 2022), enabling the network to recognize patterns regardless of their location in the input and be able to generalize better to unseen data.

2.5 Skin Cancer Detection

Binary skin lesion image classification is a crucial task in the field of dermatology and computer-aided diagnosis. It usually involves the classification of skin lesion images into two categories: malignant, or cancerous, and benign, or non-cancerous. This task plays a significant role in assisting dermatologists in the early detection and diagnosis of skin cancer, and ML algorithms, particularly DL models, have shown promising results in this domain (WU et al., 2022).

Skin cancer detection models are trained on large datasets of labeled skin lesion images, learning patterns, and features that distinguish between malignant and benign lesions, or are even able to distinguish between different types of cancerous lesions.

Said models typically utilize CNNs that are designed to extract features from skin lesion images, with the DL models being trained on annotated datasets containing diverse skin lesion images with accurate labels (HAGGENMÜLLER et al., 2021). During training, the model learns to automatically identify discriminative features indicative of malignant, benign, or other lesions. This process involves multiple layers of convolutions, pooling, and non-linear activations, enabling the model to capture complex patterns and texture information.

Nowadays, these advanced ML models for skin lesion classification may provide dermatologists and medical practitioners in this field with a valuable tool to support accurate and efficient diagnosis of skin lesions, which may improve patient outcomes by helping achieve an early diagnosis and by reducing unnecessary biopsies.

3 METHODOLOGY

This chapter pertains to the methodology developed in this work. Section 3.1 describes the database used, Section 3.2 discusses the Neural Networks applied, Section3.3 describes the way the data is used, and Section 3.4 talks about the strategies adopted to try and lessen the issues caused by the great unbalance in the data.

3.1 Database

The adopted database is part of a joint effort between the Society for Imaging Informatics in Medicine (SIIM) and the International Skin Imaging Collaboration (ISIC) to create the *SIIM-ISIC Melanoma Classification* database (ROTEMBERG et al., 2021). It is available on the Kaggle platform as part of a classification task competition from 2020 for melanoma detection in lesion images. It contains over 100 GB of data in total, comprised of images in Joint Photographic Experts Group (JPG) and Digital Image Communication in Medicine (DICOM) formats, along with patient records files in TFRecord format. ISIC is an internationally-recognized organization responsible for many of these competitions because its mission lies in supporting efforts to reduce the lethality of melanoma cancer and unnecessary biopsies by investigating options regarding the improvement of early detection using AI.

The JPG image data, which was applied in this work, is divided into folders that competition participants are able to directly download, with 33126 images in the training set, and 10982 images in the test set, all associated with a unique filename. These images all contain close-up shots of skin lesions and have varying resolutions, and each filename has a respective line in a Comma-Separated Values (CSV) file that lists additional categorical metadata for that specific lesion, such as detailed diagnosis information, sex and age of the patient, the lesion site in the body and whether it is malignant or benign, as shown in the example on Table 2.

Its images are from different hospitals and universities around the world: the Melanoma Institute Australia, the Hospital Clínic de Barcelona, the University of Athens Medical School, the Memorial Sloan Kettering Cancer Center, the University of Queensland, and the Medical University of Vienna.

Since the target variable for this competition was the binary classification of whether or not an image represents a melanoma, all images in the training set included their corresponding labels in the CSV file, with the target equal to 1 for malignant melanoma and equal to 0 for a lesion that is not melanoma, or benign.

It's important to note that the dataset exhibits great class imbalance since the vast majority of the images are of benign lesions, something that can be attributed to the fact that in nature there are less occurring cases of cancer than of lesions that are simply not cancer. 32542 out of the 33126 training set images were negative for melanoma cancer, or benign, while only 584

Table 2 – Sample of the dataset in CSV format.

Image Name	Patient ID	Sex	Age	Anatom. Site	Diagnosis	Ben./Mal.	Target
ISIC_2637011	IP_7279968	male	45.0	head/neck	unknown	benign	0
ISIC_0015719	IP_3075186	female	45.0	upper extremity	unknown	benign	0
ISIC_0052212	IP_2842074	female	50.0	lower extremity	nevus	benign	0
ISIC_0149568	IP_0962375	female	55.0	upper extremity	melanoma	malignant	1

were positive for melanoma cancer, or malignant, a ratio that must be taken into account during the development of ML models.

Another important factor to take into account is that since the dataset has a big assortment of images, it also contains many ones with unwanted artifacts that can interfere with the quality or interpretation of a lesion, or cause distractions that can make it more challenging to analyze or classify the lesions accurately. After analyzing these images, this work considers many of them to be "low-quality" or "noisy" images due to the presence of artifacts such as hair, pen markings, or the marks of the dermoscopy equipment ruler for measuring the size of the lesion. Figure 2 displays examples of these artifacts.

The interest in using this specific dataset rather than one specifically curated with only the best quality dermoscopic images is the variety of the data is the fact that it is obviously more difficult to use than a curated one. Since the ISIC datasets are collections of images from various sources they tend to be more diverse, and that ensures a broader representation of different image characteristics, such as lighting conditions and image qualities. In this work, it is considered that these factors can help the model learn robust features that are more likely to generalize in different scenarios.

3.2 Neural Networks and Evaluation Metrics

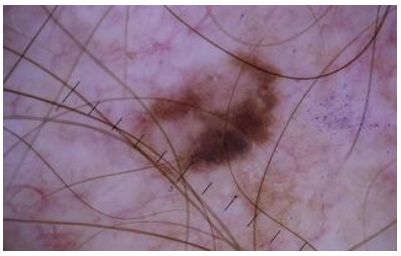
This section presents the evaluation metrics applied in this work in Section 3.2.1, with the neural networks applied for TL being described in Section 3.2.2 and the hyperparameter optimization tool Optuna being described in Section 3.2.3.

3.2.1 Evaluation Metrics

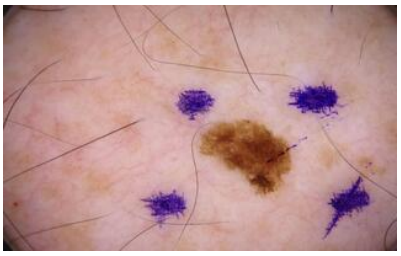
In this work, two different approaches toward the training of an ML system for melanoma classification to study the effects of having an unbalanced dataset are applied. The first approach

Figure 2 – Examples of low-quality and noisy skin lesion images.

(a) Image with hair and dermoscopy ruler.



(b) Image with hair and pen ink markings.

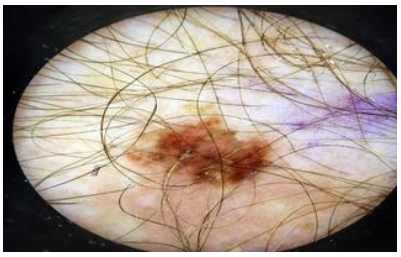


(c) Image with hair.

(d) Image that is too cropped.

(e) Image with pen ink markings.

(f) Image with microscope border and hair.



Source: the 2020 SIIM-ISIC competition dataset.

involves training a designed CNN and the second one pertains to the use of a specific CNN architecture called EfficientNet as a feature extractor for the use of TL.

Regarding the evaluation metrics of the ML systems, it is known that there are many available, the most famous of all being arguably accuracy. Nevertheless, in a skin lesion image classification task with a huge data imbalance, the model can be prone to overfitting and being biased towards the class with a bigger prevalence, which in this work's case is the negative class (lesions that are not melanoma). Therefore, if one builds a system that is classifying cancer and has a very high accuracy, one must take into consideration the possible class imbalance bias aforementioned, and the fact it might be achieving a high accuracy because it is only misclassifying the positive (lesions that are melanoma) examples, and getting all of the negative ones right. Therefore, methods to deal with class imbalance were devised and tried, and are detailed in Section 3.4.

Also, considering the fact that when discussing cancer, although it is important to reduce the number of unnecessary biopsies performed on patients, a pre-biopsy diagnosis that is falsely negative is more dangerous than one that is falsely positive, seeing as the former might cause the

disease to evolve unknowingly to the patient, while the latter might only cause an inconvenience until a proper biopsy diagnosis.

In this sense, this work considers accuracy an important metric, but one that doesn't completely describe the efficacy of a model in classifying melanomas. It is necessary to minimize the number of diagnoses that are false negatives, and a metric that takes that into account is the specificity, also called recall, which measures the number of positives that were correctly classified (BHANDARI, 2023), along with the specificity, which calculates the number of negatives that were rightfully classified. The equations for the recall and specificity are listed in Equations 3.1 and 3.2, respectively.

Recall, or True Positive Rate (TPR), is defined as

$$\text{TPR} = \frac{TP}{TP+FN}, \tag{3.1}$$

where TP (True Positives) is the number of values that were correctly predicted as positive, and FN (False Negatives) is the number of values that were incorrectly predicted as negative.

Specificity, or True Negative Rate (TNR), is defined as

$$\text{TNR} = \frac{FP}{FP+TN}, \tag{3.2}$$

where TN (True Negatives) is the number of values that were correctly predicted as negative, and FP (False Positives) is the number of values that were incorrectly predicted as positive.

Another valuable metric is the AUC, which is the measurement of the area under the Receiver Operating Characteristic (ROC) curve. This ROC curve is the result of a classification problem, and it is the plot of the recall (TPR) against the specificity (TNR) at certain threshold values. It is based on probabilities and can consider the model's performance across different possible classification thresholds, rather than a single threshold like the accuracy does. Along with the accuracy during the validation, this was the most important metric monitored.

However, one of the difficulties encountered throughout the development of the research was the fact that when training CNNs, there are many hyperparameter options that when changed even slightly, may cause varied effects on the outcome of the training. This means that it becomes harder to easily test many options in a time-saving way because the computational costs can be great and at each instance, it is necessary to train the model, it may require multiple hours to finish training and the results might not be satisfactory, which leads to repeating the process. For this reason, this work explored a way to optimize this procedure, which will be further detailed in 3.2.3.

3.2.2 EfficientNet Models and Transfer Learning

The second approach also uses the TensorFlow and Keras libraries but involved applying a pre-trained model as a feature extractor for TL, a technique that involves using said pre-trained

ML model in a new task in order to improve the performance. To use a pre-trained model for TL, one typically removes the original classification part of the model and adds new layers that are more suitable for whatever is desired. Afterward, these new layers are then trained on the new dataset while being able to keep the pre-trained weights frozen, which might help accelerate the convergence (BRINKER et al., 2018).

The model chosen for TL was the EffNet (TAN; LE, 2019), which proposes a novel and efficient method of scaling by balancing the width, resolution, and depth of the network using a compound scale coefficient, rather than using arbitrary values. Furthermore, EfficientNet has many different pre-trained variants on the ImageNet database (DENG et al., 2009), which contains thousands of images and is designed for object detection training purposes. In addition, the EffNet family has two different versions, EfficientNet and EfficientNetV2, the second one being newer and built upon the first version (TAN; LE, 2021). Its models are pre-trained on large-scale image datasets and on TensorFlow Hub, an open ML repository, they are available for use in two options: a feature extractor and a classifier.

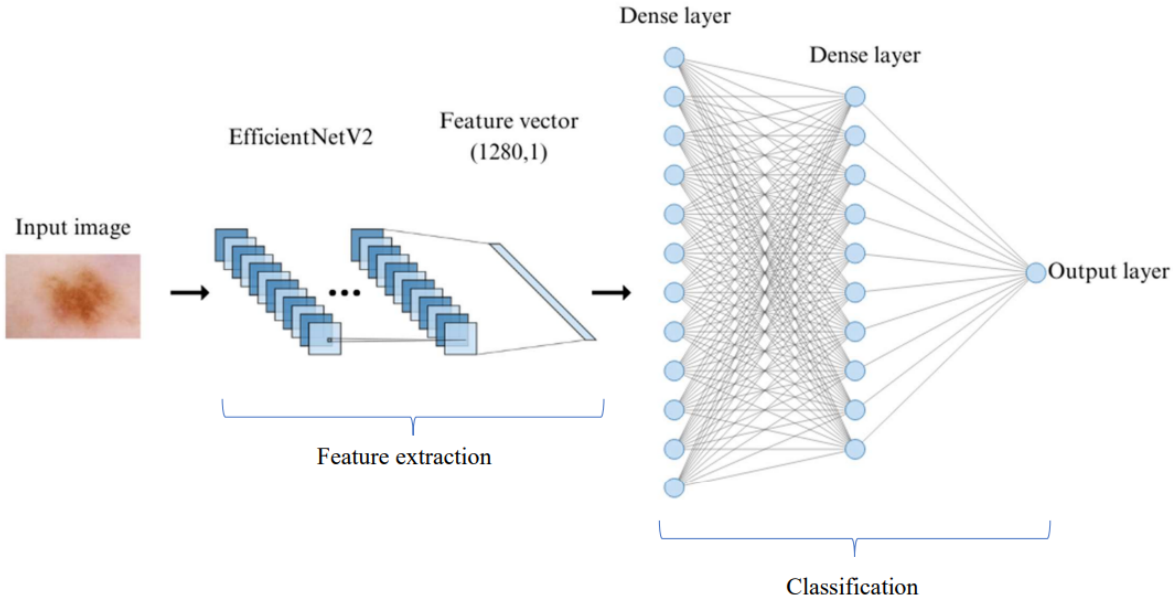
The inspiration to use the EffNet family of models came from the work by (HA et al., 2020), which applied it and managed to get the best results of the 2020 *SIIM-ISIC Melanoma Classification* challenge. Also, the models were proven to achieve a higher accuracy while having a smaller number of parameters than other existing models (TAN; LE, 2019) such as ResNet (HE et al., 2016) and Inception (SZEGEDY et al., 2016).

