Open University of Catalonia (UOC) Master's Degree in Data Science

MASTER'S THESIS

Area: Medicine

Dermatological lesion prediction

Skin disease dermatoscopic classification using convolutional networks on unbalanced dataset

---

Author: Jesús González Leal

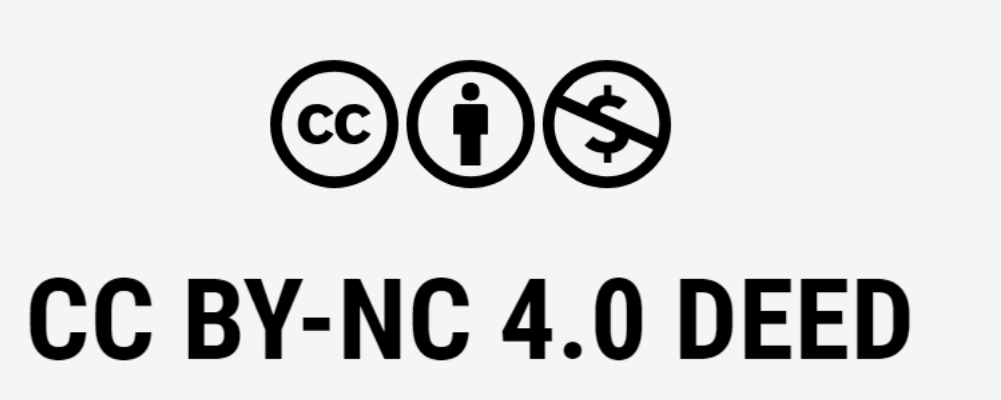
Tutor: Jordi de la Torre Gallart

Professor: Laia Subirats Maté

Barcelona, January 17, 2024

Credits/Copyright

The dataset used in this work is protected by the CC-BY-NC licence [1].



FINAL PROJECT RECORD

Title of the project:	Dermatological lesion classification
Author's name:	Jesús González Leal
Collaborating teacher's name:	Jordi de la Torre Gallart
PRA's name:	Laura Subirats Maté
Delivery date (mm/yyyy):	01/2024
Degree or program:	Master's degree in data science
Final Project area:	Medicine
Language of the project:	English
Code repository:	https://github.com/chus73/Dermatological-lesion-classification
Keywords	*I*mage classification, Transfer learning, SMOTE

Dedication/Quote

Dedicado a mis pequeños, la fuente de mi alegría. Espero serviros de inspiración y que podáis lograr todos vuestros sueños.

Acknowledgements

I would like to acknowledge the work of the International Sink Imaging Collaboration [2] in helping to reduce skin cancer mortality.

I would also like to acknowledge the "Universitat Oberta de Catalunya" for giving me the opportunity to grow personally and professionally.

To conclude this chapter of acknowledgements, I would also like to include my English teacher, Jennifer, for her patience in helping me to make progress in learning English.

Abstract

The skin is the largest organ in the body and the first barrier for defending our inner organs against aggression, besides helping the body regulate its temperature. Considering the importance of the skin for human beings, it is necessary to examine skin lesions. Because of that, dermatologists use dermatoscopy to illuminate and magnify the area to be examined to monitor any present injury.

In recent decades, enormous advances in the AI medical field have experienced significant improvements due to the use of deep networks. They can diagnose with reliability as well as speed.

The following project presents the use of deep learning algorithms in order to classify eight different diagnoses, such as melanoma, basal cell carcinoma, vascular lesion, and other lesions. In this study, we design and test two of today's most commonly used convolutional networks in the context of skin lesion classification, using the 2019 ISIC dataset. This dataset presented quality challenges, including resolution variations and very significant **inter-class imbalances**. We implemented two strategies to solve the dataset problem: an **under-sampling** and an **over-sampling** of the minority classes. We use **Synthetic minorities (SMOTE technique)** to enrich the minority classes, resulting in a 20 % increase in the total number of images processed.

Furthermore, we employ model training transfer learning techniques using EfficientNet B0 and ResNet50 as a model base, demonstrating significant improvements in classification metrics. In particular, ResNet50 showed higher performance when trained on the SMOTE-enriched dataset, and this can be attributed to its profound architecture benefiting from a larger dataset.

This research contributes to a new understanding of skin lesion classification through data preprocessing, transfer learning, and model selection, paving the way for future enhancements in dermatological image analysis and early diagnosis. Finally, it also contributes exploring the use of a statistic technique like SMOTE to solve an imbalanced dataset challenge.

Keywords: transfer learning, SMOTE, skin lesion classification, unbalance dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Skin diseases, such as skin cancer, affect a large number of people around the world. Many of them are associated with a social stigma because the lesions look impressive, causing rejection towards people having them. Others, such as melanoma, are also associated with a high mortality rate, which is gradually decreasing thanks to advances in its early detection. Premature diagnosis, especially when the disease is in its first stages, will help to improve its prognosis and evolution. As a contribution to improving skin disease detection, this work has used convolutional neural networks to classify eight different types of skin disease.

The dataset used for this study is part of the 2019 International Skin Imaging Collaboration (ISIC). The previously mentioned association launches annual challenges to stimulate researchers in detecting and classifying skin diseases. In concrete, the data used in this paper relates to the eight specific skin diseases shown in figure 1.1. The data series has been provided by different hospital contributions and has been obtained using dermoscopic techniques, which are widely used by dermatologists because they improve the diagnosis of lesions compared with the naked eye.

The different diseases collected within the data set are:

- **Melanoma (MEL)**: This skin disease is a type of skin cancer in the cells responsible for skin color, called melanocytes [3]. Although it is much less common than other types of skin cancer, its tendency to metastasize makes it much more dangerous.

- **Melanocytic nevus (NV)**: A melanocytic nevus, also known as a mole [4], is a non-cancerous skin condition that occurs in melanocytes. They usually appear during the first decades of a person's life and can be distinguished between acquired moles, which are a form of benign neoplasia, or excessive cell growth, and congenital moles, which are considered to be malformations.

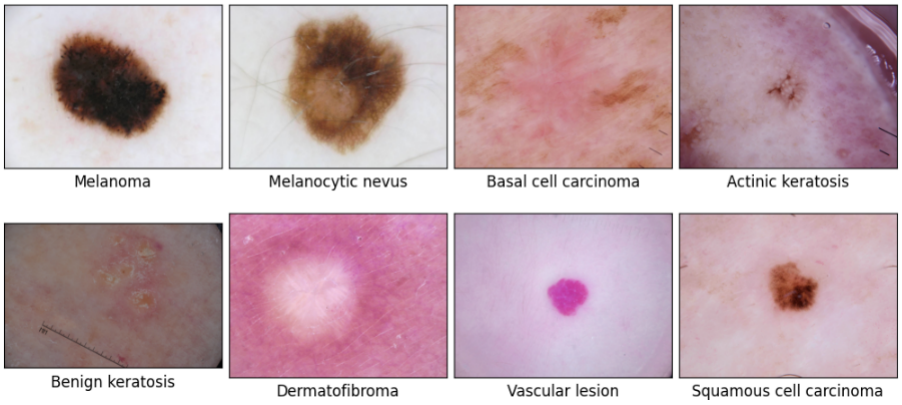


Figure 1.1: Skin lesions sample

- **Basal cell carcinoma (BCC)**: Basal cell skin cancer is one of the most common skin cancers and occurs in the basal cells found in the lower part of the epidermis [5]. Sun overexposure is often linked to its development.

- **Actinic keratosis (AK)**: An actinic keratosis is a rough, scaly patch, that presents with erythema or skin reddening in parts of the skin that are continuously exposed to the sun, such as the arms, ears, and face, neck or scalp [6]. They are usually smaller than 1cm, although sometimes a group of them may converge.

- **Benign keratosis (BKL)**: This type of keratosis is a common, non-cancerous skin neoplasm (abnormal cell growth) [7]. They have a waxy, scaly, brown, slightly elevated appearance. They usually appear on the face, neck, chest, and back, usually in adulthood, and are harmless and non-contagious.

- **Dermatofibroma (DF)**: This common skin lesion usually appears as a slow-growing nodule in the dermis [8]. Although it usually affects both sexes, it is more prevalent in women.

- **Vascular lesion (VASC)**: These are fairly common lesions affecting the blood vessels of the skin and can be acquired, or congenital, present at birth [9]. They may appear as pink, red, or purple marks.

- **Squamous cell carcinoma (SCC)**: Like basal cell cancer, squamous cell cancer is related to sun damage to the flat cells on the outside of the epidermis [5].

Throughout this work, both the name and **the abbreviation in brackets will be used interchangeably**.

1.1 Motivation

This journey towards the elaboration of this exciting work on dermoscopic classification of skin disease is a path that combines science, technology, and medical care uniquely.

Artificial Intelligence (AI) is bringing about a fourth industrial revolution that is already affecting our society. In concrete, its incorporation into medicine has a very significant impact on different medical areas ([10] y [11]) such as:

- **A more accurate and organized diagnosis**: Advanced search techniques on large medical data sets allow for more accurate and reliable diagnostics.
- **Faster and more efficient drug development**: Accelerating the research and development of new drugs.
- **Personalised care**: AI can customize treatment plans for each patient's individual needs, optimizing, at the same time, effectiveness and minimizing side effects.
- **Images diagnostic more accurate and faster**. The worldwide name of AI will be recognized henceforth in this work as Deep networks. A part of these networks is specialized image recognition. They have a pattern detection capability that is by far superior to any human being. They can also make a diagnosis with very little delay. For the aforementioned reasons, it will be an essential tool for improving patient care and treatment.

This last point is the core of the scope of this project. Personally, this project is much more than a task for ending a long master's journey. This job represents a challenge because working with large image files requires pre-processed data. In addition to this, the model to be developed is a classification one that should distinguish among nine different patterns.

It also represents an opportunity to imagine that my work could be helpful, one day, in the early detection of multiple skin diseases, and thus improve medical care and services.

1.2 Goals

This project's main objective is the classification of dermoscopic images in order to identify eight types of skin lesions. A Convolutional Neural Network (CNN) will be created to achieve this. This challenge can be separated into two goals.

- On the one hand, to analyze the current **state of the art** in automatic medical detection by imaging.
- Secondly, to **classify these lesions into one of eight classes** to be studied.

1.3 Methodology

For the implementation of this project, we will use the Cross Industry Standard Process for Data Mining or **CRISP-DM** methodology. This methodology consists of six phases that are cyclically carried out, and it allows feedback to previous phases in order to complement the deficiencies of other phases. The following diagram (figure 1.1) shows the six-phase cyclical circuit to be used in this methodology.

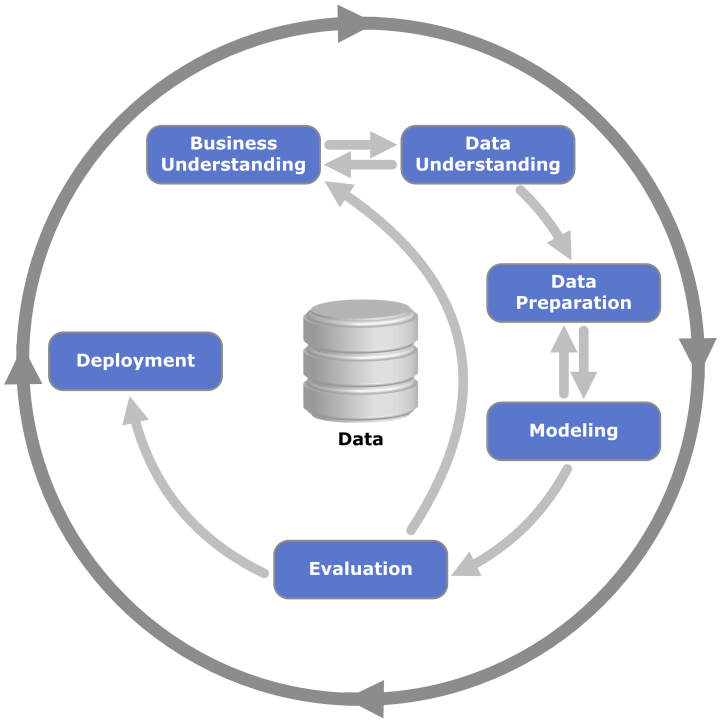


Figure 1.2: Process diagram showing the phases of the CRISP-DM methodology

The methodology phases and the adaptation to the project are described below:

1. **Business understanding**

   The first phase of the **CRISP-DM** model starts with understanding the project's main objectives. These are mentioned in the **Goals section** above. At this stage of the project, licensing costs are reviewed. Concerning licensing cost in this project there is **no associated licensing cost**, because the data is left under an open licence (see **Licensing section**). Furthermore, the project plan is also created during this phase (see **project plan section**).

2. **Data understanding**

Typically, this phase seeks to become familiar with the understanding of the data, identifying quality problems and discover hidden information or interesting subsets of the data to be processed. This allows to have point of view. A key aspect will be to verify the quality, in order to define all required strategies to address them. At the end of this phase, we will already have a conceptual data model.

As the images to be used as a data source are produced in a medical environment, they are expected to be of high quality. However, this will be validated in the early stages of the project.

3. **Data Preparation**

   In this section, applying the **CRISP-DM** method, a selection and integration of the main data together with their required attributes will be carried out. Moreover, data will be cleaned and formatted if required, looking for the main fields or characteristics of the data.

   Our project has the challenge of working with **large data input** (around 21 GB). Therefore, one of the first tasks will be **resizing** the data to reduce the size and work with a more manageable dataset. On the contrary, we will look at using some **data augmentation** techniques to help us avoid overfitting.

4. **Modelling**

   In this phase, the modeling techniques required to achieve an optimal data model will be applied. As it is shown in the phase diagram, it is a phase that usually builds on the data preparation phase, either to solve problems or to enrich the data. Additionally, to model building, some model evaluation techniques will be required. As a result, a test plan document will be done.

5. **Evaluation**

   Once the objectives set required during the initial phases have been obtained, an evaluation must be carried out to ensure that they were reached. This is done using the test plan designed in the previous phase. If necessary, the process will be reviewed, and depending on the results, the next steps will be planned. A benchmark will be carried out among various models created, and the outstanding model will be selected.

6. **Deployment**

   At the time, the model has passed all the tests and has been approved, it is required to define a plan for its deployment. Thus, creating a final model is not the end of the project, but rather a milestone within it. A model put into a production environment is

necessary to monitor and maintain constantly. Along with the previous, it is advisable to publish results that help to understand whether the objectives initially requested in phase 1 have been achieved. It is quite common that after the project has been put into production, new expectations are generated that make the model (project) generated be reconsidered. This is why **CRISP-DM** describes the life cycle for the data mining project in a circular framework, where the output feeds back into a new cycle that seeks the continuous improvement of the business.

Our project is part of a work that will develop a minimum viable product. Finally, the outstanding model will be run on the dataset test to obtain the final metrics.

After the project is structured by phases where no changes are expected in the definition of each one of them, and where the beginnings and ends of each phase are very well defined, a **waterfall model** is a better approach than other **agile strategies** for the project follow-up.

1.4 Planning

The project has been structured in **five phases** or modules, which in turn are broken down into smaller tasks. This section describes the project's planning and its main milestones.

- **Module 1 - Project definition**

  This module starts defining the project's plan: the relevant objectives of the project, the initial planning that will form the basis of the project, and the personal motivation to carry out the project on the chosen topic. During this phase, the form "Ethical and personal data protection protocol" will be filled out and presented.

- **Module 2 - State of Art**

  During this module time, we will carry out a research phase in which we will search for previous or current scientific work related to this project's main topic.

- **Module 3 - Design & implementation**

  The objectives of this activity are to implement the tasks necessary for the project's design and development according to the chosen scientific methodology. During this period, the progress made on it will be documented in order to complement the project document.

- **Module 4 - Document redaction**

  This action's purpose is to prepare all the materials required for the presentation and final TFM (Master's Thesis) assessment. These materials include:

1. The Master's Thesis document

2. An audiovisual presentation

- **Module 5 - Project defence**

  Finally, the last step of the Master's Thesis is to defend it before a board of examiners.

The following table shows each of the phases, namely: the start and end dates, as well as the estimated effort.