This work mainly utilizes the EffNetV2 as a feature extractor and then adds the extra classification layers as needed. So for ease of explanation, it is considered a two-part structure: the first one being an extractor of features (EffNet) and the second one being a classifier (Neural Network with dense layers). The EfficientNetV2 model used more often has approximately 7 million parameters, all of which are frozen for training, which means they are not re-trained. The other EfficientNetV2 model used was bigger and had around frozen 200 million parameters, but was not used as much as the smaller one. Nonetheless, every time data is fed into EffNet in order to arrive at the classifier, which has trainable parameters, they are still processed by the entire pre-trained model, and that ends up being generally a very long activity.

Consequently, a method to cut down on training time was devised: the images were fed as input to the EffNet model normally, and the code generated the input feature vector of characteristics for each class. These vectors were saved and then used directly as input for the classifier, which meant that the data did not need to be processed at each training instance by the entirety of the first part (feature extraction with EffNet), reducing exponentially the time needed for a system to converge.

Figure 3 shows the general architecture of the full model used in this work, with the input images being put through the EffNet in order to have their features extracted and saved into a vector of dimension (1280,1) features for each of the two classes. This feature vector is then used as the input for the dense layers, which in turn leads to the output layer.

Figure 3 – General architecture of the neural networks applied.



Source: author.

Even though TL reduces training time, the constant need to change and fine-tune the hyperparameters in order to try to achieve better training results still caused issues in relation to the time consumed. Therefore, an idea surged to use a specialized tool aimed at expediting the aforementioned process.

3.2.3 Optuna for Hyperparameter Optimization

The tool chosen was Optuna (AKIBA et al., 2019), a powerful open-source hyperparameter optimizer for automatically choosing hyperparameters such as the dropout rate, batch size, and layer activation as can be seen in Tables 3 and 4. Using an efficient search algorithm and pruning technique, the optimization process becomes much more cost-effective.

It supports Python 3.7 and higher, and can be easily installed and used with a multitude of ML and DL frameworks; in this work's case, TensorFlow and Keras. Optuna works by allowing the user to set a threshold of values or a vector of options which it will then use to create different combinations of hyperparameters, with each being one trained model. Every combination of hyperparameters suggested by Optuna is a new model trained, and each one of them is called a *trial*. The sequence of all the *trials* is called a *study*. Therefore, an Optuna *study* represents all the different models trained using different combinations of hyperparameters.

The number of trials can be chosen by the user, and at the end of the whole study, which includes training a new model for each trial, Optuna finds the trial with the model, or *trial*, that has the best performance and its combination of specific hyperparameters, by monitoring

Table 3 – Example of possible Optuna numerical hyperparameter options.

Hyperparameter	Interval of values		Format
	Minimum	Maximum	
Batch Size	1	15	Integer
Dropout Rate	0.2	0.8	Float
Learning Rate	1e-5	1e-2	Float
Number of Dense Layers	1	4	Integer

Table 4 – Example of possible Optuna categorical hyperparameter options.

Hyperparameter	Vector of options		Format
	Option A	Option B	
Activation Function	Relu	Tanh	Categorical
Batch Normalization	True	False	Categorical

a certain metric. From there, it is possible to generate graphs and analyses of the study results. Using Optuna along with the method mentioned in 3.2.2 of saving the output feature vectors of the EffNet model allows for much faster training, while at the same time generating a numerous amount of results with the best one delimited.

Furthermore, the number of epochs can also be chosen by the user, and remain the same for all trials. This means that *early stopping* can be implemented, a regularization technique to avoid overfitting that is able to conclude the training process if a monitored metric has not improved after a set amount of epochs.

As mentioned in Section 3.2, this work uses two metrics: the AUC of the ROC curve, and the accuracy during the validation. In such way, for each study, a metric to monitor must be chosen in order for Optuna to be able to analyze it and see if the training is succeeding or not, and it is called the *objective value*. This objective value is monitored during training, and seeing as it is computed at each epoch, they are thus considered intermediate. Therefore, a study has an objective value (either the validation accuracy or the the AUC of the ROC curve), and at each new epoch has a new *intermediate objective value*, which is the objective value computed each epoch, in order to evaluate if a trial should be aborted or not.

Additionally, Optuna contains a feature called *pruning*, which also works as an early stopper during each trial, but in an automated way. If Optuna's pruners detect that a trial is unpromising and has poor intermediate objective values, they abort it before the end. By stopping the training process for such hyperparameters and focusing on those with better values, it is possible to improve the overall performance and efficiency of the model.

3.2.4 Types of Results in Optuna

Optuna, the tool explained in Section 3.2.3 for finding a good (ideally, the optimal) tuning of the hyperparameters in a model, is able to take its results and display them in many different and useful ways for research purposes, with visualization plots that can help to understand the relationship between hyperparameters. In this work, four graphs were chosen to be interpreted.

- **Optimization History Plot**: plots the history of the Objective Value (function) being monitored in the study. Creates a comparison between the trial that got the highest score for the Objective Value, and shows the evolution of a study along the trials.
- **Hyperparameter Importance**: evaluates the hyperparameters chosen in a study by an order of importance. For example, if varying the number of dense layers caused the most significant impact on the results, the dense layer will be considered the most important hyperparameter for that study.
- **Intermediate Values Plot**: plots the intermediate values of all trials in a study, where it is possible to see which trials were pruned.
- **Contour Plot**: plots the relationship of pairs of hyperparameters in a study, as a contour plot.

In summary, both the CNN and the TL approaches apply Optuna for hyperparameter optimization, keeping in mind that in the latter, only the last layers for classification in the model are optimized. These are the efforts regarding the training process, the models used and the metrics applied. In the next subsection, the data application and processing will be explained.

3.3 Data Usage and Processing

In this work, only the train set of the 2020 SIIM-ISIC Melanoma Classification competition dataset was used, because it contained the labels for the classes, unlike the test set. Said train set contained in total of 33126 images and was redivided for the task, with 70% being used as the new train set and 30% for the test set. The validation set was made by taking out 20% of the images in the new train set, and the three sets were then divided into classes with 0 meaning the class negative for melanoma and 1 meaning positive for melanoma. The exact number of images in each class of the sets is included below in Table 5.

As it is possible to see, there is a great imbalance in the amount of images in each class, which coupled with the immense variability of the data, became two further challenges that this work aimed to solve.

Table 5 – Total number of images for each class in the training, validation and test sets.

Train set		Val set		Test set	
Class 0	**Class 1**	**Class 0**	**Class 1**	**Class 0**	**Class 1**
18227	324	4561	76	9754	184

3.4 Strategies for Unbalanced Datasets

In this section, the methods tried to reduce the overfitting of the network caused by the unbalanced dataset are described.

3.4.1 Data Balancing

It is known that in general, balanced data is often easier to work with compared to unbalanced data in ML tasks. When the dataset is balanced, unlike the one in this research, meaning that the number of instances in each class is roughly equal or evenly distributed, it allows the model to learn from a similar number of examples from each class, preventing biases toward the majority class, in this work's case, the negative class (not melanoma).

However, unbalanced data can cause challenges that may result in poor performance in the minority class. Since the model may not have enough information to learn the minority class (melanoma) patterns effectively, it can lead to low classification results which would be inadequate in this work's case considering it is important to minimize the number of wrong diagnoses.

In this sense, the first strategy devised to try to lessen the problem of the unbalanced data was simply balancing them and observing the effects that gradually increasing the number of negative samples (the majority) would have on the results. While having 324 positive samples for melanoma in the training set, it was possible to start by randomly selecting 324 negative samples from the entire negative class and gradually increment that value.

3.4.2 Data Augmentation

This work applied the Albumentations open-source library in Python for image augmentation, benefiting from its online demo where its user can see in real-time the effects of a chosen combination of transformations in an *augmentation pipeline* (BUSLAEV et al., 2020). Their values and thresholds can be set by the user, with a 50% default probability that each transformation in the pipeline will be applied to the data, an option that can be also changed by the user.

Albumentations is optimized for speed and uses the OpenCV library for Computer Vision (CV) applications, making it suitable for large-scale datasets like the one present in this work. To

create a new dataset of augmented skin lesion images, two combinations (one called C1 and the other called C3, which can be seen in Table 6) of these six transformations were applied:

a) **CLAHE**: Applies Contrast Limited Adaptive Histogram Equalization to the input images, a technique to improve the contrast in them.

b) **HueSaturationValue**: Randomly changes the hue, saturation, and value of the input images.

c) **Rotate**: Rotates the input images by an angle selected randomly from the uniform distribution, by default (-90, 90).

d) **RandomRotate90**: randomly rotates the input by 90 degrees zero or more times.

e) **HorizontalFlip**: flips the input images horizontally around the y-axis.

f) **VerticalFlip**: flips the input images vertically around the x-axis.

Table 6 – Data Augmentation Transformations Used for Training (C1 and C3).

C1	C3
RandomRotate90	CLAHE
HorizontalFlip	HueSaturationValue
VerticalFlip	Rotate

Examples of the augmented images based on an original one are shown in Figure 4. Subfigure 4a is sourced from the 2020 SIIM-ISIC Melanoma Classification competition dataset, while subfigures 4b, 4c and 4d are sourced by the author based on subfigure 4a.

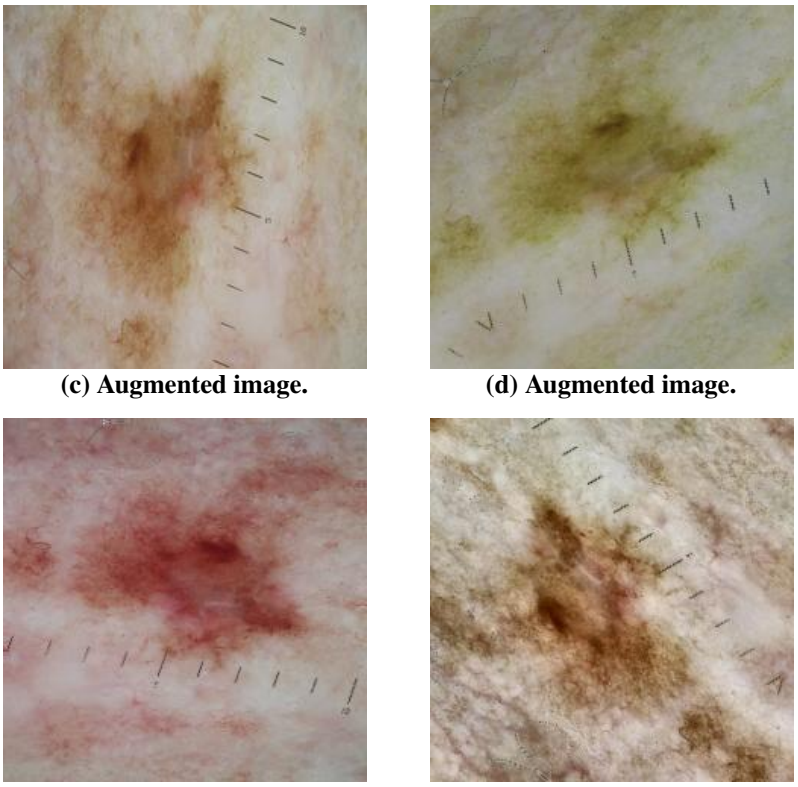
3.4.3 Weights

Using weights to balance the classes without the use of data augmentation was deemed an option after analyzing the results of a *dummy classifier*, a type of classifier that makes predictions using simple rules or strategies, without learning from the data. Therefore, it is typically used when dealing with imbalanced datasets or when the task requires random guessing (MAMONU, 2018).

The DummyClassifier available on Scikit-learn, a free ML library for Python, was used to calculate the most frequent class label seen in the training set. It verified, in basic terms, the mathematical probability of error if the classifier always chooses the label in that set that is more prevalent than the other. The result with the original training set containing 18227 negative samples and only 324 positives for melanoma was over 98%. This means that without having to learn anything specific about the data, this simple classifier would be able to correctly tell almost

Figure 4 – Original image from the dataset and examples of new images generated using Albumentations.

(a) Original ISIC 2020 image. **(b) Augmented image.**



(c) Augmented image. **(d) Augmented image.**

Source: the SIIM-ISIC competition dataset and the author.

all samples as not being melanoma, simply by always predicting “0”. Hence, a high accuracy suggests that either the classifier is performing well or that the dataset is truly highly imbalanced.

For this reason, the weights were applied, seeing as they are a very cost-effective way of balancing data without the need for augmentation or any other technique, by giving more importance to the minority class (melanoma) in the cost function of the algorithm so that it could provide a higher penalty to it, making the model focus on reducing the number of errors.

In this case, the gradient parcel is weighted differently such that the positive classes have larger weights due to their small number. The weights are calculated such that both positive and negative have a (weighted) probability of 0.5, each.

4 RESULTS

This chapter evaluates the proposed method and discusses the resulting data obtained. Firstly, it starts by explaining the types of results, then elaborates on the different types of strategies for working with imbalanced datasets. The loss function in all experiments in this chapter was the well-known binary cross-entropy for classification.

It's important to note that during the experiments, the original test and validation sets described in Table 5 were used, but not entirely, in order to make sure that the model was not achieving a high accuracy inadequately since the original sets were highly unbalanced, two balanced subsets for the validation and test were created from the original set. The number of images in each of the balanced validation and test sets is described in Table 7.

Table 7 – Number of images selected for each class in the validation and test sets.

Val set		**Test set**	
Class 0	**Class 1**	**Class 0**	**Class 1**
76	76	184	184

The following sections explain different types of experiments conducted using Optuna. In Section 4.1, the initial experiments are conducted for learning how to use Optuna and its effects. Section 4.2 studies the complexity of the dataset, while Section 4.3 talks about pre-calculating the feature vectors for the input images in order to cut down on training time. The two last sections deal with strategies for unbalanced datasets, with Section 4.4 describing the use of class weights and Section 4.5 explaining the use of data augmentation.

4.1 Initial Model Selection Experiments for Optuna Investigation

The strategy adopted for the initial experiments was of learning how to apply Optuna for the adopted dataset and understanding how it the hyperparameter selection works, therefore four different studies were deemed relevant and their results and findings are explained in Sections 4.1.1 through 4.1.6, with Section 4.1.1 indicating the first experiment conducted with Optuna and many hyperparameter intervals available for optimization. Section 4.1.2 discusses the second initial experiment which differs from the first by reducing the amount of hyperparameters Optuna can optimize. Section 4.1.3

To avoid problems with unbalanced classes, this experiment uses balanced sets with 50% positive and 50% negative labels. Therefore, the train set has 2×324 examples, 324 positive and 324 negative labels. The test set has 2×184 examples, 184 positives, and 184 negative labels. The validation set has 2×76 examples, 76 positives, and 76 negatives labels. The validation and

test sets are always kept balanced with 76 and 184 positive examples for each class, respectively. Hence, the total number of examples in each dataset is 152 and 368 examples. The same is done for the training set so that it contains 648 total samples. These numbers are low since they are the equivalent of all the samples present in the positive training, validation and test sets.

4.1.1 First Initial Model Selection

The metric monitored was the validation accuracy. It is best not to use the training set accuracy because it probably leads to overfitting the training data. Using Optuna, the first model selection was done in order to get a rough estimate of the neural network topology and hyperparameters, then refine it with other Optuna studies.

Figure 5 shows some intermediate plots for the validation accuracy (the "Intermediate Value" indicated in the y-axis). It can be seen that the best trials led to a validation accuracy between 0.7 and 0.75, but the convergence was not smooth for most trials. In some cases, the performance deteriorated when the step number increased.

Also, Figure 6 shows that the best trial was number 3, which corresponds to the fourth one (the third darker red dot). Note that said Optimization History Plot shows the best value obtained in the trials of an entire study. The "Best Value" is plotted as the value the other trials must surpass in order for it to gain a higher "position". If the other succeeding trials do not surpass the one with the best value, in this case number 3, it remains constant and equal to the last best value (red line), while plotting the objective values (blue dots) for all trials until one is able to have a better outcome than the last, or the study is finished.

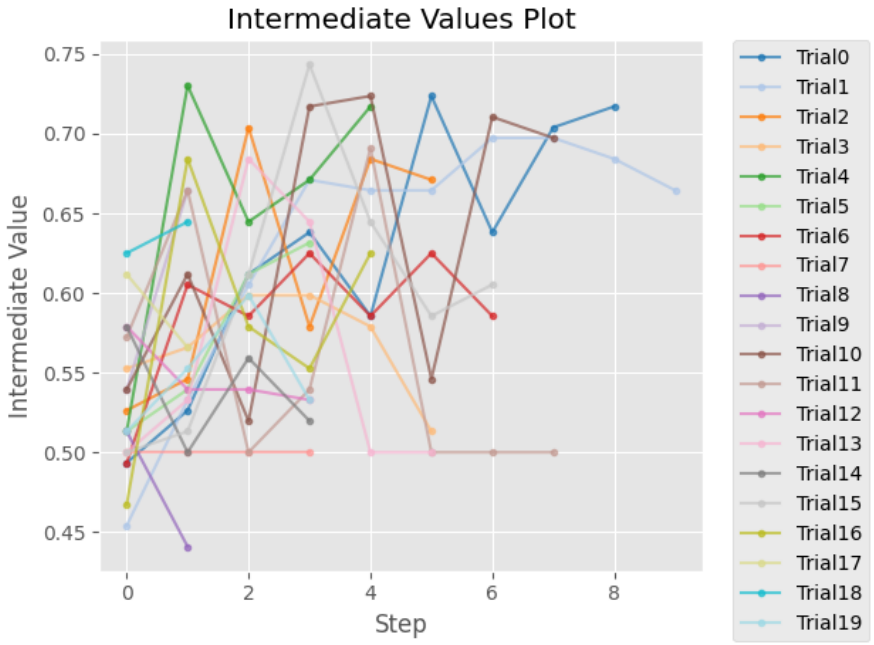
Figure 7 shows the hyperparameter importance plot for the initial experiments using the validation accuracy as the monitored metric. It can be seen that Optuna estimated that the "activation function" was the most important hyperparameter, followed by the kernel size for the first convolutional layer. Dropout and mini-batch size were also found as important as this kernel size. Both validation accuracy and validation AUC were used as the optimization metric, but Figure 7 indicates that changing this metric did not lead to significant variation in the results.

4.1.2 Second Initial Model Selection

The second model selection was important because it refined the hyperparameter search, using narrow ranges for Optuna optimization. The results in this second model selection stage provided a comparison between what would be considered a good result and a very bad one. This second initial model selection differs from the first in the fact that the first allowed Optuna to test a myriad of different parameters, while the second limits Optuna more in relation to the search space.

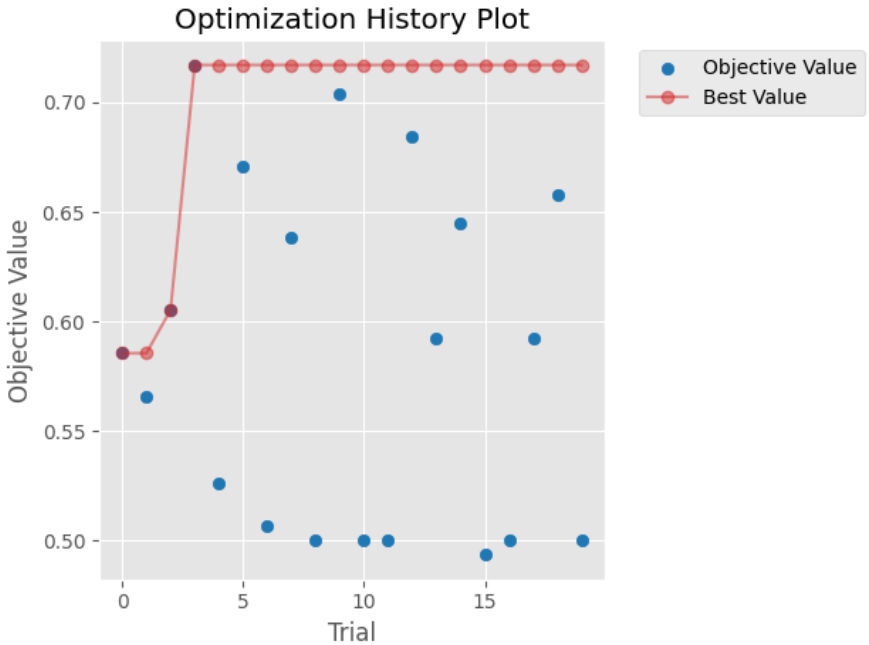
One can note from Figure 8 that pruning, along with the early stopping callback, can abort some trials prematurely (at early steps). Figure 9 shows that the first trial (number 0)

Figure 5 – Intermediate Values Plot for the first model in the initial experiments using the validation accuracy as the monitored metric.



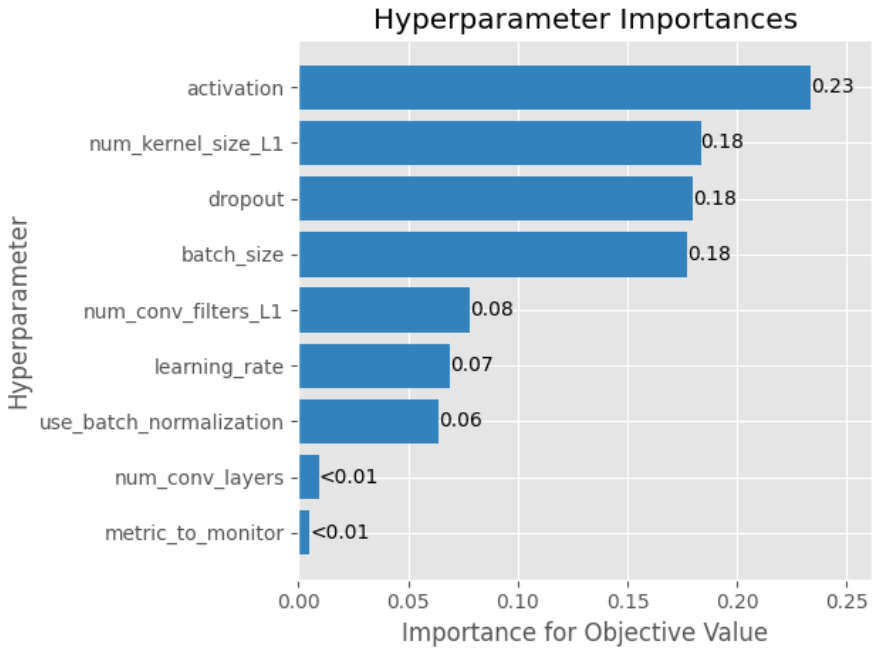
Source: author.

Figure 6 – Optimization History Plot for the first model in the initial experiments using the validation accuracy as the monitored metric.



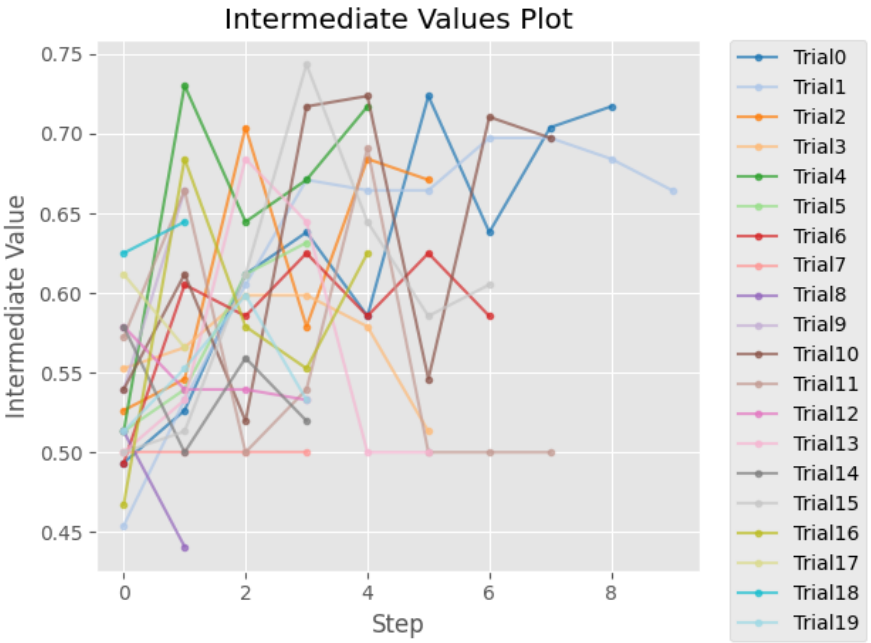
Source: author.

Figure 7 – Hyperparameter Importances Plot for the first model in the initial experiments using the validation accuracy as the monitored metric.



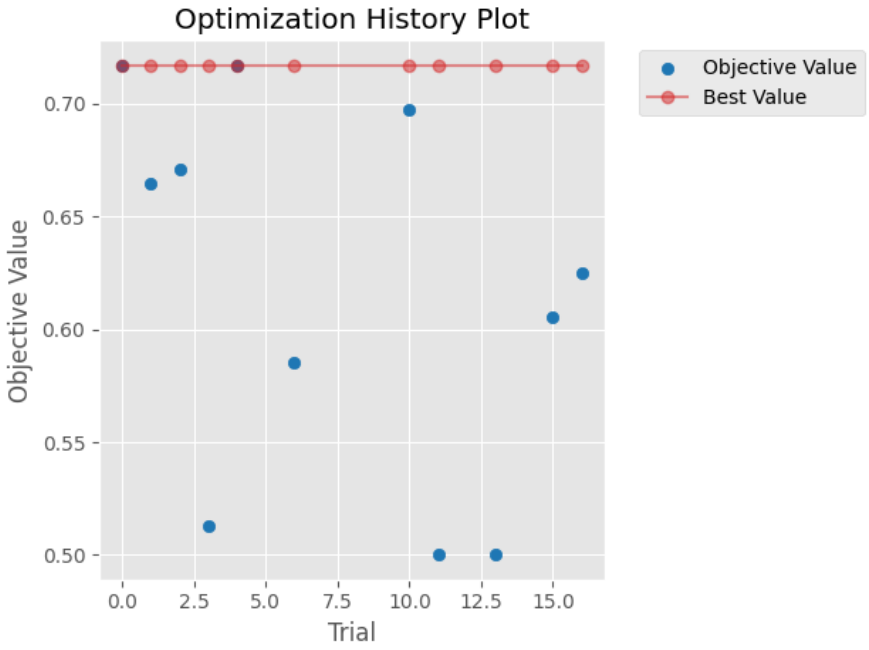
Source: author.