Module	Description	Start Date	End Date	days	Effort (h)
1	Project definition	09/27/23	10/10/23	13	26
2	State of Art	10/11/23	10/24/23	13	26
3	Design & Implementation	10/25/23	12/19/23	55	110
4	Document redaction	12/20/23	01/16/24	27	54
5	Project defence	01/17/24	02/03/24	17	34

Table 1.1: Project Timetable.

The time evolution of the project is shown in the following Gantt graphs.

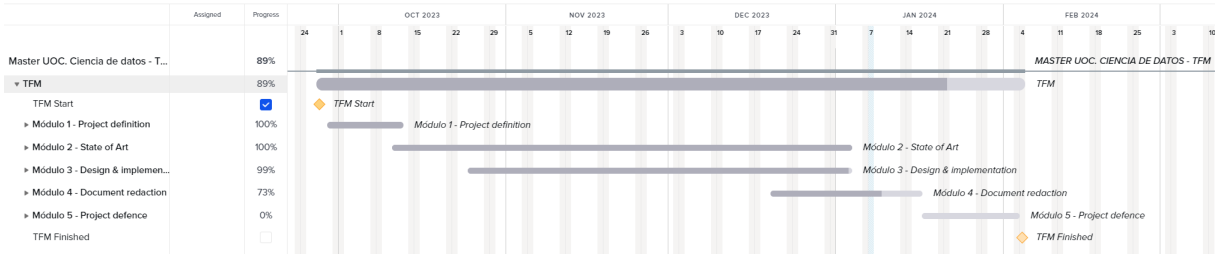


Figure 1.3: Master's final project Gantt's chart. General view

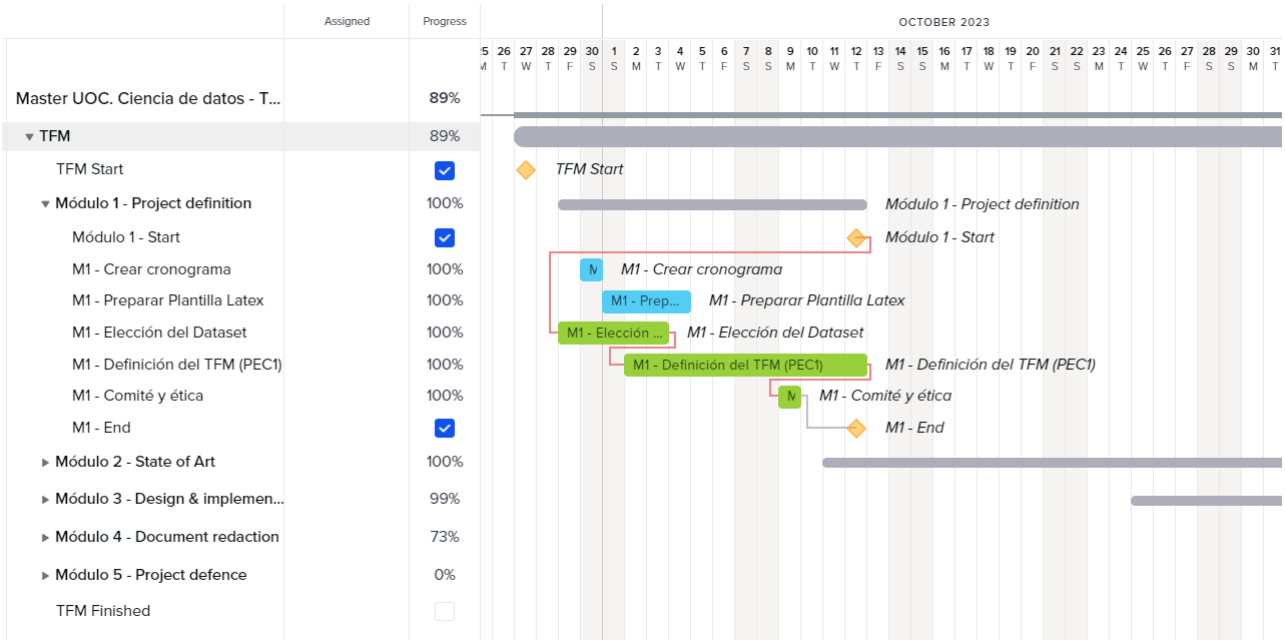


Figure 1.4: Master's final project Gantt's chart. Module 1 view

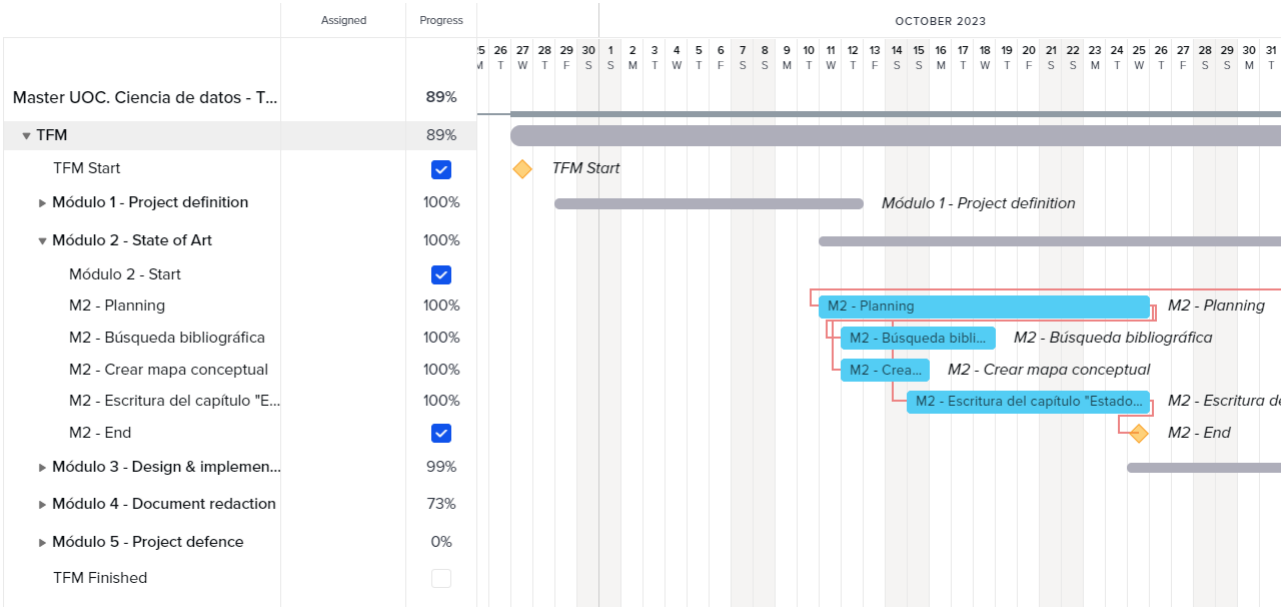


Figure 1.5: Master's final project Gantt's chart. Module 2 view

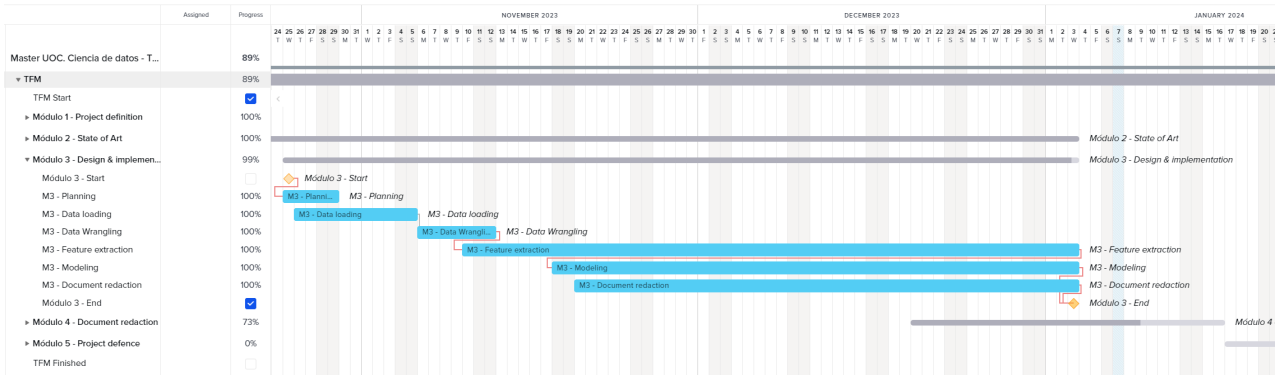


Figure 1.6: Master's final project Gantt's chart. Module 3 view

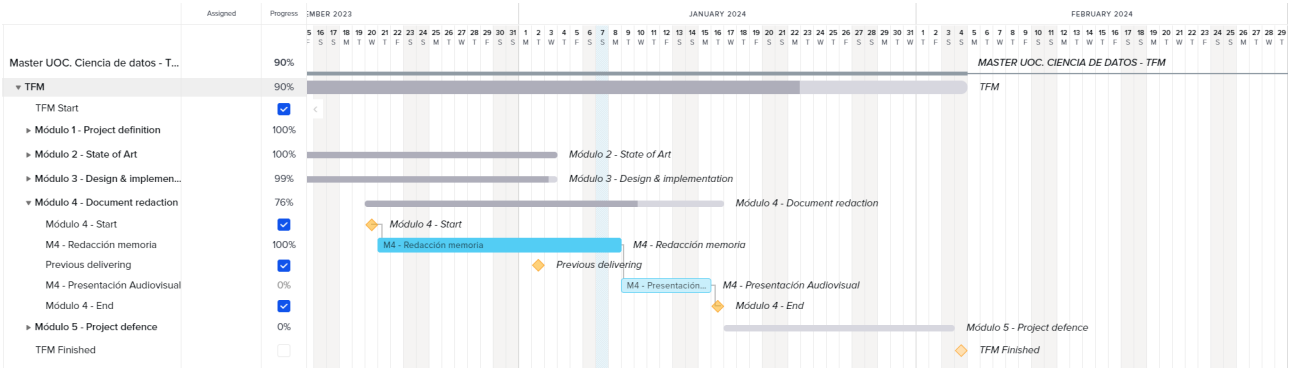


Figure 1.7: Master's final project Gantt's chart. Module 4 view

Chapter 2

State of the Art

This chapter's major aim is to revise the existing scientific literature on the classification of skin lesions, which represents a fundamental part of the research work. The section has been presented in chronological order, which allows us to see the evolution in this field of medicine over time. The following chapter is structured into three sections: First, the architectures that were initially used based on Machine Learning (ML); second, the emergence of Neural Networks using image classification; and finally, the latest trends in this kind of architecture. In each section, the advances are introduced, identifying characteristics and limitations.

2.1 Feature Extraction in Machine Learning

Since last century decade, international research into expert systems for image processing in medicine to support diagnosis has been in progress [12], [13]. We could already find diagnostic work based on neural networks [14] at the end of the 20th century, nevertheless, the trend in research was using Machine Learning (ML) algorithms for image classification [15], [16]. Thus, during the first years of the 21st century, research efforts focused on improving ML algorithms' performance by adding a **previous layer feature selection** [17]. An example of it can be found in this text [18], where the authors performed a benchmark with machine learning classifiers, measuring the impact of using a preprocess to extract texture, contour delineation, and geometric properties of the melanoma lesion before moving to the classification stage. They conclude that the three features used make the classifier perform better, with texture being the dominant feature to improve the accuracy metric. Finding feature extraction allows sample discrimination and classification which is ML algorithms' main problem when processing medical images, including dermoscopic images. In this medicine area, the asymmetry, border, colors, and dermoscopic rule (also known as **ABCD rule** in dermatology) or the color, architecture, symmetry, and homogeneity (**CASH rule**) is common to support the classification [17]. Con-

cerning this point, in 2012 [19], we could find a research document where a **complete feature extractor** was built with a preprocessing stage for lesion boundary delimitation, the symmetry presented two axes, the lesion geometry and as well as its texture. The feature output extractor fed four ML classifiers (Support Vector Machine (SVM), random forest, logistic model tree, and hidden naive Bayes) with outstanding results. In 2014, the k-means algorithm used for sample separation into colors, together with logistic regression, already achieved a sensitivity of 62% and a specificity of 76% [20]. Meanwhile, in 2015 [21], the application of SVM in cascade configuration together with a filter for hair detection, a Gaussian low-pass filtering on each of the channels allows higher accuracy, approaching 98 %.

2.2 The Image Processing Evolution with Neural Networks

As we have presented above, research in image processing using neural networks dates back to the end of the 20th century, and it is from the first decade of the 21st century that developments based on this technology begin to emerge strongly, though. This is due to the use of graphics processing units (GPUs) to compute the calculations, subsequently, tensor processing units (TPUs) were added, together with the provision of large labelled data sets and the commitment to innovate by large technology companies and institutions. Thus, in 2011, the famous paper "*Building High-level Features Using Large Scale Unsupervised Learning*" [22] by Google engineers established a turning point in image processing by using neural networks for large-scale unsupervised learning on 31,000 images. Unlike the ML models seen above, neural networks belong to the group of unsupervised classifiers, where the system identifies the main features, without relying on manual supervision [22]. In contrast to the previous trend in ML, which added complexity to models by focusing on the feature extraction layer, with neural networks, the **network learns to extract the most relevant patterns from the image by itself**.

According to the international magazine, within the set of types of neural networks, we find that there is a group with direct applications in image processing. We refer to **convolutional neural networks** (CNN) [23] [24], which extract increasingly complex patterns as we go deeper into the layers of the network, increasing the level of abstraction of the object concerning the image. In this way, the network learns features and patterns that allow it to distinguish the object through training, accurately detecting those same patterns in other images not used during that process time. Each layer of the CNN neural network acts as a filter that detects a certain pattern in the image. Therefore, it is easy to think that increasing the number of layers will increase the model's reliability. Nevertheless, this action will also increase the number of

parameters to train, increasing the training time of these models. This philosophy gives birth to the concept of **Deep Learning** (DL) [25] as a subfield within the area of ML.

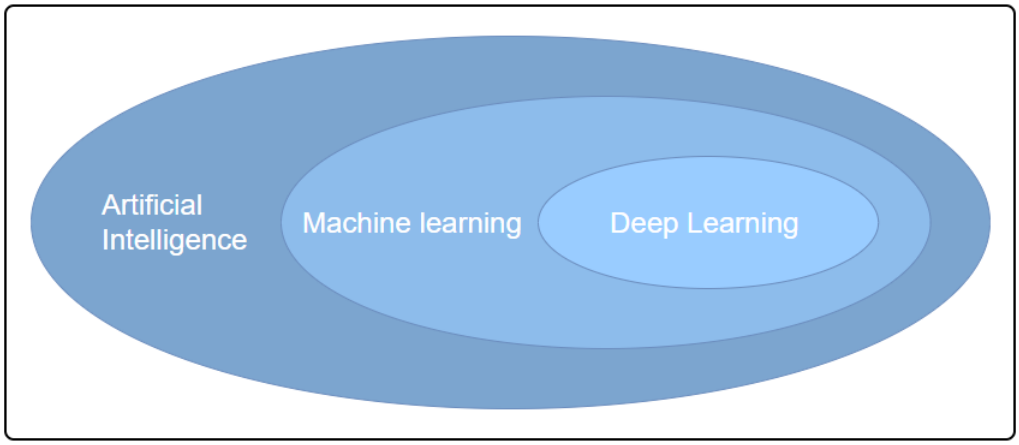


Figure 2.1: Relationships among the subgroups that make up artificial intelligence.

2.3 Advances in Image Classification for Dermatology: Overcoming Challenges with Deep Learning and Pre-processing Images

The contemporary trend is to use **intense networks** for image classification in the medical field, specifically in dermatology. However, incorporating more and more layers results in an increase in the parameter numbers, causing a possible network degradation [26]. To overcome the degradation problem in this type of network, a technique used is **residual learning**, with a Fully Convolutional Residual Network (FCRN). It can be seen in [27], where intense neural networks have used more than 50 layers with an initial image segmentation process before moving on to the classification stage. This type of network, [28], is based on the idea of allowing neural network layers to learn the differences between the input and output, also known as residual learning. This is achieved by introducing hops in one or more network layers, making convergence possible in deep networks, and avoiding exploding or vanishing gradients [29].