Figure 8 – Intermediate Values Plot for the second model in the initial experiments using the validation accuracy as the monitored metric.



Source: author.

Figure 9 – Optimization History Plot for the second model in the initial experiments using the validation accuracy as the monitored metric.



Source: author.

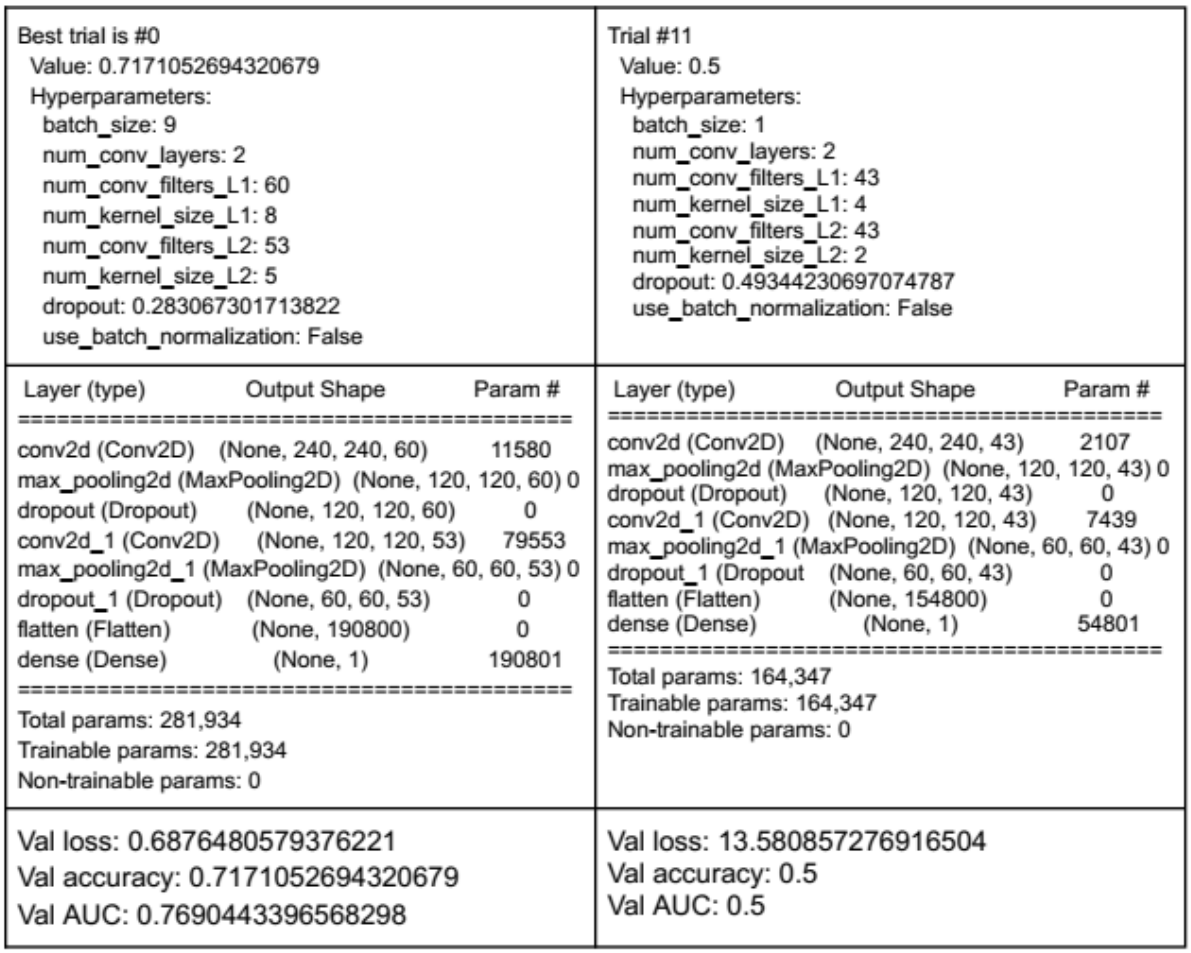
achieved the best value in the whole study. It is useful to observe in Figure 10 the difficulties of convergence of different Optuna trials. Here, trial 0 and trial 11 will be further compared. It can be seen in Figure 9 that trial 11 reached a validation accuracy of 0.5. Considering the problem is binary classification, 0.5 is the worst possible result because if smaller, one could invert the label and get above 0.5 accuracy.

Figure 10 is relatively complex but provides readers familiar with Keras a complete description of the models that were adopted. The column “Output Shape” shows the dimension of the tensor at the output of the respective layer. Because it is a binary problem, a single neuron composes the last layer. This figure also shows how hard is to work with the adopted dataset, which presents a severe variance in results. Comparing the hyperparameters in trials 0 and 11, one can see that they are not very different. For instance, trial 0 led to a neural network model with 281934 parameters with trial 11 had a model with 164347 parameters. Besides, both models had two convolutional layers, but the relatively small and difficult dataset leads to a significant discrepancy between the results of the two trials. Figure 11 completes the evaluation of the second model selection study, indicating the importance of each hyperparameter. In this case, the kernel size of the first layer was found significant.

4.1.3 Third Initial Model Selection

Here, the NN topology was kept fixed according to the best choices for the previous model selection, as can be seen in Table 8. In this case, fewer parameters were exhaustively

Figure 10 – Comparing the best result for the second model selection with a bad one.



Source: author.

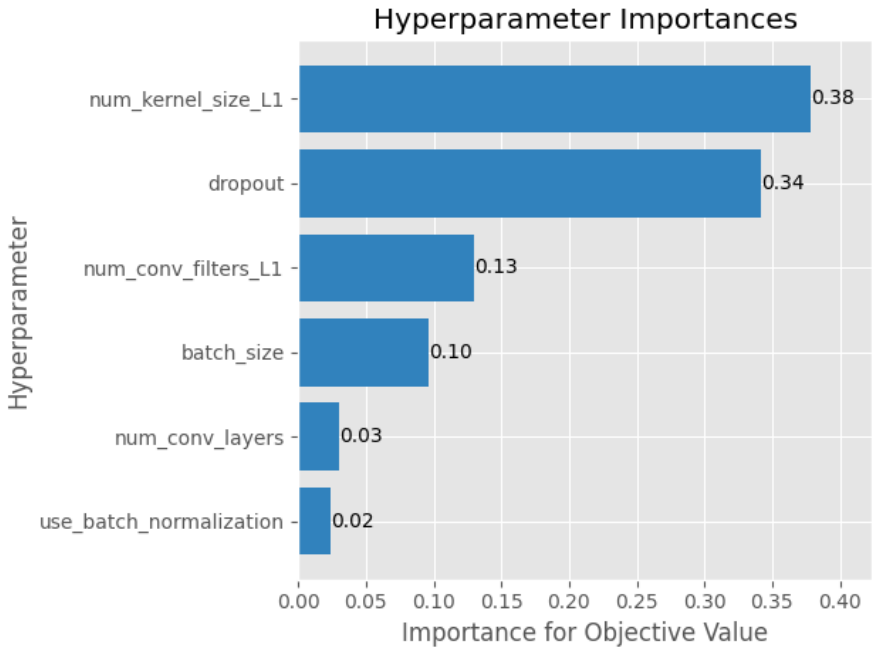
varied, according to their importance in previous model selection studies. The best trial was number 14, with a value of 0.75657, and the best hyperparameters for it along with the ones Optuna was allowed to explore are described in Table 9.

Table 8 – Fixed hyperparameters for the third initial model selection study.

Hyperparameter	Value
num_conv_layers	2
num_conv_filters_L1	60
num_kernel_size_L1	8
num_conv_filters_L2	53
num_kernel_size_L2	5

One can observe that the validation accuracy increased to 0.757, and the best dropout rate is high: 0.45, which indicates the importance of avoiding overfitting to the training set in this difficult dataset. The batch size chosen by Optuna was 12, indicating that it was beneficial to collect 12 examples to calculate a single gradient for the gradient descent algorithm. One issue

Figure 11 – Hyperparameter Importances Plot for the second model in the initial experiments using the validation accuracy as the monitored metric.



Source: author.

Table 9 – Hyperparameters for the best trial (number 14) in the third initial model selection study.

Hyperparameter	Value
batch_size	12
dropout	0.45736
batch_nor	False
activation	swish

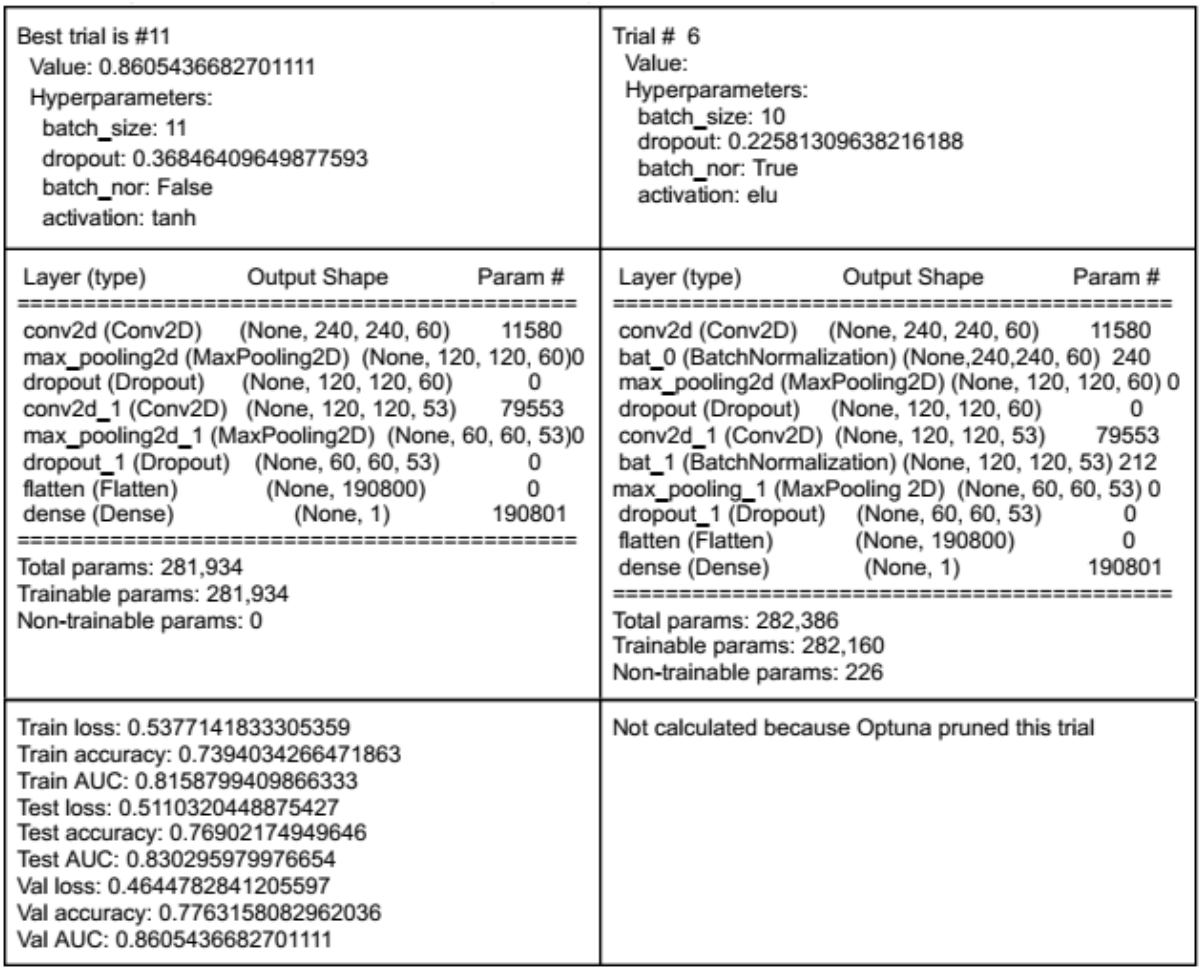
to pay attention to in imbalanced datasets is that a single batch may not have positive examples, only negative ones. This is probably not happening here because the used dataset is a balanced subset of the original data, but it is something to keep in mind.

4.1.4 Fourth Initial Model Selection

After observing that validation accuracy varies a lot due to relatively small datasets and the large diversity of distinct examples, that is to say, the task is hard for the model, the fourth model selection experiment tried to use the AUC as the metric to guide Optuna and Keras. AUC was used as the "Objective Value" for Optuna and the Keras callback methods used AUC in order to save the best model, reduce the learning rate, etc.

Figures 12 and 13 compare the results of the best trial with the ones from two other trials. Figure 12 compares the best trial (number 11) with a trial that was pruned by Optuna. The models are relatively similar, for instance, having approximately the same number of parameters.

Figure 12 – Comparing the best model of the fourth initial model selection with a pruned one (trial 6).



Source: author.

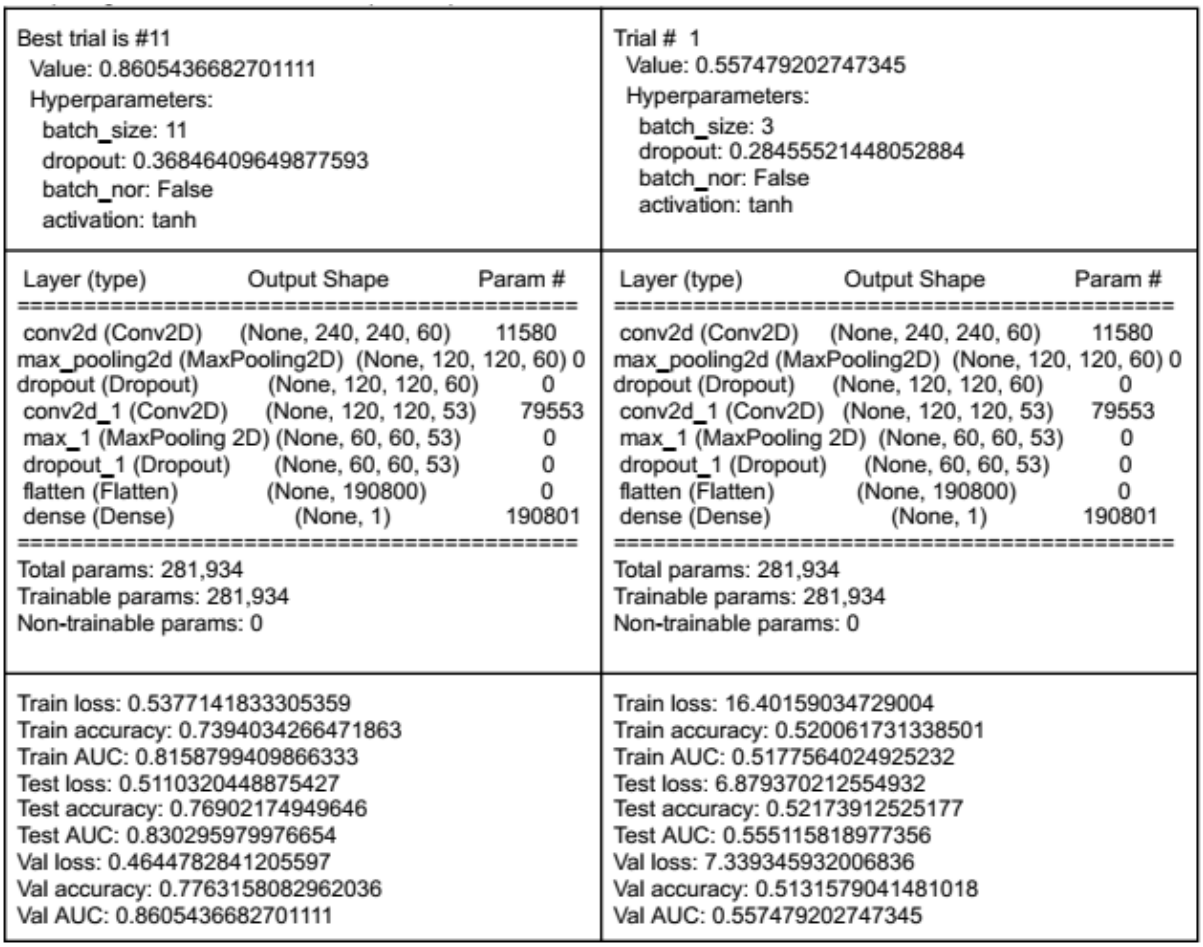
However, due to the variance caused by random initialization of the weights and other non-deterministic aspects of the training procedure, trial number 6 was prematurely aborted in its initial epochs.

On the other hand, Figure 13 compares trial 11 with trial 1, which led to a bad result but was completed. Trial 1 achieved a validation AUC of 0.557 while trial 11 reached 0.86. Figure 13 also indicates the loss (binary entropy), accuracy, and AUC for the three sets: training, test, and validation. This is a sanity check that allows us to observe that the performance is similar for the validation and test sets. For instance, the test accuracy in trial 11 was 0.769 while the validation accuracy was 0.776. Having similar performances on the validation and test sets is essential for the training procedure.

4.1.5 Discrepancies Between Last and Best Models

Unless programmed otherwise, Optuna takes into account the performance of the last model in a trial. However, with models presenting erratic convergence, this may be different than

Figure 13 – Comparing the best model of the fourth initial model selection with a bad one (trial 1).



Source: author.

the best model. One can observe that the results vary sometimes significantly. Another aspect to consider is that all positive examples of the dataset are being always used, but the negative examples are being randomly selected to compose balanced datasets in this experiment. Hence, the Panda dataframes (obtained from the datasets) used in this experiment are distinct from the ones used in the training. Therefore, the "best" is not always better than the "last" model, as one can see in Figure 14.

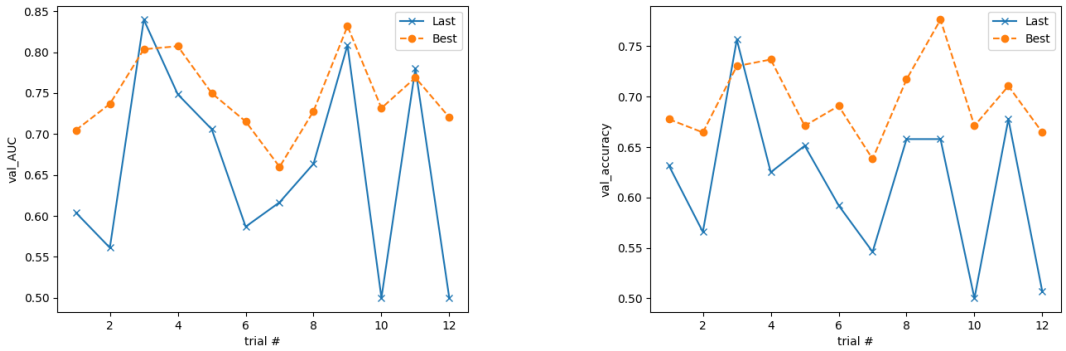
4.1.6 Conclusions Regarding the Initial Model Selection

The AUC seems better than accuracy with respect to leading to a smooth curve but leads to more discrepancy between the best and last NN models. Hence, in certain cases, accuracy seems a safer alternative. Also, several other things could be tested. For instance, the MaxPooling was fixed in 2 during these experiments.

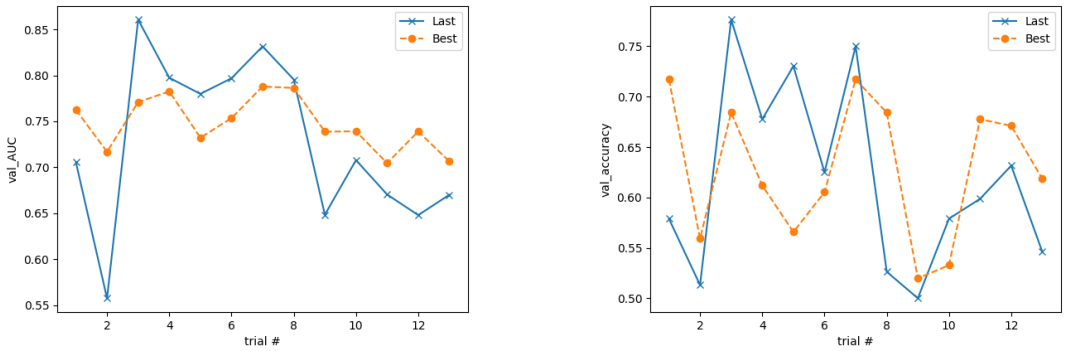
About the NN topology and hyperparameters, through experimentation the following were found to be sensible choices: a high dropout rate, no batch normalization, a batch size of more than 10 examples, and a specific number of convolutional layers, filters, and kernel sizes

Figure 14 – Plots comparing the best and last models in the third and fourth experiments, according to the metrics used to monitor them, the validation accuracy (val_accuracy) and the validation AUC (val_auc).

(a) Validation AUC for the third model. **(b) Validation accuracy for the third model.**



(c) Validation AUC for the fourth model. **(d) Validation accuracy for the fourth model.**



Source: author.

for layers L1 and L2 that is described in Table 10. In total, there were approximately 200000 parameters.

Table 10 – Fixed hyperparameters for the conclusions regarding the first initial model.

Hyperparameter	Value
num_conv_layers	2
num_conv_filters_L1	60
num_kernel_size_L1	8
num_conv_filters_L2	53
num_kernel_size_L2	5

4.2 Studying the Sample Complexity of Unbalanced Sets

Here the impact of the number of training examples on performance, which is called *sample complexity* (BARTLETT, 1998), will be studied. Based on the model selection experi-

ments conducted in Section 4.1, the following model and hyperparameters that were chosen for this next stage can be seen in Table 11.

Table 11 – Fixed hyperparameters to study the sample complexity of the data.

Hyperparameter	Value
batch_size	12
num_conv_layers	2
num_conv_filters_L1	60
num_kernel_size_L1	8
num_conv_filters_L2	53
num_kernel_size_L2	5
dropout	0.4
use_batch_normalization	False
activation	swish
learning_rate	0.0005
metric_to_monitor	val_accuracy

Callbacks were adopted, such as EarlyStopping, and ReduceLROnPlateau to reduce the learning rate and another one to save the best model, with Tensorboard also being used to monitor the model training. The EarlyStopping patience was made 10 to promote a larger number of steps. Because the best model is kept, it is not a problem if the last model diverges too much from the best (after 10 epochs as designed by the patience).

A batch file was created to execute in sequence the same code but gradually increase the number of training examples from 30 to the maximum of 18551, which would be by using the entire training dataset, with its 18227 negative samples and 324 positive samples. As before, the validation and test sets are always kept balanced with 76 and 184 positive examples, respectively. Hence, the total number of examples in each of these sets is 152 and 368 examples.

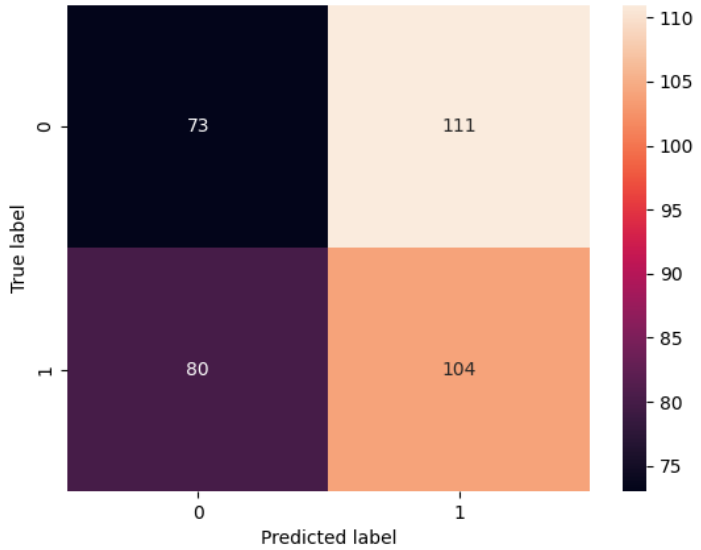
The total number of positive examples in the original training set is 324. Whenever the subset created for training has a number not larger than 648 examples, it is kept balanced. For instance, if the number N of training examples is 200, then 100 are positive and 100 are negative. If N is larger than 648, then 324 are positive examples, and N-324 negative examples are randomly chosen until completing N examples.

The results in Figure 15 indicate some overfitting, although there is no issue with unbalanced sets. Another issue might be the variance, with the model not providing a smooth curve for training and validation accuracy. The bias versus variance trade-off (SINGH, 2018) is well-known in ML and in this case, it can be seen by the relatively strong impact that small changes of parameters and dataset examples cause in performance.

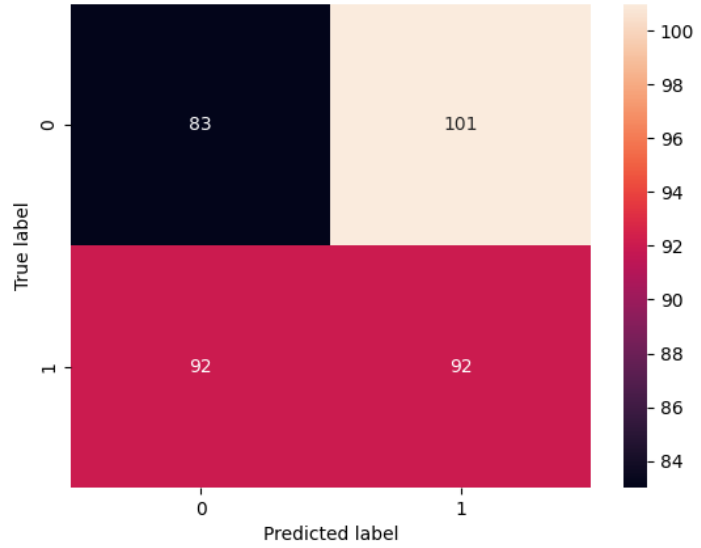
For instance, along the simulations, the test and validation sets are held constants, but the examples that compose the training set are randomly chosen. In large and well-behaved (small variance) ML problems, choosing examples randomly does not impact the results too much, but

Figure 15 – Confusion matrices for training with two balanced data subsets, with 150 and 648 samples.

(a) N=150 examples (75 of each class).



(b) N=648 examples (324 of each class).



Source: author.