In order to have a better model accuracy without increasing the number of layers too much, it is necessary to go back to the idea of implementing a pre-processing layer to extract the main features before the classification. Thus, in this work [30], a k-means machine learning algorithm is employed at the pre-processing stage, whose purpose is to segment the pixels by color. In this

sense, these layers aim to isolate the lesioned region from the rest of the skin and then go on to a feature extraction stage to determine the color, the information intensity of each channel, and the lesion pattern using different algorithms. The result is then passed to a feature selection stage that uses the statistical *t-test* to decide the feature's importance. These are classified by category to determine the image's dominant feature set, and the classified features feed a deep neural network. The results obtained are significant, reaching 99 % Accuracy with 98.33 % sensitivity.

Another trend used in the last decade for the specific case of skin photo processing is the incorporation of a filter before the convolutional network removes the noise caused by hair in dermoscopic images.[31], [32], [33].

One of the problems facing neural network training is the need to process an extensive set of quality data. To minimize the impact of having poor data and improving the time used during this part, the **transfer learning** technique has been implemented in neural network architectures over the last decade [34], [35], [36], [37]. The concept behind this technique lies underneath, taking advantage of the knowledge acquisition process for other long and specific neural networks. Instead of training a model from scratch for a new task, transfer learning uses a pre-trained model from a previous task as a starting point. This pre-trained model has already learned useful features and data patterns that can be applied to the new task, which speeds up the training process and can result in better performance.

Another problem often faced by CNNs is the risk of overfitting due to limited data sources. In these cases, it is common to incorporate **data augmentation** techniques to introduce variability and increase the amount of data [38]. These techniques consist of transformation methods to the image, such as rotation, translation, flipping, zooming or noise injection. There are other techniques used to increase the number of samples in small datasets. In 2002, Chawla, Bowyer, Hall, and Kegelmeyer proposed **SMOTE** (Synthetic Minority Over-sampling Technique) [39], an algorithm that generates new synthetic samples by interpolating minority class elements. To do this, as they describe in their paper, they operate on the feature space of the sample, taking distance from it and its nearest neighbors and combining their features with a random weighting combining images. Furthermore, we have found other variations of this algorithm, such as **SMOTE-I**, carried out by J. Moreno et al. [40], which allows working with more than one minority class since it takes into account the classes distribution and generates the synthetic instances according to their distribution. More recently, in 2015, María José Basgall et al. presented on spark an adaptation of the original algorithm called **SMOTE-BD** [41], that is prepared to work with large massive data.

Chapter 3

Design and Implementation

The content of the following chapter is about the dataset source description and the images' origin of dermatological characteristics. Further on, the text indicates the **data pre-processing** procedures in order to re-scale the original images as well as normalize the labels that identify the skin diseases, besides the augmentation on data processing.

Another section forming this chapter is the **implementation** of a Convolutional Neural Network based on *EfficientNet* architecture and pre-trained with *ImageNet* dataset. This neural network will be trained with a small selection of data to obtain appropriate metrics using fine-tuning parameters.

Based on the previously mentioned architecture, different neural networks will be built, trained, and tested. Meanwhile, their scores will be reviewed in the next chapter.

3.1 Exploratory analysis

In this section, the *ISIC Challenge dataset* 2019 was used. It is formed by a set of 25,331 dermoscopic images related to 8 different types of skin diseases [42], [43] y [44]. The images are stored in **various resolution sizes that need to be unified into a smaller single-size**. A representation of the image resolution is given in the following table 3.1.

img_resolution	counts
1024x1024	12414
600x450	10015
1024x680	1121
1024x768	774
1024x682	173
...	...
1024x878	1
1024x861	1
1024x858	1
1024x830	1
966x645	1

Figure 3.1: Images resolution distribution Spreadsheet.

As these are real patient images, we also encounter other shapes and features, such as hair, lesion markers, and other objects that can distort the sample.

The dataset used for this research implies a larger quantity of one specific kind of images that causes an imbalance concerning the proportion of the total images quantity in each class, being Nevus, the one with the biggest class with more than 10.000 images. Compared with DF only having 215 images, this causes an enormous disparity. Figure 3.2 shows how the different classes are distributed across the dataset.

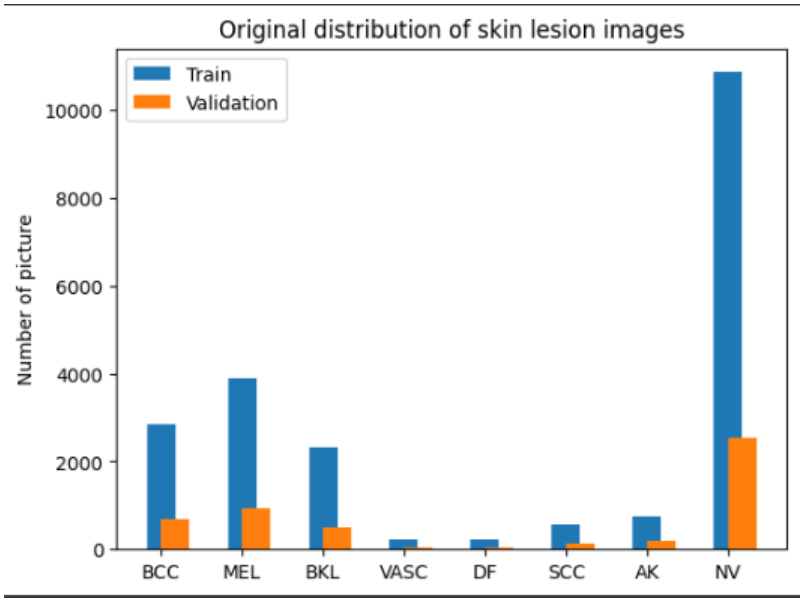


Figure 3.2: Original class distribution for training.

A disproportioned dataset could result in wrong output metrics because there is an induced bias during the network training. This might cause the network to give more importance to those classes with more weight in the dataset distribution. As a result, the lower weight classes could be ignored affecting the neural network performance. Due to all this, it is necessary to balance the dataset prior to training the neural network, so that every dataset class has the same weight, improving the network capability to learn.

Different techniques to correct an unbalanced dataset exist, such as:

1. **Re-sampling Techniques**: Consists of applying oversampling or sub-sampling in order to make equal class proportions.

2. **Data augmentation**: Involves doing different copies of a picture, introducing variables such as picture proportion, brightness, color saturation, and others.

3. **Feature selection**: Includes a careful selection of the main feature that describes the image to improve the minority classes' representation.

4. **Sintetic image generation**: This technique involves the creation of artificial images that are very similar to the real thing.

The data preparation framework to be used in this work includes the **sub-sampling** and **over-sampling** phases, which will allow us to have a balanced dataset, and we will apply **data augmentation** to improve the data diversity in order to make our models generalize better avoiding the overfitting.

3.2 Data Preparation

Neural networks process data in matrix format. The size of the dataset downloaded from ISIC exceeds 2 GB of information in image format. This makes it difficult to handle as a whole. In addition, as we have seen above, these images have d**ifferent resolutions and an imbalance between the different classes** 3.2. The high level of processing required to convert the original resolution images into matrix numbers is shown in 3.3. This one is composed of five steps that will be explained as follows:

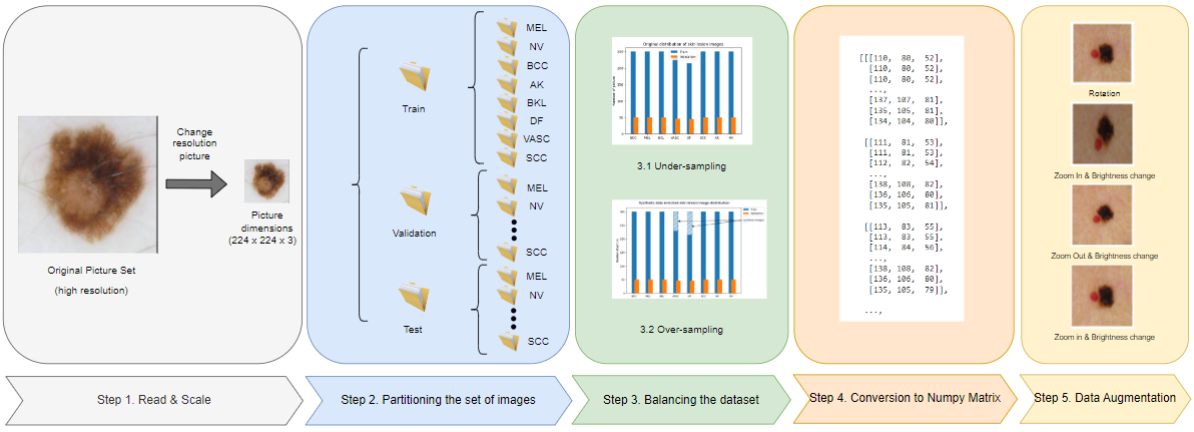


Figure 3.3: Data Preparation Framework.

3.2.1 Changing image resolution (Step 1)

As we introduced in the previous chapter, images came originally in a wide variety of resolutions. According to that, it will be necessary to **reduce and reshape** them. So as to, make them to be the same resolution and square-shaped. The size is determined by the network used as the feature extractor. In this work, we will use EfficenteNet B0 and ResNet50 networks, which require an input size (224x244x3) corresponding to the width, height, and number of channels of the image.

3.2.2 Image set partitioning (Step 2)

The higher the number of images, the higher the memory consumption during the classification process. With small datasets, we would choose to load the matrices in memory, which would increase our models' training speed. With the data volume we manage in this job, however, we will have to choose **batch loading from disk**. To achieve this, the pipeline needs to classify the images according to their type by distributing them in different folders, both for training and validation.

Additionally, the batch process needs a determined folder structure to work properly. It needs the images will be classified according to their nature. During this process, the set of images **label** is determined by the name of each folder, as shown in figure 3.3, step two.

The test dataset provided by ISIC 2019 does not have the labels to identify the type of lesion, and therefore, we cannot validate the performance of our models at the end of our work. Consequently, we will generate a test dataset from the original training set, randomly selecting a certain number of images for each class.

3.2.3 Balancing the dataset(Step 3)

As we have seen above in the description of the dataset, it is extremely important to work with a dataset in which all the classes are represented in the same proportion, in what is known as class balancing. This avoids introducing important **biases** that could lead to errors in the interpretation of the results and, especially when working with neural networks, avoids overfitting the training. As we indicated in the previous section, there are different techniques for managing a dataset with low representation in some of the classes. In this paper, we will use **under-sampling**, together with **synthetic image generation**, and a third, **data augmentation**, which we will apply at a general level to avoid network specialization. Additionally, an **over-sampling** process based on SMOTE will be used. This will be carried out in parallel in order to study the impact of introducing new synthetically generated images.

1. **under-sampling** To obtain a balanced dataset, we apply a trim to the majority classes so they can be made equal in size. We will apply a **random selection** to the samples of each class because we want to preserve the richness of having samples from different hospitals.

   In the picture, we can see the result of applying the trimming in each of the classes. The number of images to keep per class is determined by the number of minority classes, and the limit will be set at 250 images per class.

2. **Synthetic image generation** Applying trimming to the dataset will allow us to have a

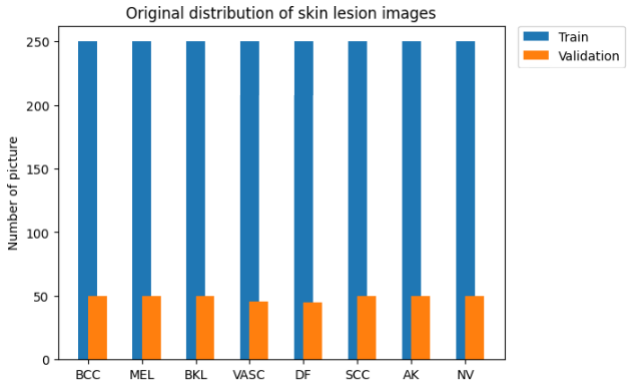


Figure 3.4: Skin Lesions balance distribution after applying a reduction on majority classes.

balanced dataset, but on the other hand, we will lose statistical diversity in the data by discarding images of the majority classes. Using techniques such as data augmentation, also used in this work, allows us to add variation to the data and avoid specialising the network. However, an alternative solution, which also increases the statistical richness, is to **artificially increase the data set of the minority classes**. This is the purpose of the **SMOTE procedure**, which will be the subject of a more detailed discussion later in this paper.

3.2.4 Image conversion to matrix (Step 4)

Since images are made up of pixels, their colors are constituted by numbers. For this work, they are specifically represented in RGB encoding. This means that each pixel has a numerical translation ranging from 0 to 255 values. Depending on the feature extractor used, the values of the numerical matrix have to be **normalized** between 0 and 1 or between -1 and 1.

3.2.5 Data Augmentation (Step 5)

Small data sets are often used to train neural networks. This means that the network tends to **specialize** immediately due to the small variety of data, in a process called **overfitting**. With the data augmentation technique, we generate a **series of derived images** from an original one. We have applied to them shifts, rotations, zooms, splits, or changes in brightness or color saturation to obtain new pictures. This makes the dataset richer and increases the number of images to train. As a demonstration of this, we have applied the transformations that we will perform on all the images in the training dataset to a random image in the figure 3.5.

Figure 3.5: Data augmentation examples.

3.2.6 Synthetic image generation

As we saw earlier at 2 when working with an unbalanced dataset, it is necessary to apply a number of pre-processing techniques to achieve a balance between the classes that make up the dataset. One of the ways to rebalance the dataset is to use oversampling by generating new samples. To do this, we will use the Synthetic Minority Oversampling Technique, or SMOTE [45] [46] [47], to add new images to the minority classes.

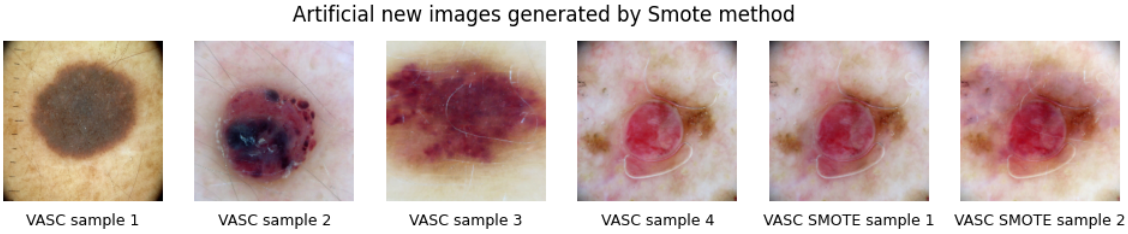


Figure 3.6: Example of SMOTE generated synthetic images.

The idea behind SMOTE is to identify the nearest neighbors of the sample and **interpolate new data** along the line connecting the sample to its neighbors according to a random value. Thus, this new synthetic data will have attributes of the original data and its neighbors while generally retaining visual features, texture, and spatial information [45]. The percentage of inherited features will depend on the proximity of the new data to the sample or neighbor.

To better understand this concept, the above has been illustrated in figure 3.7, where the base image has been linked to its three neighbors, as well as the artificially generated images along the imaginary line connecting them. The generated image 1 (*VASC SMOTE sample 1*) has been placed very close to the base image because it shows only very slight color changes. On the other hand, the second image (*VASC SMOTE sample 2*) has characteristics of both the base image and the neighbor *VASC SMOTE sample 3* and is closer to the last one.

In order to test whether it is possible to use SMOTE to improve classification efficiency,

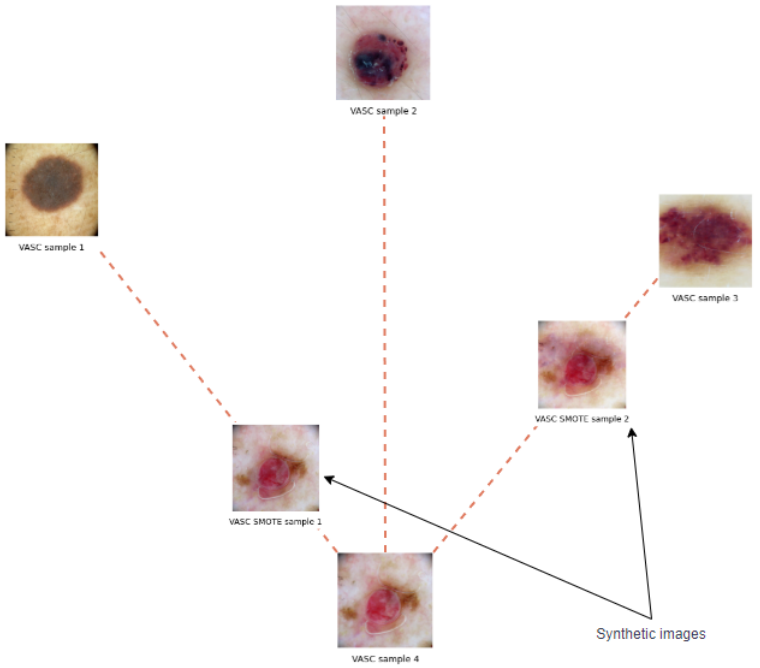


Figure 3.7: Synthetic image generation representation in 2D space

two variations of the same experiment are generated. In the first, where no synthetic samples are used, we will train the convolutional networks using a subsampling of the majority classes, setting the number of samples to 250 for each class, as shown in the figure 3.4. In a second variation, we will increase the size of each class to 300 images, and using SMOTE, we will synthetically generate 150 samples in the minority classes until the defined level is reached. A representation of the distribution of data for each

At the end of the process described above, the synthetically generated images are stored on disk to make them available for further training. Three random samples were taken from each minority class, and the result can be observed in figure 3.9.