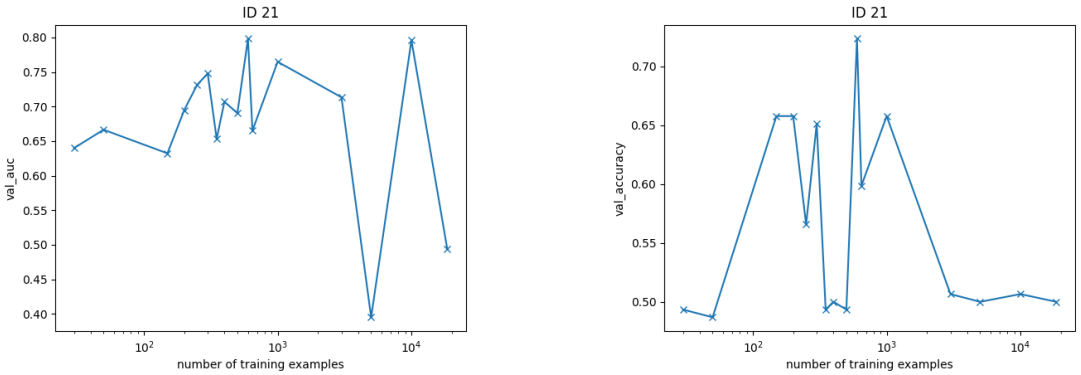
this is not the case here.

Study number 21 was created in order to evaluate the sample complexity using 16 distinct training examples. More specifically, the following number N of training examples were used for training the respective models: 30, 50, 150, 200, 250, 300, 350, 400, 500, 600, 648, 1000, 3000, 5000, 10000, 18551. As can be seen in Figure 16 (the x-axis is in logarithmic scale), the val_auc and val_accuracy and results varied significantly with the number N of training examples. For instance, when $N = 1000$, val_accuracy is approximately 0.65. But using $N = 3000$ decreased val_accuracy to approximately 0.5, probably because the class unbalancing was too extreme in the latter case.

The models trained with N=1000 and N=18551 examples in Figure 16, contain training data that is unbalanced in both cases. Nevertheless, given that the maximum number of positive examples is 324 when using N=1000, there were 1,000-324 = 676 negative examples, while for N=18551, there were 18227 negative examples.

Figure 16 – Sample Complexity Curves.

(a) Area Under the Curve. **(b) Validation accuracy.**



Source: author.

4.3 Model Selection with Pre-calculated Backend Outputs

This section presents certain studies that were deemed the most relevant due to their results and thus selected. Therefore, the number of each trial may vary.

When a large backend neural network model is used as a feature extractor, the goal is often to train only the layers responsible for the classification stage. However, during training and inference, after an image is fed to the backend input, all layers of the backend need to be processed until getting the features that feed the classification layers. This feature extraction can correspond to a huge computation when the backend is large, with millions of parameters.

An idea that occurred during the execution of this work was to pre-calculate the features, such that the backend did not need to be invoked, as explained in Section 3.2.2. Hence, a script was created that runs the EffNet backend for feature extraction and converts all input images into feature vectors, saving them and using them as input in the classifier. After that, the EffNet backend is not used anymore, and the training is faster.

This procedure involves using the three datasets for training, validation, and testing and generating a Python Pickle file for each of them that contains said feature vectors for the images. The total number of images in the three sets was 648, 368, and 152, respectively, and the number of features for each of the images in the three sets was 1280, as pre-defined by the EffNet model.

Using these saved feature vectors from EffNet, it was possible to significantly cut down on training and pre-processing time. When the model selection was done solely with the original

script which processed the backend layers each time, calculating each trial in Optuna took 3.7 minutes. Using the pre-calculated features reduced the time per trial to only 0.09 minutes. This corresponds to a speed factor of $3.7/0.09 \approx 41$, which enabled several simulations that would be unfeasible when using the EffNet backend for feature extraction during every training instance.

For instance, Study number 26 used 3000 Optuna trials with a maximum number of epochs increased from 10 to 100. The whole study took approximately 500 trials per hour or 0.118 minutes per trial, with the best results being trial 72 with a value of approximately 0.8026 for the validation accuracy. It had 119227 parameters, and its complete hyperparameters are shown in Table 12 while the entire results are Table 13.

Table 12 – Hyperparameters for the best trial (number 72) in Study 26.

Hyperparameter	Value
batch_size	2
num_dense_layers	1
num_neurons_L1	93
dropout	0.69174
regul	True
l1_weight	0.0001
l2_weight	0.0001
activation	swish

Table 13 – Results of the metrics for val, train and test sets in trial 72 of Study 26.

Metric	Result
Train loss	0.51918
Train accuracy	0.82098
Train AUC	0.90205
Test loss	0.61893
Test accuracy	0.73641
Test AUC	0.83087
Validation loss	0.57845
Validation accuracy	0.80263
Validation AUC	0.86400

The results are similar to those of previous studies: the network is relatively small, with a single dense (fully connected) layer between the EffNet extractor and the final neuron. The EffNet does not show up in the description of the model, given that the input sets used the feature vectors with 1280 values. But the number of weights (parameters) in the first layer provides a hint about the input dimension given that $(1280+1)93 = 119133$ weights, indicating that for this fully connected layer, each of the 1280 features, plus 1 for the bias, are connected to each of the 93 neurons of the first dense layer.

4.4 Unbalanced Datasets and Training with Class Weights

Weighing the gradient parcel differently so that the positive class has a higher weight yielded improved results. Study 32, shown in Table 14, used a training dataset with N=3000 examples, and the features are obtained via the EffNet pre-trained model.

Table 14 – Hyperparameters for the best trial (number 155) in Study 32.

Hyperparameter	Value
activation	elu
dropout	0.63135
lea_rate	0.00218
neurons_L1	294

The validation accuracy in Study 32 is equal to approximately 0.8289 and it is the best obtained up to now. This indicates that the larger number of examples (N=3000) and the weighting method helped to train the model. Also, Study 32 used the best parameters of previous studies. Meanwhile, Study 33 used a large number of Optuna hyperparameters, such that a new chance could be given to new topologies, with a larger number of layers, but did not achieve a better result than Study 32. Figure 17 shows the results of these two studies using the validation accuracy as the objective value (monitored metric).

The dense layers could vary from 1 to 4 in Optuna, but Optuna found 2 layers as the best option. This is an increase with respect to the previous one, which chose 1 layer. Maybe using N=3000 allowed a larger NN model to still be robust against overfitting, or perhaps the result was just due to the high variance in this problem.

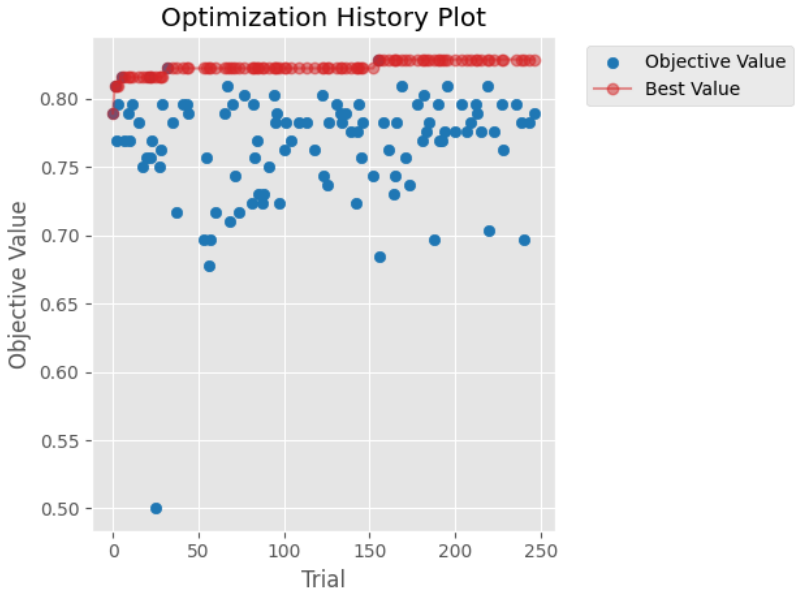
4.5 Data Augmentation

This study used positive examples created with data augmentation. Based on the examples created using the C1 dataset of augmented images described in Table 6, another set of three Pickle files with extracted features was created with the smaller backend model, using N=10000 training examples. Note that in this case, Study 35, N=10000 train examples were used, with 5000 positives and 5000 negatives. That is, when using augmentation, one can use balanced datasets. The negative examples are the original ones, while the positives are augmented. The best model was from trial 559, shown below in Table 15, which achieved a value of 0.7894 for the validation accuracy.

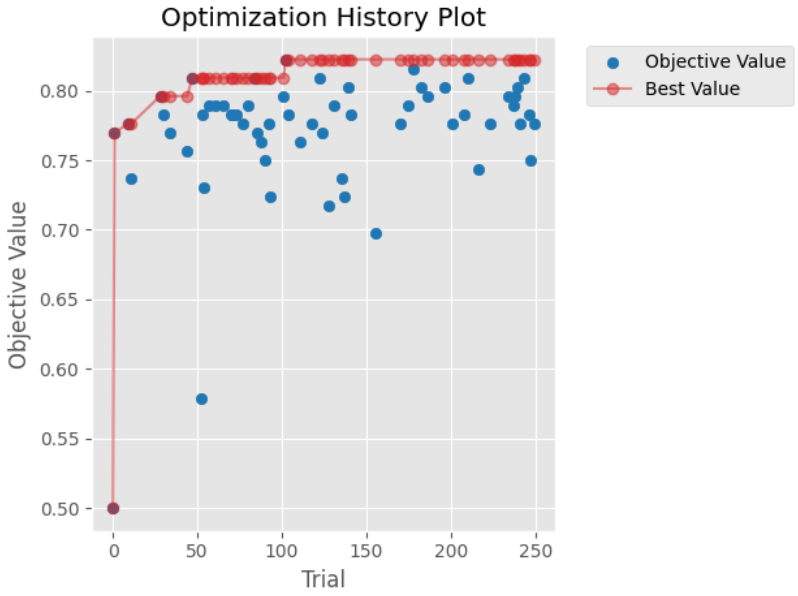
L1 regularization was adopted to try to further combat overfitting, and the dropout chosen by Optuna was relatively high (0.59). Note that 3 dense layers, besides the output one (which completes 4 layers) were chosen. From the Optuna log file, it is possible to then tell that the model only had 87288 parameters, being relatively small.

Figure 17 – Optimization History Plots for two studies (32 and 33) using class weights to deal with imbalanced datasets.

(a) Study 32 with weights.



(b) Study 33 with weights.



Source: author.

Fig. 18 shows some of the results for Study 35. One can see the clear importance given to dropout and batch size since they are the ones with the highest values of importance for the objective values analyzed. This means that Optuna noted that changing these hyperparameters caused the most repercussions, so they became more relevant.

In the previous study (Study 35), the smaller EffNet model with 7 million parameters was used. A new Study, number 39, was done with a larger EffNet model, which has approxi-

Table 15 – Hyperparameters for the best trial (number 559) in Study 35.

Hyperparameter	Value
activation	elu
batch_nor	False
batch_size	9
dropout	0.59393
l1_weight	0.01
l2_weight	0
lea_rate	1.37264e-05
neurons_L1	67
neurons_L2	17
neurons_L3	16
num_dense_layers	3
regul	True

mately 200 million parameters. Study 39 still used N=10000 examples with augmented positive examples. Instead of using the C1 data augmentation dataset used in Study 35, the new Study 39 used the C3 data augmentation dataset for training, described in Table 6. It is important to recall that the data augmentation was only applied to the training set, as there was no need to augment the test and validation sets were kept balanced, as seen in Table 7.

The Optuna parameters to be optimized changed a bit in 39 with respect to 35. Some were kept fixed, such as the dropout probability. The result for Study 39 was 0.75657 and the hyperparameters can be seen in Table 16. The results in Study 39 were not better than the previous ones.

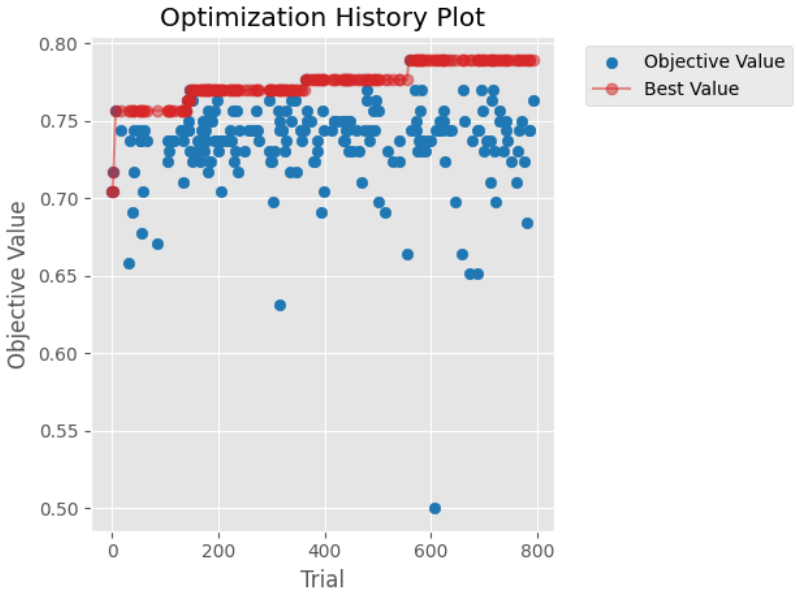
Table 16 – Hyperparameters for the best trial (number 538) in Study 39.