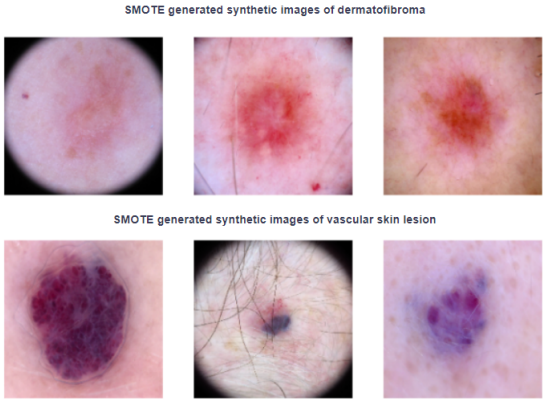


Figure 3.9: Artificial sampling generated by SMOTE

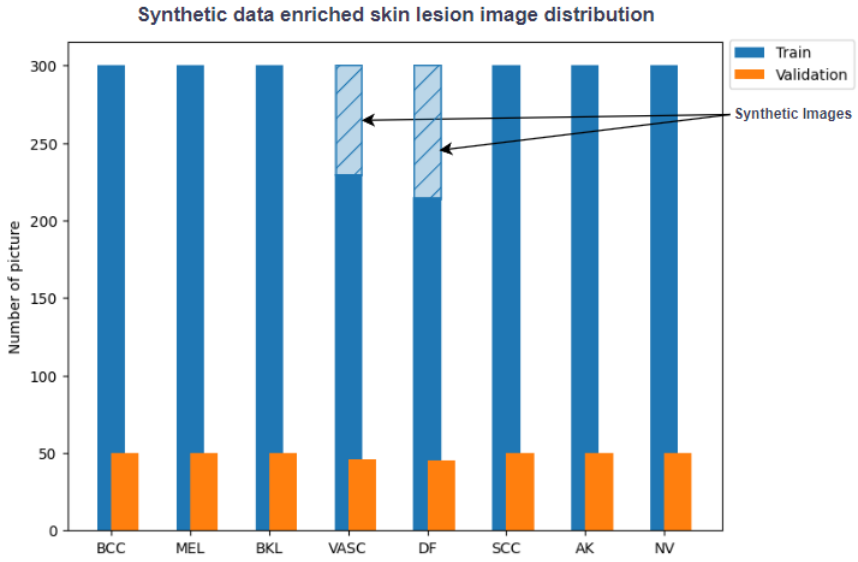


Figure 3.8: Synthetic data distribution between minority classes

3.3 Looking for the best model

In this chapter, the most practical, the training of convolutional neural networks for skin lesion classification will be carried out. Two of the best-known networks will be used: EfficientNet in its less complex version (B0) and ResNet50.

First, we will introduce the main components of this type of classification network: the feature extractor and the classification layer. The section will be illustrated with a representation of one of the networks used, the ResNet50, together with the classification block used during the training process.

In the following, we will describe the learning process that we will use during training. With this process, we will train the network to use the knowledge learned by the feature extractor to distinguish shapes and contours by transferring the weights of Imagenet, a dataset consisting of more than 1000 labeled classes with a total of 14 million images.

In the following section, the hyperparameters of the network are searched. This is the first step that any model must go through before being trained, and it consists of studying the network's behavior during short training sessions while varying the training parameters. This will allow us to decide on the best combination of parameters to use during the training process.

Finally, the chapter will conclude by describing the results of the training processes with the two networks used in this work, **EfficientNet B0** and **ResNet50**.

3.3.1 Image classification model assembly

As we have seen throughout this thesis, convolutional neural networks (ConNet) are widely used in image classification processes. These networks are decomposed into two stages:

1. The **feature extractor**, the first stage, is responsible for detecting patterns in the image whose complexity increases as we deepen the network.

2. The second stage is applied for the **classification layer** is connected to the output of the feature extractor and is responsible for determining the class to which the input belongs.

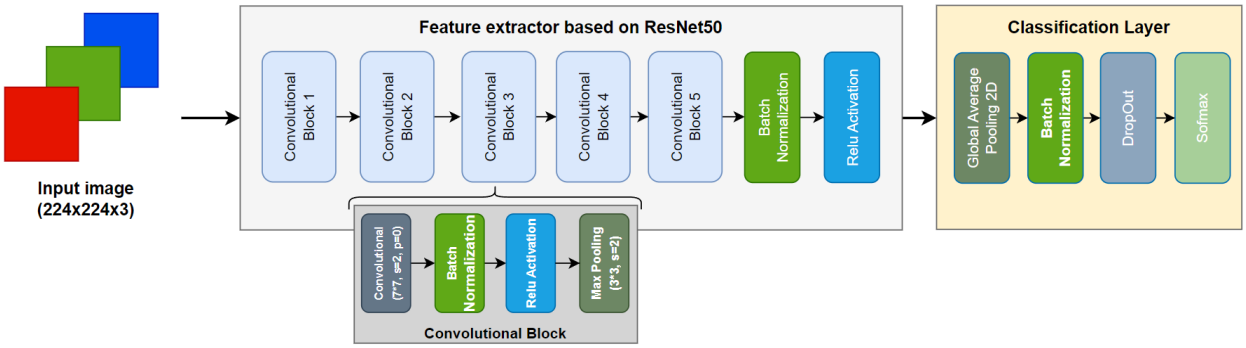


Figure 3.10: ResNet50-based assembly model used for testing

The figure 3.10 shows the two stages discussed above. For the feature extractor stage, also called the **base model**, the RedNet50 network map used in this project was chosen. The other network, *EfficientNet*, changes the number of convolutional blocks used and their composition, but the construction scheme follows the same pattern for didactic purposes.

The **base model**, based on ResNet50, consists of five convolutional blocks, followed by a batch normalizing layer and activated by ReLu. **Batch normalization** layers calculate the mean and standard deviation of a batch during the training process, ensuring the stability of the network by centering and normalizing each mini-batch of data, which is placed in front of an activation function, as shown in the figure. In contrast, the **ReLu activation** function is used to bring the network into non-linearity. This function, which is very common in neural networks, is responsible for initializing the negative values of the layer's input function to zero. On the other hand, each block contains a convolutional layer with a 7x7 kernel, a stride set to two and no padding, a batch normalization layer, a layer applying the ReLu function, both described above and a **max-pooling layer**. This last layer is responsible for reducing the dimensionality by selecting the maximum value of the region in which it operates, which, in the case of the convolutional blocks of ResNet50, corresponds to a 3x3 matrix.

The second stage of the model, the classification module, assigns labels to the feature extractor's output results. This stage consists of four layers: The operation of the first of these, **Global Average Polling 2D**, is very similar to Max Pooling above, except that it calculates the average of all input values for each channel. This reduces the dimensionality of the number of channels in the input layer. The second layer, the Bach normalization, was explained in the previous section. Following this is a **dropout layer**, which randomly deactivates certain neurons during each iteration. This introduces randomness to the training process and reduces the risk of overfitting. In this paper, the dropout rate has been set to 0.2, meaning that 20% of the neurons will be deactivated. Finally, the classification module's last layer utilizes the **softmax** function to convert input values, known as logits, into probabilities. This ensures that the sum of the values of each class equals unity. The softmax function is commonly used in multi-class classification problems, such as the one analyzed in this paper. Its output facilitates the interpretation of the network output.

To conclude this section, we will discuss one parameter that must be chosen to train the networks. This is the **loss function**, and it is used to quantify the difference between the model predictions and the validation data. The result of this function is used to adjust the neural network weights at each training step, aiming to minimize the error obtained by the function. The function chosen for this purpose, and one of the most common in the world of deep learning, is the **categorical cross-entropy** [48], which is described by the following formula.

$$CrossEntropy = -\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{c} y_i log y'_i$$

This function requires the label values to be **encoded by one-hot**. This type of encoding assigns the value 1 to the correct class, leaving the rest of the classes at 0. The nomenclature used in the above formula is as follows: "i" represents the class, "c" the total number of classes, "y_i" is the value of the actual class and "y'" the value of the predicted class. Due to the use of one-hot encoding, the calculation will be applied only to the real class, and to the value to be obtained at the output of the softmax function, reducing to zero the remaining sums of the polynomial belonging to the rest of the other classes.

3.3.2 Transfer learning

To achieve high metrics, it is necessary to train large data sets and long neural networks with many layers. An alternative to it is to use a pre-trained network with a very large dataset. This network has already learned to extract basic features from images efficiently.

Have you ever seen a cartoonist working? He starts by drawing basic shapes, lines, circles,

and ovals, and he composes the painting as an overlay of these shapes, adding shadows and gradients to simulate volume. The idea behind transfer learning is similar to this example. It consists on using this network in the first step as a pattern recognition layer, known as **a feature extractor**. The goal of this is to extract the most informative, identifying edges, corners, and blobs and extracting, at the same time, the color histogram. As well as in this step, a data dimensionality reduction to make it suitable for the next layer. Then, the classification layer is added according to each specific categorized problem. This way, transfer learning can be applied to tasks where the dataset is not large enough to train a complete model from scratch.

Transfer learning is performed according to the following workflow:

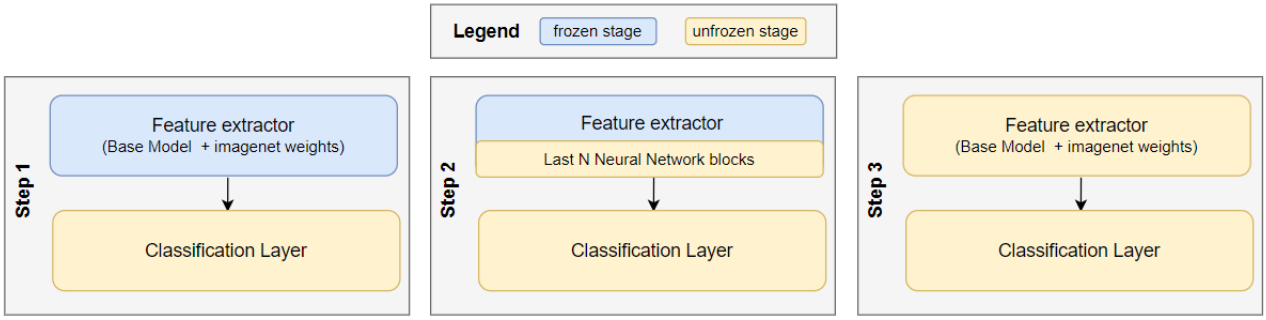


Figure 3.11: Transfer learning phases.

1. In the first step, a **pre-trained model**, which will be called the **base model**, is loaded with the training weights. This model is loaded without the classification layer (the model's top layers). The models used in this job are: *Efficient B0* and *ResNet 50*, both with the preloaded "*ImageNet*" weights.

2. Since we want to preserve its ability as a feature extractor, we **freeze all layers** in the base model. This way, we avoid destroying the information in future training rounds. Furthermore, a new set of trainable layers is added on top of the base model to form the **classification layer**. The whole ensemble is trained on the dataset of skin lesion images, and their weight are saved for using them in the next step.

3. The next step is loading an **identical model** (the feature extractor plus a classifier) with the last unfrozen feature extractor convolutional block. This new network is loaded with the weights from the previous training, and then it is trained again.

4. Finally, we perform a fine-tuning by creating a new copy of the neural network, on which we unfreeze the whole model to retrain it with the new data using **a very low-speed learning rate**. This way, significant improvements can be achieved by adapting gradually the pre-trained features to the new data.

3.3.3 Search of Hyper-parameters

A neural network is managed by a large number of external parameters that need to be tuned to achieve **network convergence**. This is achieved when the output of the neural network does not register significant changes even though the training process continues, maintaining the weights and biases without significant changes. Thus, Hyperparameter search is the process of finding the optimal set of parameters that define the model to obtain the best metrics but also to achieve convergence in the shortest possible time.

The parameters that will be adjusted in our work, and that are generally used in the majority of deep learning networks are described as follows:

1. **Learning Rate**: It controls how much the model parameters are adjusted at each training step. A high learning rate can cause the model to converge quickly, but there is also a risk the network may be lost in a global minimum and it will not be able to complete its learning. Besides that, a low learning rate may lead to slow convergence or no convergence at all.

2. **Epochs number**: This parameter controls the number of times the learning algorithm iterates over the entire training data set. During each epoch, the weights of the neural network are updated in an attempt to minimize the Loss function, thus improving the network learning process for each epoch. The most common values for this parameter are 16, 32, and 64, and it will always be expressed in multiples of two.

3. **Batch Size**: The number of training samples used in a model update iteration. Its value and the dataset size determine the number of iterations required to complete an epoch. If its value is low, the model may not have enough time to learn the data patterns. On the contrary, a high epoch value may cause the model to be susceptible of overfitting by incorporating data noise into the adjustment weight.

4. **The optimizer**: It is the element responsible for calculating the cost function gradient in each network parameter/dimension. It is an algorithm that tries to minimize the error function by searching the minimum of this function. Therefore, it is a crucial element in the network learning process, and its choice will determine the network performance. There are several **gradient descent optimizers** in the market, but in this research, we have chosen three of the most used in classification works: **SGD**, **RMSProp** and **Adam**.

   Stochastic Gradient Descent (SGD): This optimizer limits the gradient calculation to one per batch. It is used as a baseline in order to compare it with the others.

   RMSProp: Just as the SDG optimizer performs the descent gradient over a batch. This second optimizer uses the gradient-weighted average calculations over a set of samples.

This optimizer introduces variability into the training factor concept at each iteration, unlike the SGD optimizer, which keeps it stable.

AdaGrad: Even if we will not use this optimizer for the current job, it is important to mention it because it adapts the learning rate for each parameter individually based on the historical gradients, making it particularly suitable for sparse data. Compared to SGD, Adagrad adjusts learning rates per parameter, sharing similarities with RMSProp.

Adaptative moment estimation (Adam): It inherits and combines the improvements of the AdaGrad and RMSProp algorithms while bringing in a gradient impulse averaging. This is similar to AdaGrad but performs the training factor scaling by a different formula (the exponential decay of the square of the gradients).

Furthermore, the behavior of SDG and RMSProp optimizer can be improved using the ***momentum*** parameter. This one is used to accelerate the gradient descent and soften its oscillations [49]. A reference value of 0.9 is used during the hyperparameter step.

As a complement, the following reference was added [50] showing the different optimizers operation graphically, as well as their corresponding results.

In order to find the most optimal parameters to train our model, we followed a **grid search** model. Thus, we set the number of **epochs up to 32** since this value is the most frequently used as a starting point. From this point on, we will try different learning rate combinations and optimizers.

As shown in figure 3.12, training the model with a low learning rate value and SGD or Adam optimizer is the best combination to guarantee a smooth model convergence. In the RMSProp case, the learning process seems to be more unpredictable than the previous one, and because of this, we discarded it for our job. After comparing SGD and Adam graph's results, we observed that obtained values with the **learning rate of 0.001**, the lowest one used, guaranteeing a stable learning long last time. Considering all the previous information and discarding between these two last optimizers, we will opt for the one that reaches the convergence point faster, that is **Adam**.