Hyperparameter	Value
batch_nor	True
lea_rate	1.62805e-05
neurons_L1	251
neurons_L2	248
neurons_L3	66
neurons_L4	18
neurons_L5	8
num_dense_layers	5
regul	False

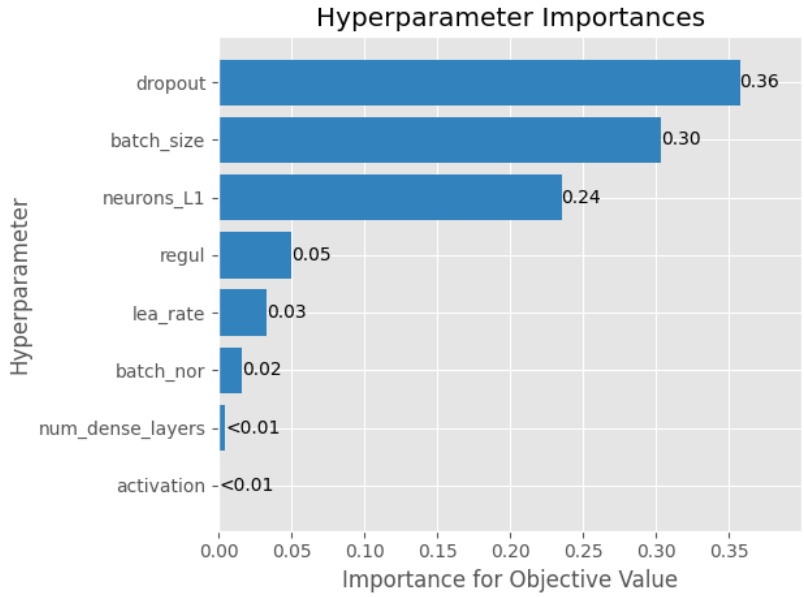
Study 40 was then designed with a fixed number of 1 hidden layer, to observe the impact of varying its number of neurons. This network had 3 M parameters, and only 4 epochs were executed due to overfitting. Fig. 19 presents some of the results, and its best trial was number 38, described in Table 17. Meanwhile, the overall metrics results with the best model of Study 40 are shown in Table 18.

Figure 18 – Optimization History and Parameter Importances Plots for Study 35 with data augmentation.

(a) Optimization history of study 35.



(b) List of hyperparameter importance for study 35.



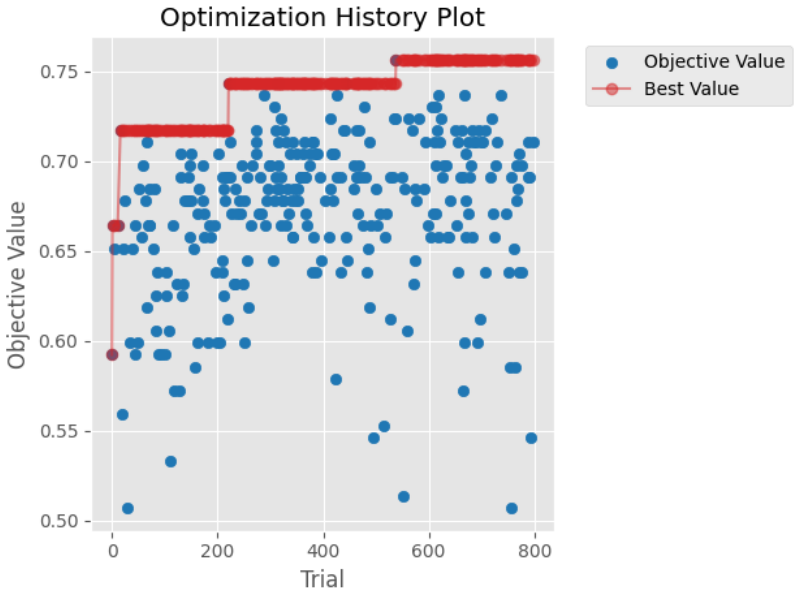
Source: author.

Study 41 was designed to show sensible contour plots displaying the relation between two hyperparameter choices, in this case, the dropout rate and the number of neurons on the L1 layer. The objective value of the best trial (number 1) was 0.73684 for the validation accuracy, and the dropout rate was 0.68662, while the number of neurons in L1 was 1137. It differed from Study 40 only in not having a fixed value for the dropout and not applying regularization.

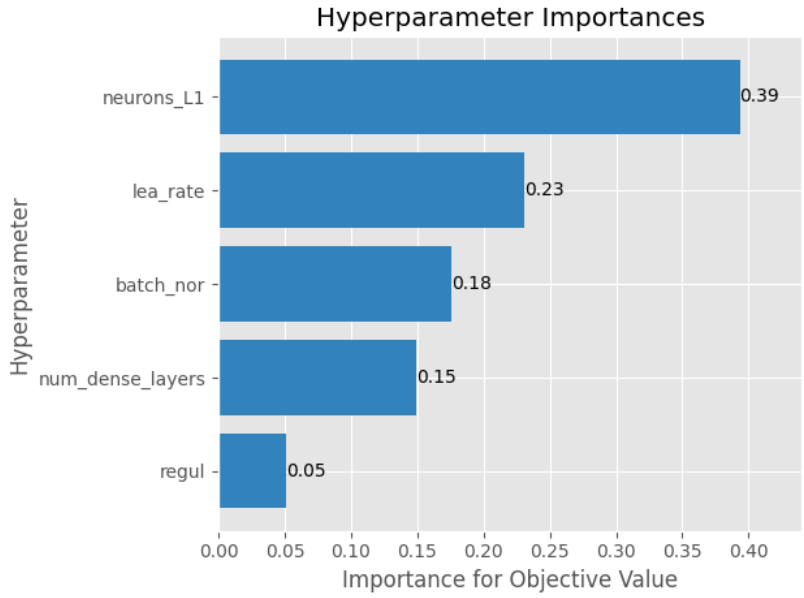
It can be seen from the Contour Plot in Figure 20 that good results (around 0.70) are also

Figure 19 – Optimization History and Parameter Importances Plots for Study 40 with data augmentation.

(a) Optimization history of Study 40.



(b) Parameter Importance for Study 40.



Source: author.

achieved with less than 500 neurons, but the best one was achieved with 1137 neurons. Another thing that is clear is that the best dropout is large, with a value of 0.68. The contour plot clearly shows that small dropout values are in a lighter blue (lower values of the validation accuracy metric) than larger dropout values.

Table 19 shows the overview of the main characteristics for the most relevant Studies using Optuna and the EffNet as the backbone, noting that for the experiments in said table two

Table 17 – Hyperparameters for the best trial (number 38) in Study 40.

Hyperparameter	Value
l1_weight	0
l2_weight	1e-05
neurons_L1	2497

Table 18 – Results of the metrics for validation, train and test sets in trial 38 of Study 40.

Metric	Result
Train loss	0.20984
Train accuracy	0.93339
Train AUC	0.98458
Test loss	0.95554
Test accuracy	0.71195
Test AUC	0.79135
Validation loss	1.0134
Validation accuracy	0.71710
Validation AUC	0.80618

different EffNet models are used, the one called *EffNet-1k* is the one with fewer parameters, around 7 million, which used images with the input size of 240x240 pixels and was trained with the ImageNet1k database, which contains around 1000 classes. On the other hand, the model called *EffNet-21k* is the one with more parameters, around 200 million. It used input images of 512x512 pixels and was trained on the ImageNet21k, a database that has over 21000 classes.

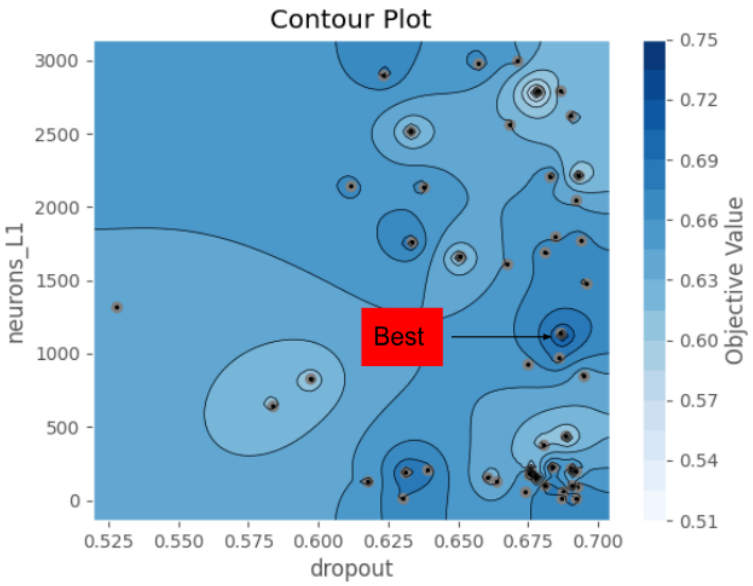
Table 19 – Overview of the characteristics of each main Study.

Study	Best trial	Metric monitor.	Backbone	# of images 0 / 1	Class Weights	Data augm.	# Param.
26	72	Val_acc	EffNet-1k	324 / 324	No	No	119k
32	155	Val_acc	EffNet-21k	324 / 2676	Yes	No	337k
35	559	Val_acc	EffNet-1k	5k / 5k	No	Yes (C1)	87k
39	538	Val_acc	EffNet-21k	5k / 5k	No	Yes (C3)	403k
40	38	Val_acc	EffNet-21k	5k / 5k	No	Yes (C3)	3.2M
41	1	Val_acc	EffNet-21k	5k / 5k	No	Yes (C3)	1.4M

Moreover, Table 20 shows the accuracy and AUC for the validation and test sets, approximated to four decimal places. Additionally, Figure 21 shows the metrics results for the six different Studies analyzed. In it, it is possible to see that Study 32, which used class weights and no data augmentation, yielded the best results, with an AUC value of 0.8698 and an accuracy value of 0.8125 on the test set.

In summary, when all studies are taken into account, it can be seen that the variance of the results is significant. Besides a relatively large and imbalanced dataset, this work also dealt

Figure 20 – Best model in contour plot between the number of neurons in layer 1 (1137 neurons) versus the dropout rate (0.687) in Study 41.



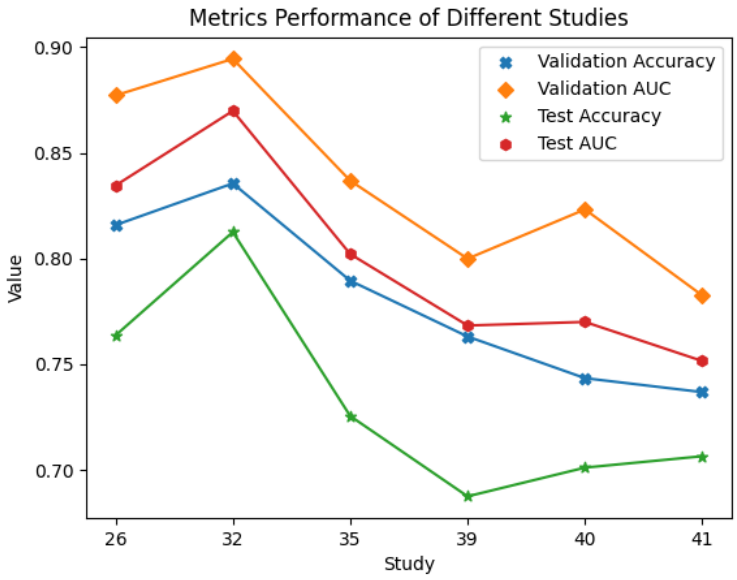
Source: author.

Table 20 – Overview of the metrics results in each main Study, according to the validation and test accuracy and AUC.

Study	Best trial	Val_acc	Val_AUC	Test_acc	Test_AUC
26	72	0.8158	0.8771	0.7636	0.8345
32	**155**	**0.8355**	**0.8944**	**0.8125**	**0.8698**
35	559	0.7895	0.8368	0.7255	0.8021
39	538	0.7631	0.7999	0.6875	0.7683
40	38	0.7434	0.8232	0.7011	0.7700
41	1	0.7368	0.7825	0.7065	0.7515

with a large variety of results. In the case of Study 40, for instance, a model with 3 M parameters was selected by Optuna, but models with a smaller number of parameters led to similar results during the Optuna optimization. On the other hand, dealing with a difficult problem requires investigating all remedies to combat overfitting and other issues, and this dataset was ideal to reinforce important concepts in ML applied to medical images.

Figure 21 – Metrics results for the accuracy and AUC for the Studies analyzed.



Source: author.

5 CONCLUSION AND FUTURE WORK

Image-based AI classification for skin cancer is very important to our society, especially because it can substantially help early detection. In spite of several advances during the last few years, this area still needs further research. The current state-of-art achieves excellent results when the datasets are well-organized and relatively homogeneous with respect to the images, but when images vary a lot, the ML models face difficulty to achieve a good generalization capability. Hence, new methods are still needed until ML-based models for skin cancer detection, especially neural networks, become widely used in everyday medical practice.

This work adopted a database that contains approximately 33000 varied dermoscopic images. A key aspect is that this dataset is known for having images with many artifacts, and were taken in a variety of conditions. In other words, the used dataset is not considered of high quality and the results are impacted by that. Another important factor is the difference in the quantity of images that actually positive for melanoma and images that are negative, with approximately only 1.8% of the samples in the entire dataset being positive. This combination of reasons makes using the dataset a hard task, with the need to overcome overfitting, unbalanced data, and variance, but one that highlights the impact of these difficulties.

This work aimed to contribute to the medical image AI community by approaching a relevant topic of study and providing the code used to develop a pipeline that is optimized for cost-efficiency and uses modern state-of-the-art Python libraries such as TensorFlow, Keras, and Optuna, along with techniques like data augmentation and weighting of classes. The code can be applied to other datasets of medical images, or other distinct tasks. Also, it studied the different effects of methodologies used to reduce the disparities of the classes, in order to compare what strategies yield the best results for classification between a non-melanoma lesion and a melanoma.