As shown in the graph 3.13, the process has also been performed in the second neuronal network based on ResNet50. In this grid search, we also used SGD, RMSProp, and Adam optimizers trained in the same learning rate range as in the previous process. As with the previous model, we discarded RMSprop from the outset because of the instability of the samples. In the case of SGD, training with the lowest learning rate values seems to offer stability, as does training with the Adam optimizer. However, we opted for the latter because we thought it would reach convergence faster.

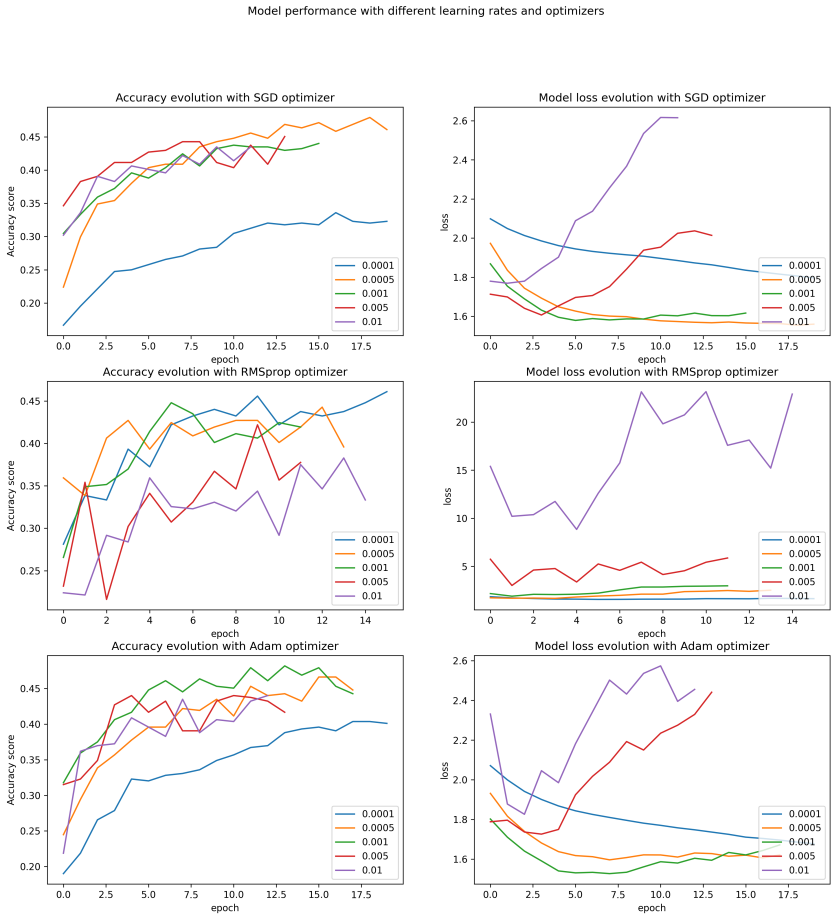


Figure 3.12: Efficient B0 base model hyperparameters search.

3.3.4 Evaluation Metrics

Since what is not measurable cannot be improved, we define a series of metrics that will allow us to know how well our models will classify. This will help us know the progress as we move forward with the transfer learning process throughout the development of the tests.

In a multiclass classification context, to visualize the successes or failures of the prediction, we compare the obtained values with the real ones in a confusion matrix, as shown in the figure 3.1. The main diagonal of the matrix shows the True Positives (TP). In this representation, we have focused on one of the classes (MEL) to show the False Positives (FP) or those predictions that are not correct. With the same class, we can see those images belonging to the MEL class have been classified to a different class and, therefore, will be False Negative (FN).

The main metrics to be used are the following:

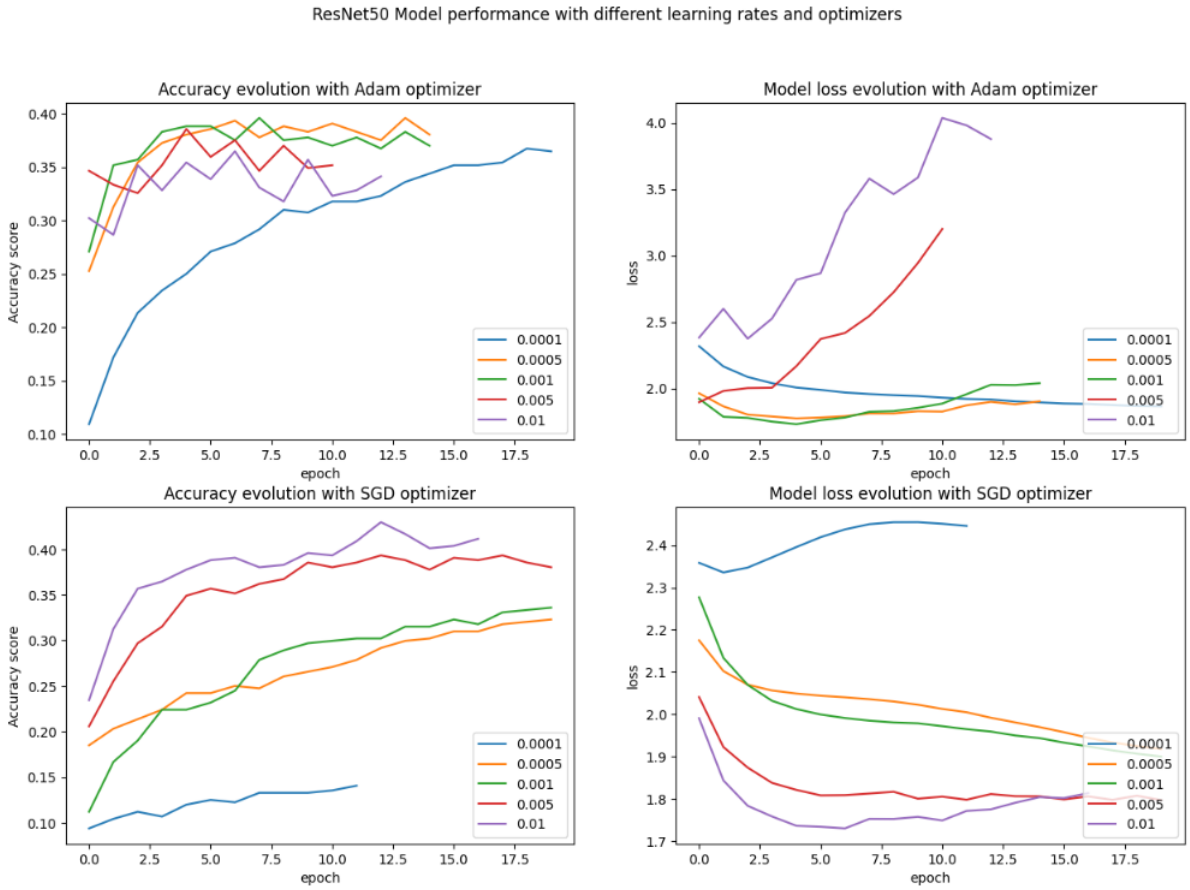


Figure 3.13: ResNet50 base model hyperparameters search.

1. **Accuracy** (Acc): A metric that measures a model's predictions' overall level of correctness. It is calculated as the ratio of the number of correct predictions, positive and negative, to all predictions made.

$$Acc = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

2. **Precision** (Prec): Measures the proportion of correctly identified positive instances among all instances identified as positive. It is calculated as the ratio of true positives to the sum of true positives and false positives.

$$Prec = \frac{(TP)}{(TP+FP)}$$

3. **Sensitivity** (Sen): Sensitivity measures the ability of a model to identify positive cases among all true positives. It is a metric often called Recall and is calculated as the ratio of True Positive predictions to the sum of True positive and False Negative predictions.

$$Sen = Rec = \frac{(TP)}{(TP+FN)}$$

		Prediction							
		AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
Real	**AK**	TP							
	BCC		TP			FP_1			
	BKL			TP		FP_2			
	DF				TP				
	MEL		FN_1	FN_2		TP			
	NV						TP		
	SCC							TP	
	VASC								TP

Table 3.1: Multiclass matrix example

4. **Specificity** (Esp): This metric indicates the ability of a model to correctly identify negative cases out of the total number of true negatives. It is calculated as the ratio of true negative predictions to the sum of true negatives and false positives.

$$Esp = \frac{(TN)}{(TN+FP)}$$

5. **F1-score** (F1): When evaluating classifiers, trying to estimate the value of false positives and negatives is sometimes complex. The F1-score metric allows us to gain insight into classifier model performance where accuracy and Recall may not represent overall model performance. This metric is calculated as the harmonic mean between precision and Recall, and thanks to this type of calculation, it is a metric that is not as affected when there are imbalances between classes. These penalizing models have unequal performance between positive and negative class identification.

$$F1 - score = 2 * \frac{(Prec*Sen)}{(Prec+Sen)} = \frac{(TP)}{(TP+\frac{1}{2}(FP+FN)}$$

In a multiclass classification context such as the one we are working on in this article, the calculation is performed with the One-vs-Rest (OvR) strategy, in which the value is calculated for each class separately. Subsequently, the values are averaged to extract the overall value of the metric. However, we are faced with three possibilities: Performing an arithmetic average across all classes, in which case we speak of F1 calculated with Macro Average. Another option would be to use a weighted average to penalize possible imbalances in the dataset. This second case is known as the F1 Weighted Average. The third case, perhaps the most interesting of all, sums the values of TP, FP, and FN for all classes, using the second of the formulas for the calculation. This metric we will use in our study is called the F1-score for Micro Average.

3.3.5 Skin lesion classification with EfficientNet B0

Traditionally, high accuracy in image classification has been achieved by increasing the number of layers in convolutional neural networks (ConNets). In 2020, Mingxing Tan and Quoc V. Le rethink the scaling-up process in this type of usage in the article "*EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*" [51]. To understand the process, we need to remember that any neural network can be characterized as three dimensions: depth, width, and resolution. In the paper cited above, the authors observe empirically that **these dimensions are not independent** and they are related to each other. Therefore, when scaling the network, **it is necessary to maintain a balance between the dimensions to achieve greater accuracy and efficiency**. For this reason, the authors propose a new composite scaling method with the aim of being able to scale the dimensions in a uniform way and thus increase the network efficiency. Finally, the work succeeds in obtaining a new neural network family model called EfficientNets, which is composed of eight variations from B0 to B7. Each variation increases the number of layers, from 213 in EfficientNet B0 to 817 in version B7 [52].

In this chapter, we will perform transfer learning using the least parameterized network of the EfficientNet family, the B0. In order to do this, we are going to follow the steps that have been described at the beginning of the chapter 3.3.2.

3.3.5.1 Step 1. Base model froze

As mentioned above, the 'ImageNet' weights will be loaded into the base model while the feature extractor is frozen in the first stage of the learning transfer process. Training is then performed using the optimal parameters determined during the hyperparameter search, which include a learning rate of 0.001, a batch size of 32, 40 epochs, and a patience factor of 10.

The model's training efficiently evolved and reached convergence on the 40th epoch during testing. A difference was observed when training the model using the extended dataset with synthetic images through SMOTE 3.14. In this case, **the model became more stable**, even though its training became a little slower and with a bit less accuracy for the same training time, a fact that we will better evaluate by visualizing the metric tables at the end of this section.

In figure 3.15, it is possible to compare the confusion matrices trained on the two datasets, revealing a subtle but important change when designing classifiers in the medical world, where classifiers are intended to minimize prediction errors that could lead to misdiagnosis. Thus, by examining the figure on the right, one can see **how false positives have been reduced in some of the classes**. This fact is especially relevant with VASC, where, in the SMOTE version of the dataset, these false positives decrease considerably with respect to the non-

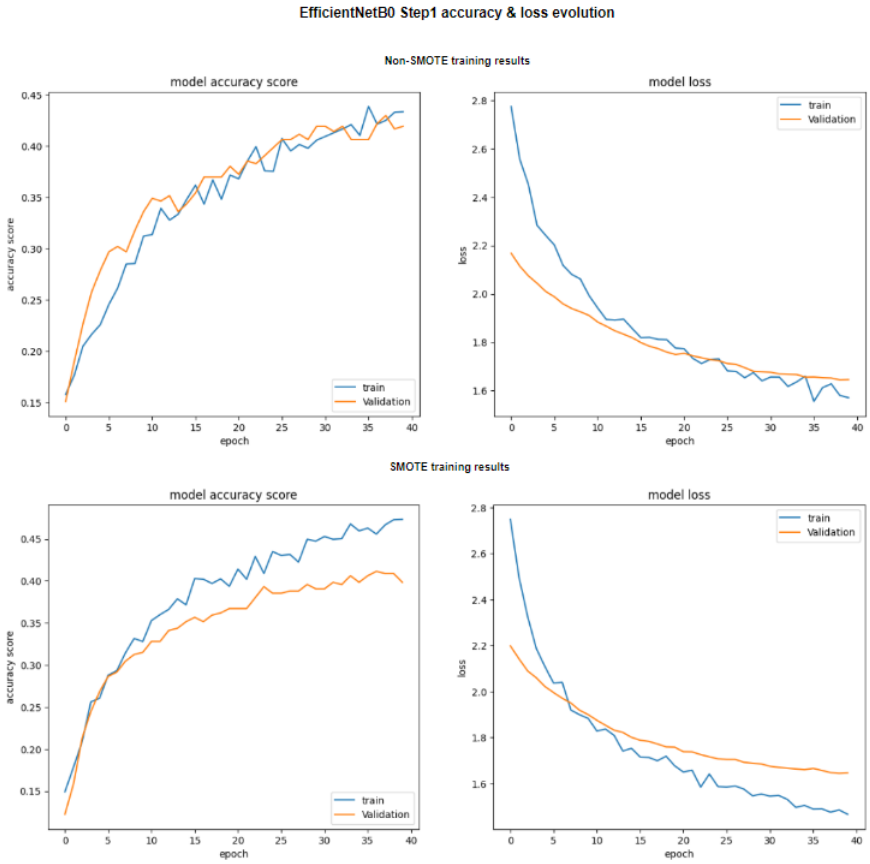


Figure 3.14: EfficientNetB0 model with feature extractor freeze. Accuracy-loss diagram with and without SMOTE.

SMOTE-enriched dataset.

	Values with dataset without SMOTE								Values with SMOTE dataset							
	AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**	**AK**	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
precision	0.30	0.54	0.34	0.56	0.37	0.33	0.18	0.60	0.31	0.38	041	0.82	0.42	0.29	0.13	0.81
sensitivity	0.26	0.40	0.22	0.44	0.54	0.60	0.14	0.52	0.22	0.38	0.52	0.36	0.50	0.66	0.08	0.26
F1 score	0.28	0.46	0.27	0.49	0.44	0.42	0.16	0.56	0.26	0.38	0.46	0.50	0.45	0.41	0.10	0.57

Table 3.2: EffcientNet B0 Step 1. Metrics obtained by class.

The table 3.2 shows the numerical representation of each metric for each of the classes. As mentioned in the previous paragraph, when looking at the confusion matrices, we see a 21% increase in classification accuracy for the VASC class when using SMOTE, as well as for the DF class, which has a significant increase in this metric. However, the F1 score for both classes increases by one percentile due to the decrease in sensitivity. Let us remember that the F1 metric, at a general level, will give a more balanced indication of the classifier's ability to discriminate between healthy and sick people, taking into account both the correct positive predictions and the ratio of all positive predictions, whether true or not. This statement can

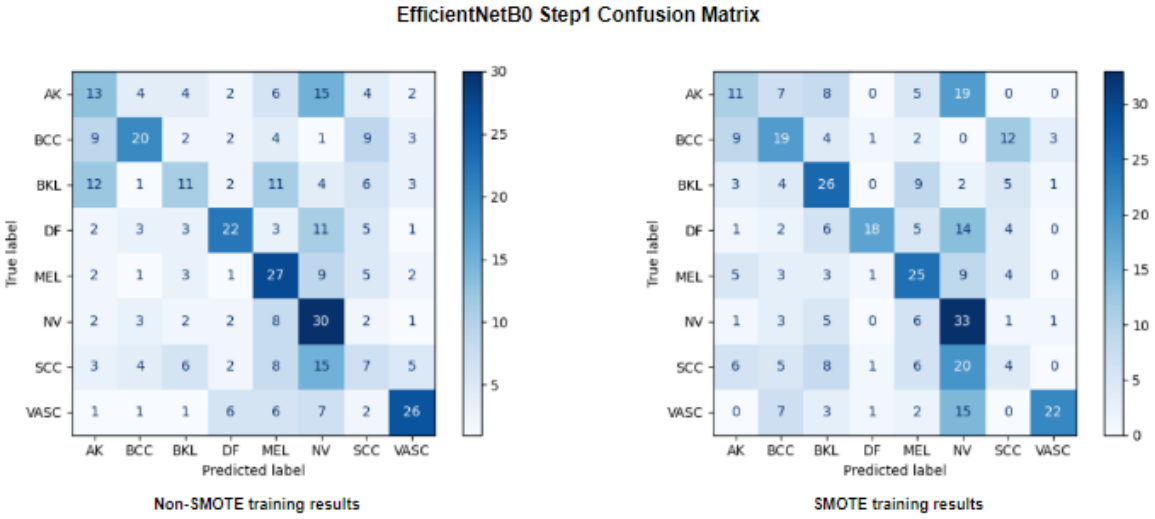


Figure 3.15: EfficientNetB0 model with feature extractor freeze. Confusion matrix.

also be seen in the table 3.3, where despite a 5-point increase in accuracy obtained by training with the SMOTE-treated dataset, it only translates into a slight increase of one point in its F1-score.

	Accuracy	**Precision**	**sensitivity**	**F1-score**
Values without SMOTE dataset	0.39	0.40	0.39	0.38
Values with SMOTE dataset	0.40	0.45	0.40	0.39

Table 3.3: EffcientNet B0 Step 1. Global Metrics.

3.3.5.2 Step 2. Unfroze Model base last block

The next step to improve the metrics of our model will be to unfreeze the last convolutional block, the feature extractor, and transfer the learning obtained in step one by transferring its training weights. The base model's last convolutional block (Conv 7) is formed for 16 layers, and it has 4 more from the classification layers. For this, the total number of layers to be trained will be 20.

The training will be carried out with the next parameters: a learning rate of 0.001, a batch size of 32, and a patience factor of 10. During this stage of the transfer learning process, the model is prone to overfitting quickly. To avoid this, training is limited to eight epochs.

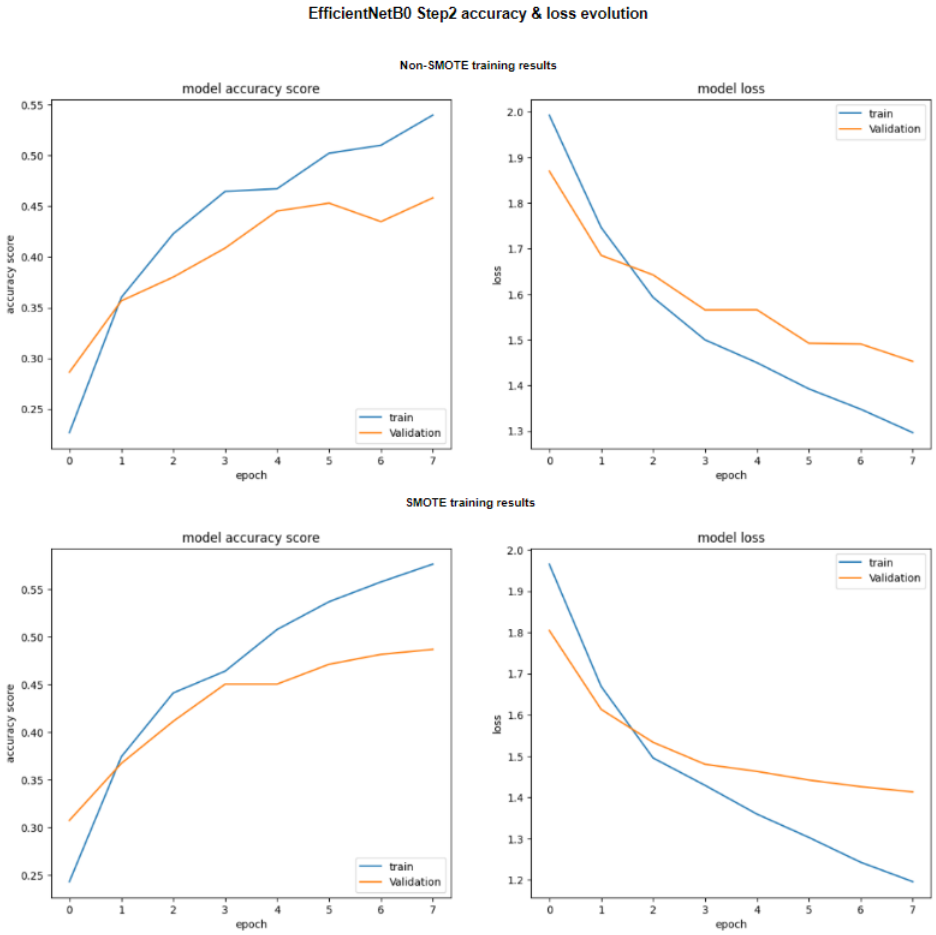


Figure 3.16: EfficientNetB0 model with the last block unfroze. Accuracy-loss diagram.

Examining the graphs of the training with both datasets 3.16, we also observe that although the metrics obtained are similar, the model trained with the SMOTE dataset shows more regularity. This can be followed in epoch six, in the graph showing the accuracy, and in epoch four, in the graph showing the evolution of the loss function.

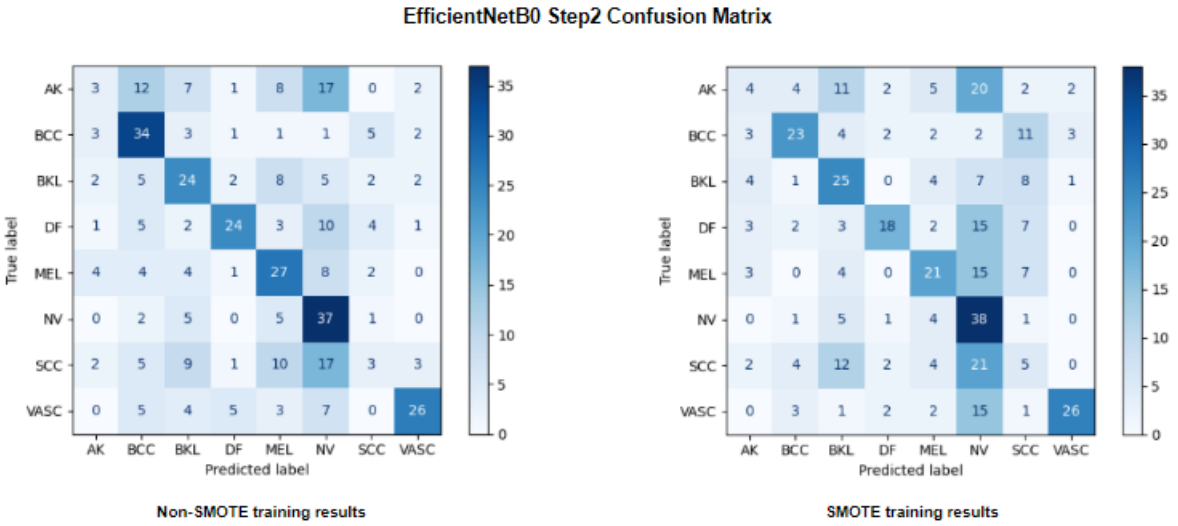


Figure 3.17: EfficientNetB0 model with the last block unfroze. Confusion matrix.

The comparison between the training confusion metrics 3.17 of both datasets shows an increase of false positives with some classes when training the dataset with SMOTE, as is the case of BKL.

	Values with dataset without SMOTE								Values with SMOTE dataset							
	AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**	**AK**	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
precision	0.2	0.47	0.41	0.69	0.42	0.36	0.18	0.72	0.21	0.61	0.38	0.67	0.48	0.29	0.12	0.81
sensitivity	0.06	0.68	0.48	0.48	0.54	0.74	0.006	0.52	0.08	0.46	0.50	0.36	0.42	0.76	0.10	0.52
F1 score	0.009	0.55	0.44	0.56	0.47	0.49	0.09	0.61	0.12	0.52	0.43	0.47	0.45	0.42	0.11	0.63

Table 3.4: EffcientNet B0 Step 2. Metrics obtained by class.

Comparing the training between the two datasets 3.4, in general, there is no significant improvement when training with the SMOTE-enriched dataset. This is also supported by examining the global metrics 3.5, where a metrics degradation is observed when training with this dataset.

	Accuracy	**Precision**	**sensitivity**	**F1-score**
Values without SMOTE dataset	0.45	0.43	0.44	0.41
Values with SMOTE dataset	0.40	0.45	0.40	0.39

Table 3.5: EffcientNet B0 Step 2. Global Metrics.

3.3.5.3 Step 3. Unfroze the model

During the final stage of our training, we will **unfreeze the entire model**, including both the feature extractor and the classifier, and the weights from the previous training will be loaded. In order to improve the network metrics, we will perform the training with a **lower learning rate**. In this way, the training parameters used will be similar to those used in previous training sessions, except for the learning rate, which is set at 0.0001. The training duration is fixed at 40 epochs due to overfitting appearing after this time.

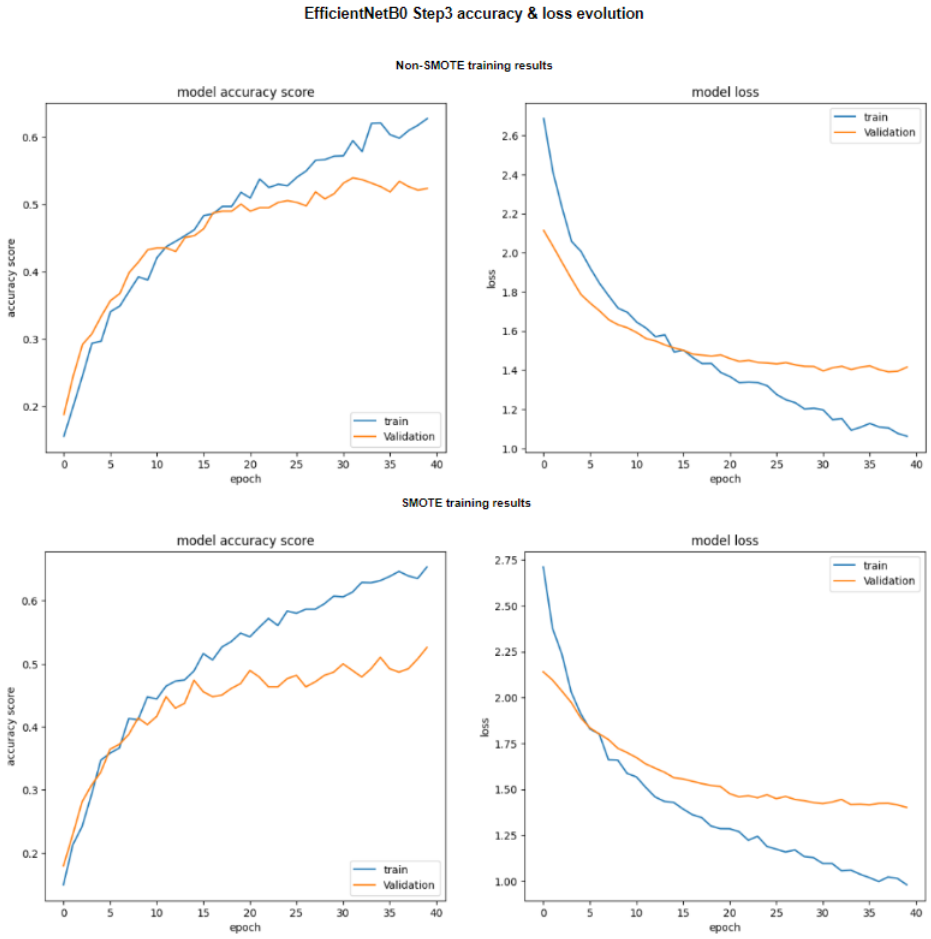


Figure 3.18: EfficientNetB0 model. Training all the model layers. Accuracy-loss diagram.

The accuracy and loss function evolution graphs 3.18 in this third round of training do not show a substantial difference when using one dataset or the other.

A comparison of the confusion matrices 3.19 shows that, although slightly, the **model trained with the dataset without synthetic images achieves better results**, both in identifying true positives (diagonal of the matrix) and false positives. However, if we look at the Dermatofibroma class (DF) whose population has been synthetically increased, we can observe

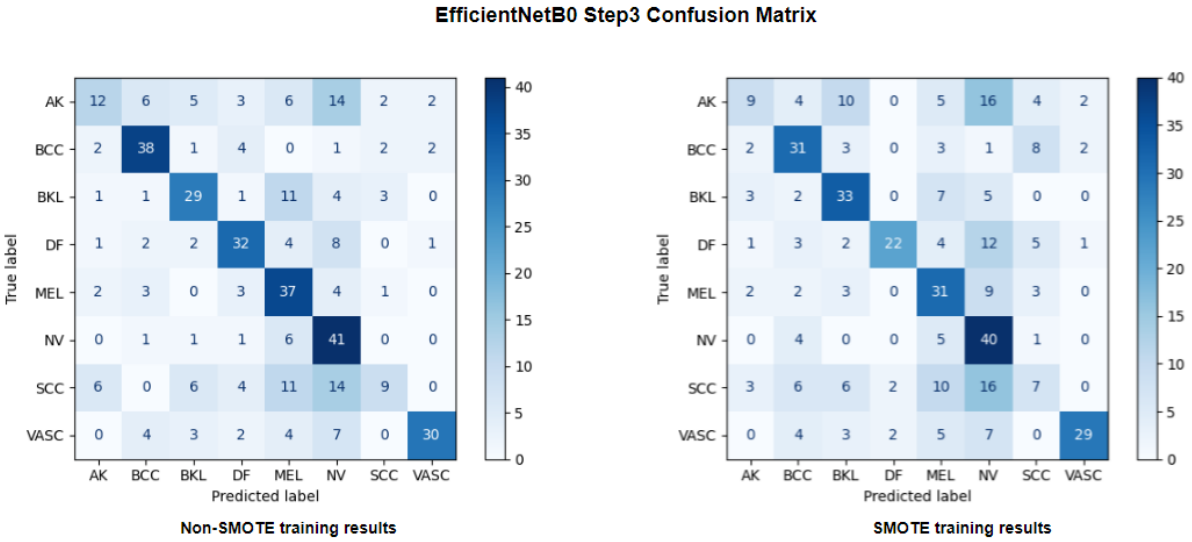


Figure 3.19: EfficientNetB0 model. Training all the model layers. Confusion matrix.

a considerable reduction in false positives.

	Values with dataset without SMOTE							
	AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
precision	0.50	0.69	0.62	0.64	0.47	0.44	0.53	0.86
sensitivity	0.24	0.76	0.58	0.64	0.74	0.82	0.18	0.60
F1 score	0.32	0.72	0.60	0.64	0.57	0.57	0.27	0.71

Values with SMOTE dataset							
AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
0.45	0.55	0.55	0.86	0.44	0.38	0.25	0.85
0.18	0.62	0.66	0.44	0.62	0.80	0.14	0.58
0.26	0.58	0.60	0.58	0.52	0.51	0.18	0.69

Table 3.6: EffcientNet B0 Step 3. Metrics obtained by class.

An examination of the two tables 3.6 3.7 numerically corroborates the claim that the EfficientNet B0-based model performs better with the normal dataset without synthetic sample enrichment, achieving an accuracy of 57% with an F1-score value of 55%.

	Accuracy	**Precision**	**Sensitivity**	**F1-score**
Values without SMOTE dataset	0.57	0.59	0.57	0.55
Values with SMOTE dataset	0.51	0.54	0.50	0.49

Table 3.7: EffcientNet B0 Step 3. Global Metrics.

3.3.6 Skin lesion classification with ResNet50

The second neural network we will use in this work is the one known as ResNet50, a deep neural network composed of 48 convolutional layers, more than one MaxPool, and an average pool layer, which is one of the most popular ConNets used for image classification. This network

was developed in 2015 as a result of the study "*Deep Residual Learning for Image Recognition*". [53], in which the authors presented a residual learning framework with a deep neural network that is easily trainable and avoids the problem of gradient descent **vanishing** as the network gets deeper. Consequently, as the number of layers increases, the neural network can learn more features, but accuracy can slowly degrade due to saturation. The concept behind this neural network is the use of the residual function, which creates a shortcut to the output of the layer by adding the output of the previous layer. At depth, the residual blocks act by feeding back the gradient, preventing fading and allowing for deeper ConNets.