The main lessons taken from the several experiments are the following:

a) Having a library such as Optuna for optimizing the hyperparameters is essential when the training procedure has many degrees of freedom. The optimizer may not achieve a good result if a vast number of possibilities is considered, but it enables a semi-automatic procedure, in which the user learns from results such as contour plots and interacts with the optimizer (Optuna in this case) to narrow the hyperparameter search space.

b) Given the small number of positive examples, training was always difficult due to overfitting. Some of the strategies to circumvent it were: large values of dropout probability, relatively simple models (small number of parameters), regularization (L1 and L2), few training steps, and a relatively small learning rate. Among them, the number of model parameters seemed to be the easiest to control.

c) The augmentation adopted in this work helped the training procedure but was not enough

to produce a large improvement in performance. Better augmentation techniques could be eventually designed.

d) Weighting differently the examples of each class in order to give more weight to the rare positive examples also helped training, but similarly to the augmentation results, it did not produce large improvements.

e) Besides having imbalanced classes, the dataset also led to high variance in performance. This means that relatively small changes in the training procedure could produce strong effects on the model performance.

f) It is discussed in the literature that the AUC of the ROC or Precision-Recall curves may lead to better results than the validation accuracy for such binary problems. However, the experiments conducted in this work did not always confirm this observation. The ROC AUC would eventually improve performance in some experiments, but fall behind accuracy for others, with the best result having been one where the validation accuracy was monitored (Study 32). This was probably a consequence of the variance in the results.

g) The simulations showed the importance of pre-computing and saving the features generated by a backend model instead of always invoking the backend. This led to reducing the simulation time by a factor of approximately 40 for the adopted backend models. Such method is essential when one does not have access to many fast computers such as in this work.

h) This work also emphasized the importance of adopting a rigorous methodology for evaluating the models, which, in this scenario of a large variance of results, proved to be extremely important. In this work, all hyperparameter search was conducted using the validation set, never the test set. In case the test set was checked, the accuracy could be improved to more than 80% (in the test set). For instance, consider that a model led to an accuracy of 78% and 82% in the validation and test sets, respectively, and another model led to 81% and 79%. According to the adopted methodology, the chosen model would be the second, and its performance of 79% of accuracy (in the test set) is the number representing its generalization capability. If the test set is repeatedly used during the hyperparameter search, the estimated generalization capability would be overly optimistic (82% in this example).

There are several research groups worldwide working with the same dataset, and the state-of-art results are better than the ones presented here, such as the work done by (HA et al., 2020). Some of the reasons are that this work looked only at the images, and the derived models did not benefit from the metadata that is available. Another aspect is that some solutions used for this dataset are based on an ensemble of models, where many models are trained and the final decision is based on their votes. Because the goal of this work was to learn and apply modern

tools, and not to maximize accuracy, ensembles were not adopted due to their relatively high computational cost.

This work was positioned as a preliminary study on a difficult task and, naturally, there are several possible improvements. One direction is to improve data augmentation, using more sophisticated techniques, tailored to skin cancer. This work adopted the generic image transformations that are already implemented in advanced libraries such as Albumentations, but better results can be eventually obtained by methods that take into account the characteristics of skin cancer and create sensible positive examples. Using synthetic data to improve the training set of ML models is an intense area of research. Generative AI, with models such as Generative Adversarial Networks (GANs) is something that has been used also in medical images.

Another possible improvement is to adapt the code to run on the cloud using paid services, such as Amazon's AWS (AWS, 2023). This is important if the goal is performance. It was not the case in this study, but having access to a high-quality Graphics Processing Unit (GPU) or several allows us to quickly test new ideas and train large models using a very large number of examples. Also, a possible other option might be to run multiple studies simultaneously.

In another direction, the code can be changed to improve documentation and readability. It can be used by students and practitioners of ML, as a complete example of what is nowadays available from tools such as Optuna. One specific aspect is that, because this work posed the problem of binary classification, the code allows us to observe several concepts that are associated with this case of having only two classes. For instance, the code provides examples of how to derive the precision-recall curves, which may not be studied if one focuses only on the accuracy of non-binary problems.

BIBLIOGRAPHY

AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: **Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2019.

ALAM, T. M. et al. An efficient deep learning-based skin cancer classifier for an imbalanced dataset. **Diagnostics**, MDPI AG, v. 12, n. 9, p. 2115, Aug 2022. ISSN 2075-4418. Disponível em: <http://dx.doi.org/10.3390/diagnostics12092115>.

ALWAKID, G. et al. Melanoma detection using deep learning-based classifications. In: MDPI. **Healthcare**. [S.l.], 2022. v. 10, n. 12, p. 2481.

American Cancer Society. **Skin Cancer Quiz**. 2023. Accessed: June 13, 2023. Disponível em: <https://www.cancer.org/cancer/types/skin-cancer/skin-cancer-quiz.html>.

AMRANE, M. et al. Breast cancer classification using machine learning. In: **2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)**. [S.l.: s.n.], 2018. p. 1–4.

AWS. **Aprendizado profundo na AWS**. Filbert Pub., 2023. Disponível em: <https://aws.amazon.com/pt/deep-learning/>.

BARTLETT, P. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. **IEEE Transactions on Information Theory**, v. 44, n. 2, p. 525–536, 1998.

BHANDARI, A. **Guide to AUC ROC curve in machine learning: What is specificity?** 2023. Disponível em: <https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/>.

BRINKER, T. J. et al. Skin cancer classification using convolutional neural networks: systematic review. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 20, n. 10, p. e11936, 2018.

BUSLAEV, A. et al. Albumentations: Fast and flexible image augmentations. **Information**, v. 11, n. 2, 2020. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/11/2/125>.

Cancer.Net. **Skin Cancer (Non-Melanoma): Introduction**. 2022. Disponível em: <https://www.cancer.net/cancer-types/skin-cancer-non-melanoma/introduction>.

CHAR, D. S.; ABRAMOFF, M. D.; FEUDTNER, C. Identifying ethical considerations for machine learning healthcare applications. **The American Journal of Bioethics**, Taylor & Francis, v. 20, n. 11, p. 7–17, 2020. PMID: 33103967. Disponível em: <https://doi.org/10.1080/15265161.2020.1819469>.

CHEN, Y. et al. Generative adversarial networks in medical image augmentation: A review. **Computers in Biology and Medicine**, v. 144, p. 105382, 03 2022.

CUMMINS, D. L. et al. Cutaneous malignant melanoma. **Mayo Clinic Proceedings**, v. 81, n. 4, p. 500–507, 2006. ISSN 0025-6196. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0025619611618983>.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 248–255.

DHILLON, A.; VERMA, G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. **Progress in Artificial Intelligence**, Springer, v. 9, n. 2, p. 85–112, 2020.

DILDAR, M. et al. Skin cancer detection: a review using deep learning techniques. **International journal of environmental research and public health**, MDPI, v. 18, n. 10, p. 5479, 2021.

ESTEVA, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. **nature**, Nature Publishing Group, v. 542, n. 7639, p. 115–118, 2017.

FELMINGHAM, C. et al. The importance of incorporating human factors in the design and implementation of artificial intelligence for skin cancer diagnosis in the real world. **American Journal of Clinical Dermatology**, Springer - Adis, v. 22, n. 2, p. 233–242, mar. 2021. ISSN 1175-0561.

FELMINGHAM, C. M. et al. The importance of incorporating human factors in the design and implementation of artificial intelligence for skin cancer diagnosis in the real world. **American Journal of Clinical Dermatology**, Springer, v. 22, n. 2, p. 233–242, 2021.

GéRON, A. **Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems**. [S.l.]: O'Reilly Media, 2017. ISBN 1491962291.

HA, Q.; LIU, B.; LIU, F. Identifying melanoma images using efficientnet ensemble: Winning solution to the siim-isic melanoma classification challenge. 2020.

HAGGENMÜLLER, S. et al. Skin cancer classification via convolutional neural networks: systematic review of studies involving human experts. **European Journal of Cancer**, Elsevier, v. 156, p. 202–216, 2021.

HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778.

KASSEM, M. A. et al. Machine learning and deep learning methods for skin lesion classification and diagnosis: a systematic review. **Diagnostics**, MDPI, v. 11, n. 8, p. 1390, 2021.

KAUL, V.; ENSLIN, S.; GROSS, S. A. History of artificial intelligence in medicine. **Gastrointestinal Endoscopy**, v. 92, n. 4, p. 807–812, 2020. ISSN 0016-5107. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0016510720344667>.

KOUROU, K. et al. Machine learning applications in cancer prognosis and prediction. **Computational and structural biotechnology journal**, Elsevier, v. 13, p. 8–17, 2015.

LINARES, M. A.; ZAKARIA, A.; NIZRAN, P. Skin cancer. **Primary care: Clinics in office practice**, Elsevier, v. 42, n. 4, p. 645–659, 2015.

MAGALHAES, C.; MENDES, J.; VARDASCA, R. The role of ai classifiers in skin cancer images. **Skin Research and Technology**, Wiley Online Library, v. 25, n. 5, p. 750–757, 2019.

MAMONU. **What is the scikit-learn dummy classifier?** Medium, 2018. Disponível em: <https://medium.com/@mamonu/what-is-the-scikit-learn-dummy-classifier-95549d9cd44>.

MANDAL, M. **Introduction to convolutional neural networks (CNN)**. 2021. Disponível em: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>.

MELARKODE, N. et al. Ai-powered diagnosis of skin cancer: A contemporary review, open challenges and future research directions. **Cancers**, U.S. National Library of Medicine, Feb 2023. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9953963/>.

MENDONçA, T. et al. Ph2 - a dermoscopic image database for research and benchmarking. In: **2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. [S.l.: s.n.], 2013. p. 5437–5440.

National Cancer Institute. **What Is Cancer?** 2021. Disponível em: <https://www.cancer.gov/about-cancer/understanding/what-is-cancer>.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. 2015.

OSISANWO, F. et al. Supervised machine learning algorithms: classification and comparison. **International Journal of Computer Trends and Technology (IJCTT)**, Seventh Sense Research Group, v. 48, n. 3, p. 128–138, 2017.

PEREZ, F. et al. Data augmentation for skin lesion analysis. In: SPRINGER. **OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis: First International Workshop, OR 2.0 2018, 5th International Workshop, CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16 and 20, 2018, Proceedings 5**. [S.l.], 2018. p. 303–311.

PUBLISHING, S. **Histology, Skin**. Treasure Island (FL): StatPearls Publishing, 2021. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK441980/#:~:text=The\%20skin\%20is\%20primarily\%20made,and\%20contributes\%20to\%20skin\%20tone.>

ROTEMBERG, V. et al. A patient-centric dataset of images and metadata for identifying melanomas using clinical context. **Sci Data**, v. 8, n. 1, p. 34, 2021. Disponível em: <https://doi.org/10.1038/s41597-021-00815-z>.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: a modern approach**. 3. ed. [S.l.]: Pearson, 2009.

SALIDO, J. A.; RUIZ, C. Using deep learning for melanoma detection in dermoscopy images. **International Journal of Machine Learning and Computing**, v. 8, p. 61–68, 02 2018.

SEBASTIAN, A. M.; PETER, D. Artificial intelligence in cancer research: Trends, challenges and future directions. **Life**, MDPI AG, v. 12, n. 12, p. 1991, Nov 2022. ISSN 2075-1729. Disponível em: <http://dx.doi.org/10.3390/life12121991>.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of big data**, SpringerOpen, v. 6, n. 1, p. 1–48, 2019.

SINGH, A. **Demystifying the mathematics behind Convolutional Neural Networks (cnns)**. 2020. Disponível em: <https://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network/?utm_source=reading_list&utm_medium=https%3A%2F%2Fwww.analyticsvidhya.com%2Fblog%2F2021%2F05%2Fconvolutional-neural-networks-cnn%2F>.

SINGH, S. **Understanding the bias-variance tradeoff**. Towards Data Science, 2018. Disponível em: <https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229>.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 2818–2826.

TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. **CoRR**, abs/1905.11946, 2019. Disponível em: <http://arxiv.org/abs/1905.11946>.

TAN, M.; LE, Q. V. Efficientnetv2: Smaller models and faster training. **CoRR**, abs/2104.00298, 2021. Disponível em: <https://arxiv.org/abs/2104.00298>.

TEAM, G. L. **What is Cross Validation in machine learning? types of cross validation**. 2022. Disponível em: <https://www.mygreatlearning.com/blog/cross-validation/>.

TSCHANDL, P.; ROSENDAHL, C.; KITTLER, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. **Scientific Data**, Springer Science and Business Media LLC, v. 5, n. 1, aug 2018. Disponível em: <https://doi.org/10.1038\%2Fsdata.2018.161>.

UConn Today. **Skin Cancer: When Found and Treated Early, Is Highly Curable**. 2021. Disponível em: <https://today.uconn.edu/2021/05/skin-cancer-when-found-and-treated-early-is-highly-curable/>.

WEN, D. et al. Characteristics of publicly available skin cancer image datasets: A systematic review. **The Lancet Digital Health**, v. 4, n. 1, 2022.

WU, Y. et al. Skin cancer classification with deep learning: a systematic review. **Frontiers in Oncology**, Frontiers, v. 12, p. 893972, 2022.

ZAFAR, A. et al. A comparison of pooling methods for convolutional neural networks. **Applied Sciences**, MDPI AG, v. 12, n. 17, p. 8643, Aug 2022. ISSN 2076-3417. Disponível em: <http://dx.doi.org/10.3390/app12178643>.