3.3.6.1 Step 1. Base model froze

The strategy for training this network is the same as the one used for the EfficienteNetB0-based network. In the first step, we train with all base model layers frozen. The training parameters to be used are a fixed learning rate of 0.001, a batch size of 32, a patience parameter of 10, and a training duration of 30 epochs. A transformation is applied to the source image matrix to ensure that the ResNet neural network has a values matrix between [-1, 1].

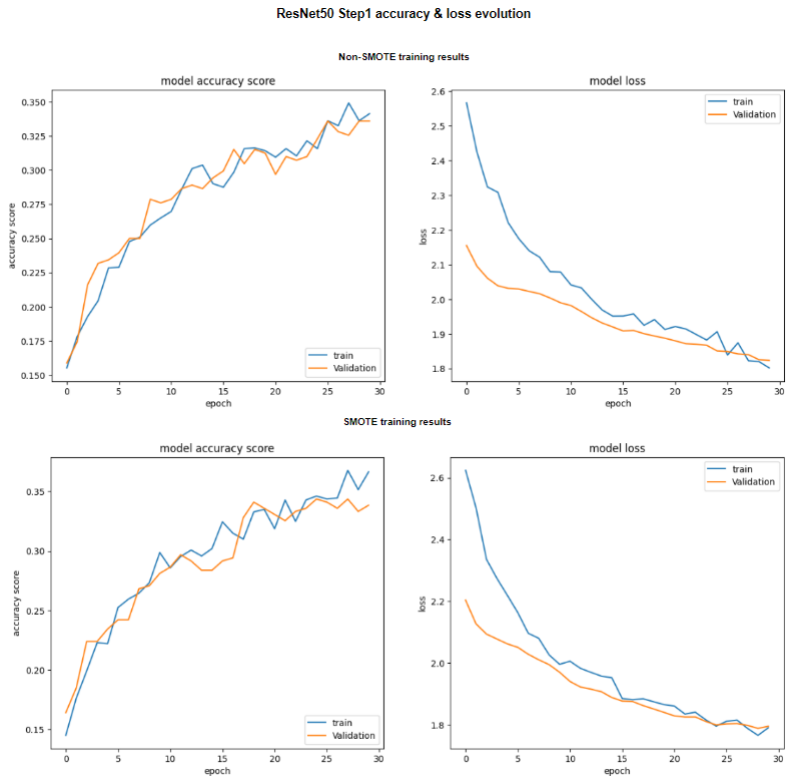


Figure 3.20: ResNet50 model with feature extractor freeze. Accuracy and loss graph.

Looking at the graph of accuracy and loss function evolution 3.20, there is no significant

difference between training with the SMOTE augmented dataset and the first training step with EfficientNet. Both graphs converge to the same values for similar training times, and show marked jumps in the accuracy progression.

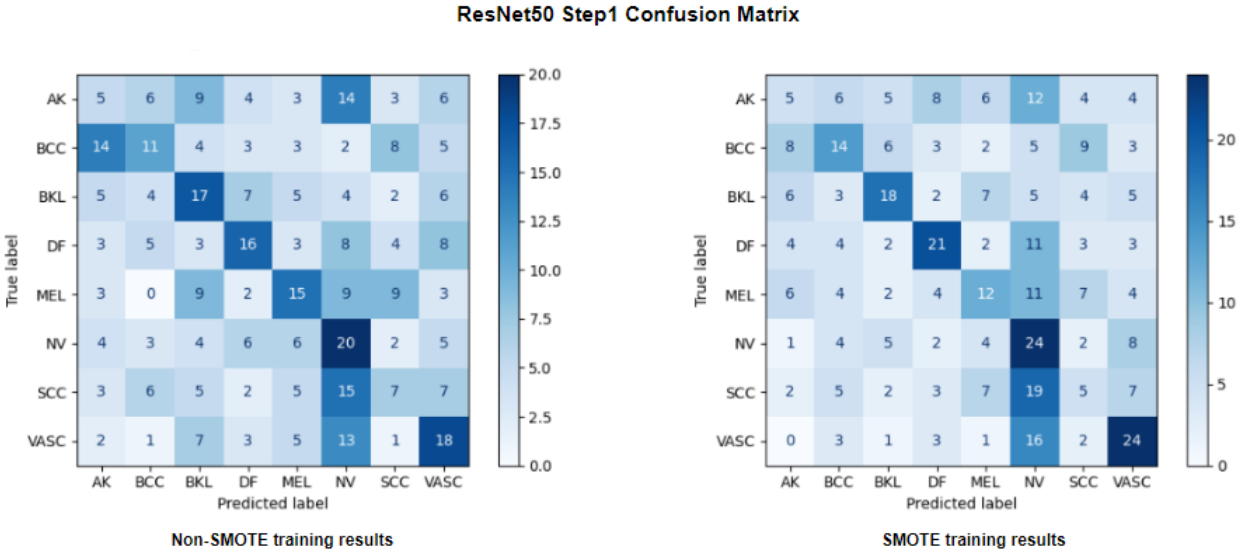


Figure 3.21: ResNet50 model with feature extractor freeze. Confusion matrix graph.

Comparison of the confusion matrices 3.21 shows an improvement in accuracy, the true positives, in training with the SMOTE-enriched dataset.

	Values with dataset without SMOTE								Values with SMOTE dataset							
	AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**	**AK**	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
precision	0.13	0.31	0.29	0.37	0.33	0.24	0.19	0.31	0.16	0.33	0.44	0.46	0.29	0.23	0.14	0.41
sensitivity	0.10	0.22	0.34	0.32	0.30	0.40	0.14	0.36	0.10	0.28	0.36	0.42	0.24	0.48	0.10	0.48
F1 score	0.11	0.26	0.31	0.34	0.2	0.30	0.16	0.33	0.12	0.30	0.40	0.44	0.26	0.31	0.12	0.44

Table 3.8: ResNet50 Step 1. Metrics obtained by class.

Examining the results in the table 3.8, it is possible to observe that all metrics results are higher for almost all classes. This fact is also observed in the table 3.9 showing the global prediction results.

	Accuracy	**Precision**	**Sensitivity**	**F1-score**
Values without SMOTE dataset	0.27	0.27	0.27	0.27
Values with SMOTE dataset	0.31	0.30	0.31	0.31

Table 3.9: ResNet50 Step 1. Global Metrics.

Unlike the previous table, this one 3.9 shows the indicators calculated at the global level without distinguishing between classes. It can be seen that working with the SMOTE-enriched dataset gives us a slight improvement in all the indicators.

3.3.6.2 Step 2. Unfroze Model base last block

For the second step of the ResNet training, the feature extractor convolutional block's last layers will be unfrozen. In total, adding the last 36 layers of the convolutional block five and the 4 layers of the classifier, 40 layers of the neural model will be trained. The parameters to be used will be the same as in the previous step, except for the duration, which will be set at 5 epochs.

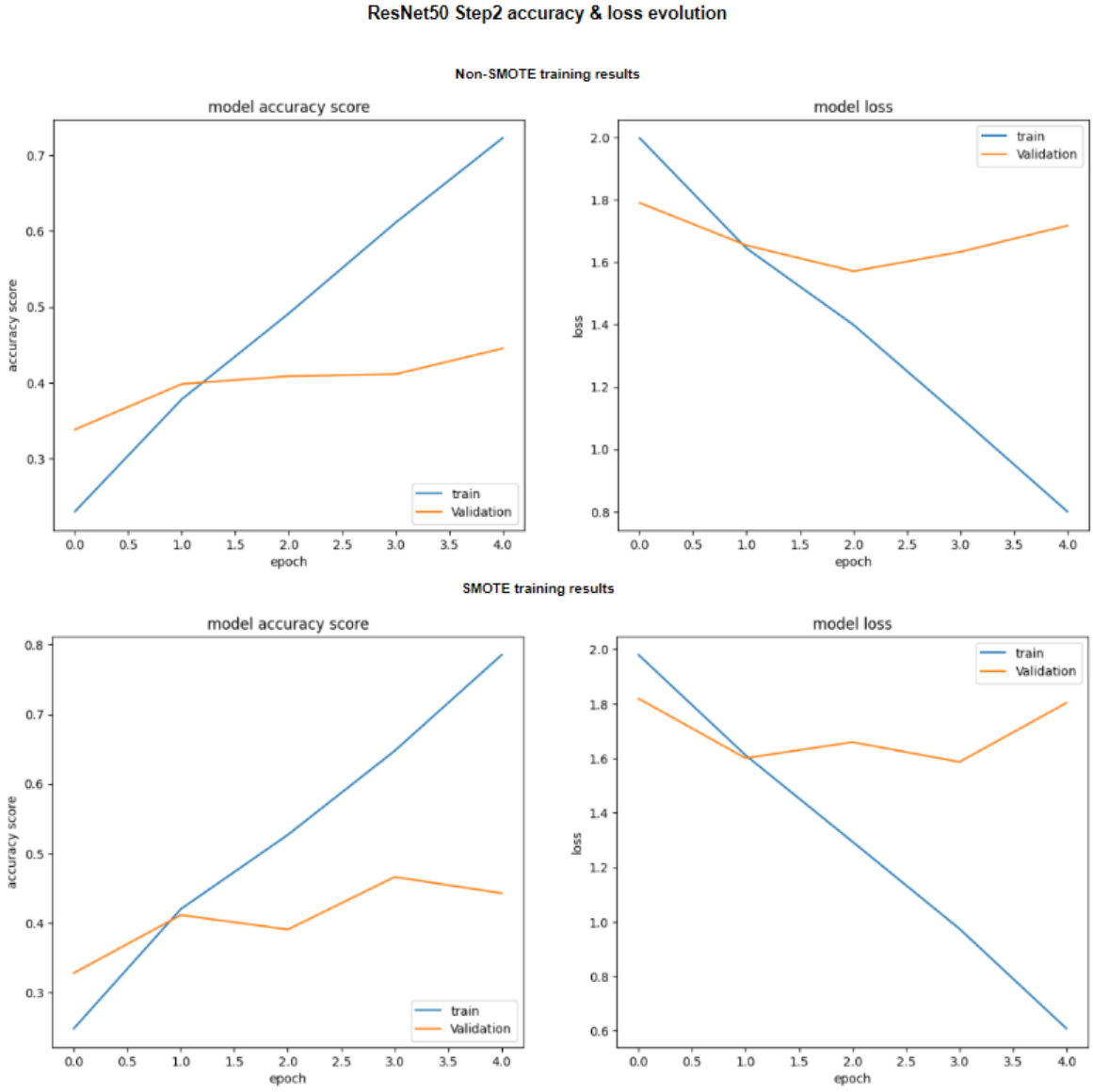


Figure 3.22: ResNet50 model with the last block unfroze. Accuracy and loss graph.

The graphs 3.22 show how overfitting appears already after the first training rounds, around the third epoch, with both datasets.

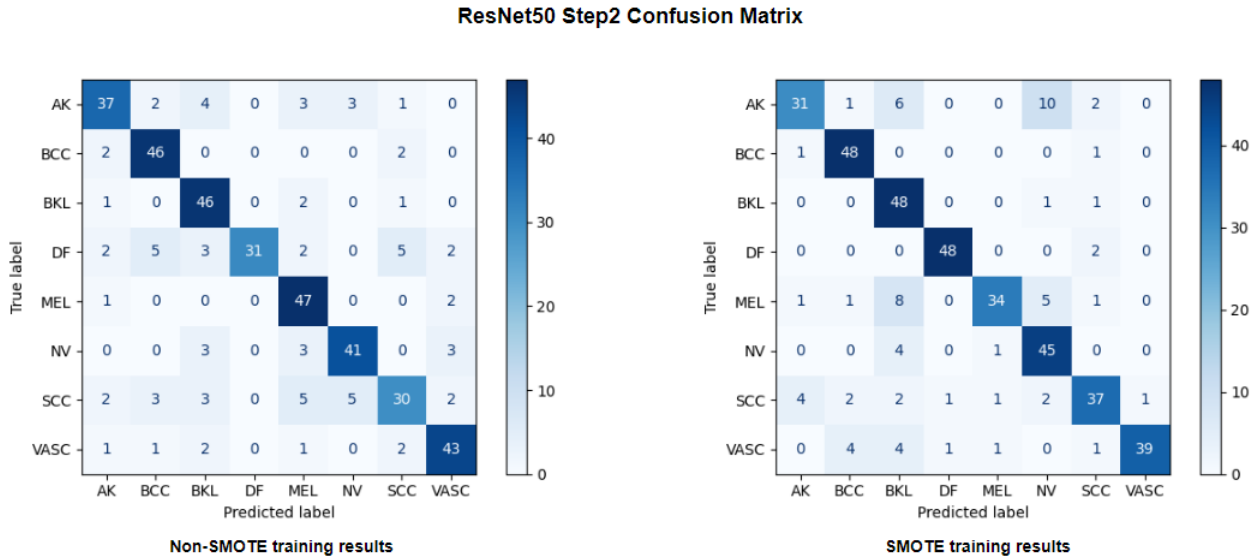


Figure 3.23: ResNet50 model with the last block unfroze. Confusion matrix graph.

The confusion matrices 3.23 show a very positive evolution with respect to the values obtained in step 1. The comparison between the two graphs shows that the true positives have increased significantly. For example, both *Basall Cell Carcinoma* (BCC) and *Benign Keratosis* (BKL) classes with the SMOTE dataset obtained 48 correct images while the values for these classes in step 1 were 14 and 18 images, respectively. False positives have also been reduced considerably, being 1 in the case of *Vascular lesion* (VASC) for the dataset augmented with synthetic images.

	Values with dataset without SMOTE								Values with SMOTE dataset							
	AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**	**AK**	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
precision	0.80	0.81	0.75	1.00	0.75	0.84	0.73	0.83	0.84	0.86	0.67	0.96	0.92	0.71	0.82	0.98
sensitivity	0.74	0.92	0.92	0.62	0.94	0.82	0.60	0.86	0.62	0.96	0.96	0.96	0.68	0.90	0.74	0.78
F1 score	0.77	0.86	0.83	0.77	0.83	0.83	0.66	0.84	0.71	0.91	0.79	0.96	0.78	0.80	0.78	0.87

Table 3.10: ResNet50 Step 2. Metrics obtained by class.

Examining the results by class 3.10, it can be seen that for some classes, better results are obtained with the dataset without SMOTE, while others obtain better results using this dataset. It is remarkable that the *dermatofibroma* (DF) class, which in the first case obtained an accuracy of 100%, but when trained with SMOTE, its sensitivity and, therefore, its F1 score increased. The opposite is the case with the *vascular* (VASC) class, in which the accuracy is increased to 98% using SMOTE, with a reduction in sensitivity. However, for this class, the classifier performs better with SMOTE, achieving an F1-score value three points higher.

In general, the model achieves better results when trained with the SMOTE-enriched dataset, which can be seen in the table 3.11.

	Accuracy	Precision	sensitivity	F1-score
Values without SMOTE dataset	0.80	0.81	0.80	0.80
Values with SMOTE dataset	0.83	0.84	0.82	0.82

Table 3.11: ResNet50 Step 2. Global Metrics.

3.3.6.3 Step 3. Unfroze the model

In the last step of our training, as shown in the description of the Transfer of Learning steps 3.3.2, we will train our model in all its layers, starting from the weights inherited from the previous step and, as in step 3 of the EfficentNet training, we will do it with a lower learning rate, set to 0.0001, keeping the rest of the parameters except for the number of epochs, which will be set to eight.

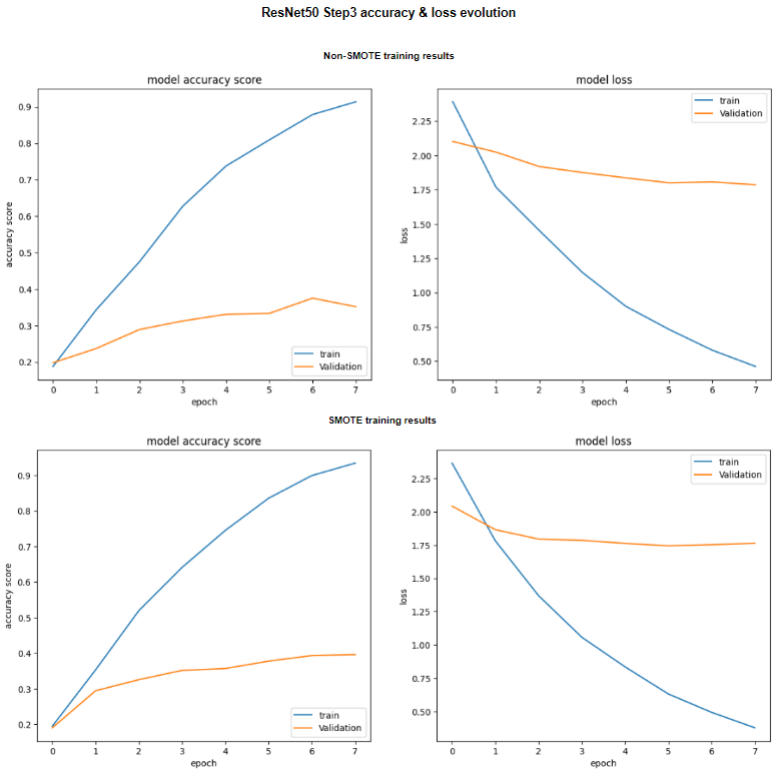


Figure 3.24: ResNet50 model. Training all the model layers. Accuracy and loss graph.

Analysis of the plots 3.24 shows that although the training values appear to be good if we inspect the validation data, we see that there is practically no training beyond the fifth epoch, and even though it is not shown in the tests, it was seen that the network shows overfitting from the eighth epoch and beyond.

The values shown by both confusion matrices 3.25 are similar and also very good. It

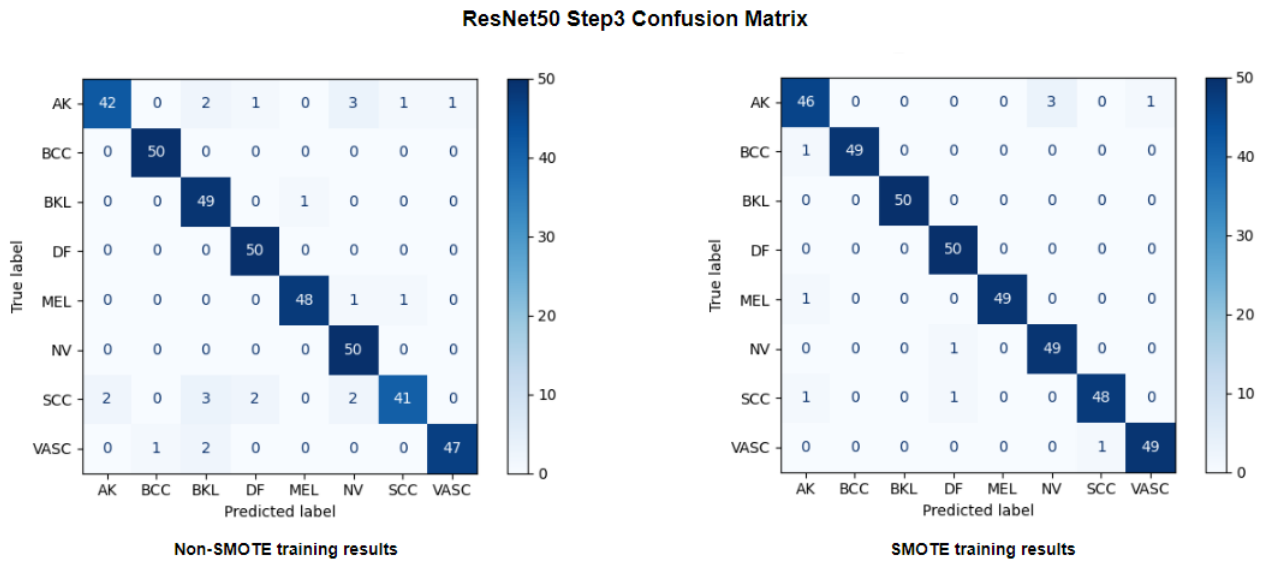


Figure 3.25: ResNet50 model. Training all the model layers. Confusion matrix graph.

is remarkable that those obtained by training with the SMOTE-enriched dataset show several classes where false positives, *Basal cell carcinoma* (BBC), *benign keratosis* (BKL), and *Melanoma* (MEL), have been reduced to zero, and two other classes, *Squamous cell carcinoma* (SCC) and *Vascular lesion* (VASC), show a single case.

	Values with dataset without SMOTE								Values with SMOTE dataset							
	AK	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**	**AK**	**BCC**	**BKL**	**DF**	**MEL**	**NV**	**SCC**	**VASC**
precision	0.95	0.98	0.88	0.94	0.98	0.89	0.95	0.98	0.94	1.00	1.00	0.96	1.00	0.94	0.98	0.98
sensitivity	0.84	1.00	0.98	1.00	0.96	1.00	0.82	0.94	0.92	0.98	1.00	1.00	0.98	0.98	0.96	0.98
F1 score	0.89	0.99	0.94	0.97	0.97	0.94	0.88	0.96	0.93	0.99	1.00	0.98	0.99	0.96	0.97	0.98

Table 3.12: ResNet50 Step 3. Metrics obtained by class.

The results obtained when examining the metrics for each class 3.12 show that the F1-score reaches 100 % in the case of *benign keratosis* (BKL), and in others, it is between 98 and 99 %. The accuracy for *melanoma* detection (MEL), as well as for *Basal Cell Carcinoma* (BCC) and the *benign keratosis* (BKL), is 100 %.

	Accuracy	**Precision**	**sensitivity**	**F1-score**
Values without SMOTE dataset	0.94	0.94	0.94	0.94
Values with SMOTE dataset	0.98	0.98	0.98	0.98

Table 3.13: ResNet50 Step 3. Global Metrics.

The global metrics 3.13 show that the classifier trained following the transfer learning process shows a better response when trained with the SMOTE-based dataset, achieving an F1-score of 98 %.

Chapter 4

Summary and Conclusion

The success or failure of a machine learning or deep learning process is directly proportional to the quality of the data used. The data source for this work, the *2019 ISIC dataset*, presented several problems that required some treatment application to ensure the **elimination of biases** and to improve the project's viability. On one hand, the data was collected from different medical sources with different resolutions. On the other hand, there was a very significant difference in the number of elements per class, resulting in a very **unbalanced dataset**. In the first case, the images were transformed to the resolution required by the neural networks. In the second case, two different strategies were used: One consisted of trimming the samples according to the maximum number in the minority class (under-sampling technique). This approach is easy to apply, although its disadvantage causes the loss of richness by discarding data that could improve the model's metrics. To compensate this, we have opted for a second option: enriching the minority classes with synthetic images using SMOTE (**over-sampling** technique). This procedure has allowed us to increase the total number of images processed from 250 per class to 300, an increase of 20%. In both options, it has been possible to train the models with a balanced dataset.

To train the models, we used the **transfer learning** techniques. For this purpose, a classifier has been assembled using a **feature extractor** based on two well-known networks on the market, *EfficientNet B0* and *ResNet50*. This architecture was loaded with the weights of the *ImageNet* dataset. This has allowed us to take advantage of the extractor's pattern recognition capabilities, complemented with a classifier stage. During the evolution of this work, the three steps of the transfer learning process have been demonstrated, showing in the second step, a substantial improvement of the network classification metrics was achieved.

In addition to the previous, two assembled networks have been tested with the two treated datasets; the first has been pruned, and the second the number of samples has been increased artificially employing SMOTE. As a result, we have observed while the EfficientNet-based

classifier does not show any significant difference between training with one or the other dataset, the ResNet50-based network improves its metrics when trained with the SMOTE-based dataset. One possible explanation for this can be found by comparing the **depth of these networks**. Indeed, if we examine the right-hand column of the table 4.1, we can see a significant difference in the number of training parameters ResNet50 versus EfficienNet B0. Thus, a deeper network such as ResNet would benefit from training with a larger dataset.

4.1 Metrics summary

	Acc		Sen		Esp		F1-score		Time (sec)				
Model name	No SMOTE	With SMOTE	No SMOTE	With SMOTE	No SMOTE	With SMOTE	No SMOTE	With SMOTE	No SMOTE	With SMOTE	# Epochs	# Trainable params	# Total params
model ENetB0 1	0.39	0.40	0.40	0.45	0.39	0.40	0.38	0.39	3191	3560	40	12808	4064939
model ENetB0 2	0.45	0.40	0.43	0.45	0.44	0.40	0.41	0.39	2017	625	8	1361208	4064939
model ENetB0 3	0.57	0.51	0.59	0.54	0.57	0.50	0.55	0.49	2925	3244	40	4020356	4064939
model RNet1	0.27	0.31	0.27	0.31	0.27	0.31	0.27	0.30	2851	2266	30	20488	23589384
model RNet2	0.80	0.83	0.81	0.84	0.80	0.82	0.80	0.82	179	210	5	15250440	23589384
model RNet3	0.94	0.98	0.94	0.98	0.94	0.98	0.94	0.97	354	497	8	23539848	23589384

Table 4.1: Table summarising each model metrics

This table 4.1 shows the global metrics values of different training sessions. The first column on the left shows the name followed by a suffix indicating the step within the transfer learning process pointed out in the section 3.3.2. We included two columns for each metric reflecting the obtained value when the model was trained with the balanced dataset after under-sampling (identified as "No SMOTE"). The second one is when the training was performed with the extended dataset by over-sampling (identified as "With SMOTE"). In addition to the classical classification metrics, another column was inserted with the training duration and the epochs number. We observed that calculating the quotient between both values could allow us to extract the time per epoch, which can help making a prediction of the training duration in case it would be necessary to extend the number of epochs required.

Also, in this table, as we examined the accuracy and **F1-score** the first model values, it can be empirically proven that it did not improve when it was trained with the SMOTE-enriched dataset. The opposite can be seen when observing the values evolution on the second network, with a significant increase between the first and second steps, achieving good results in the third step. A comparison among the obtained results with both networks can be seen in the last step, the third. We saw a significant difference among them, which we obtained using EfficientNet B0, in the best case, with 57 % accuracy, compared to 98 % obtained by the ResNet50 network. Furthermore, in the third step, this model obtained **well-balanced metrics**, both in sensitivity and specificity, translating them into a high F1-score value; because of that, we considered it an excellent multiclass classifier.

4.2 Sample prediction

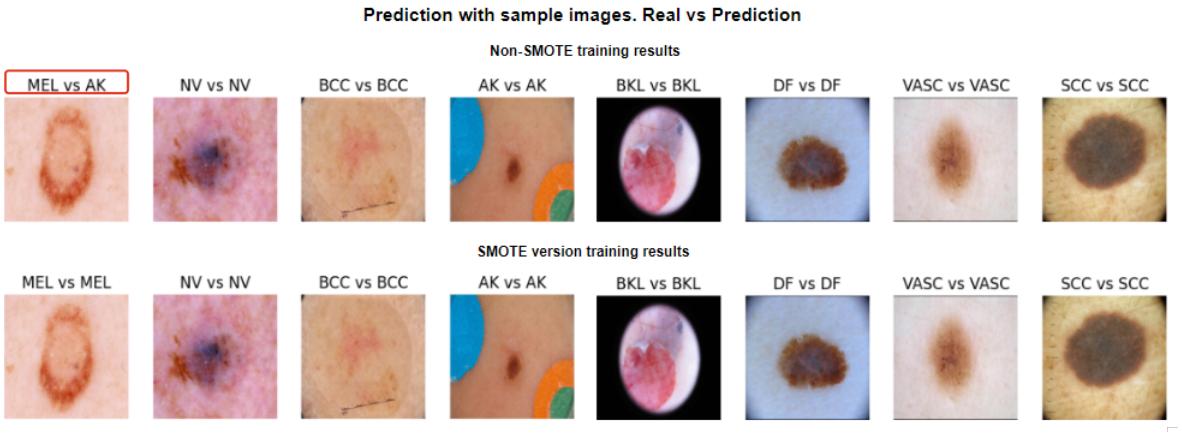


Figure 4.1: Test with a selection of sample images. Actual vs. Prediction Comparison

Finally, we have tested the ResNet50-based model predictive capacity, which provided the best results in our study. In order to test it, one image from each class of the test dataset was randomly selected from the test dataset. To follow the same line as the one used throughout our work, two models were used: the model trained with the dataset that has been sub-sampled and the one that was over-sampled with SMOTE. The results can be seen in the figure 4.1. In the first image line, we have the predictions obtained after using the model trained with the first dataset, obtaining a success rate of seven out of the eight samples. In fact, to be easily distinguished, we marked the wrong value in a red box. The second row shows the predictions obtained using the same trained model with the dataset treated with SMOTE, where the Convolutional network obtained a success of 100 %

4.3 Suggestions

We would like to propose several lines of action to continue the work done:

- The first and most urgent action is to increase the number of synthetic samples performed with SMOTE. The hypothesis to prove or refute would be to see the maximum value of synthetic samples possible before the network stops learning due to the sample degradation for the feature repetition, so as to find the ceiling of this over-sampling technique.

- The second item proposed is to perform tests using deeper networks different from ResNet 50. For example, a variant of the ResNet with 101 layers [54] doubles the number of the parameters at about 44 million. It is also possible to test the VGG16 behavior [55], because it has approximately 138 million parameters, and we can even venture to test

with a vision transformer (ViT) [56], which is an architecture different from the one used in this work, which attention mechanisms replace the convolutional layers.

- Although we have seen that the second network capacity for classifying is high, another option could be working on filtering incorporation to eliminate or attenuate those undesired image elements, such as hairs or marks, to delimit the lesion.

Bibliography

[1] CC BY-NC 4.0 deed attribution-NonCommercial 4.0 international creative commons. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, https://creativecommons.org/licenses/by-nc/4.0/.

[2] ISIC | international skin imaging collaboration. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, https://www.isic-archive.com.

[3] What is melanoma skin cancer? | what is melanoma? https://www.cancer.org/cancer/types/melanoma-skin-cancer/about/what-is-melanoma.html.

[4] Melanocytic nevus. https://en.wikipedia.org/w/index.php?title=Melanocytic_nevus&oldid=1180791786, Rights: Creative Commons Attribution-ShareAlike License, Page Version ID: 1180791786.

[5] What are basal and squamous cell skin canceers. https://www.cancer.org/cancer/types/basal-and-squamous-cell-skin-cancer/about/what-is-basal-and-squamous-cell.html.

[6] Rafael Carmena-Ramón, Almudena Mateu-Puchades, Sergio Santos-Alarcón, and Sofía Lucas-Truyols. Queratosis actínica: nuevo concepto y actualización terapéutica. *Atención Primaria*, 49(8):492–497, 2017. ISSN: 02126567, https://www.elsevier.es/es-revista-atencion-primaria-27-articulo-queratosis-actinica-nuevo-concepto-actua DOI: 10.1016/j.aprim.2017.01.004.

[7] Queratosis seborreica-queratosis seborreica - síntomas y causas. Web site: Mayo Clinic, https://www.mayoclinic.org/es/diseases-conditions/seborrheic-keratosis/symptoms-causes/syc-20353878,.

[8] L Hueso, O Sanmartín, A Alfaro-Rubio, C Serra-Guillén, A Martorell, B Llombart, C Requena, E Nagore, R Botella-Estrada, and C Guillén. Dermatofibroma gigante: descripción de un caso y revisión de la literatura. *Actas Dermo-Sifiliográficas*, 98(2):121–124, 2007. ISSN: 00017310.

[9] Vascular lesions of the skin - dermatologic disorders. https://www.msdmanuals.com/professional/dermatologic-disorders/benign-skin-tumors,-growths,-and-vascular-lesions/vascular-lesions-of-the-skin, MSD Manual Professional Edition.

[10] luciaclemares. ¿qué beneficios tiene la inteligencia artificial en la medicina? https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, https://www.telefonica.com/es/sala-comunicacion/blog/que-beneficios-tiene-la-inteligencia-artificial-en-la-medicina/.

[11] Aplicaciones reales de la inteligencia artificial en la medicina. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, https://www.apd.es/aplicaciones-inteligencia-artificial-en-medicina/.

[12] R Beuscart, M Wartski, MH Bourguignon, and C Foucher. An expert system for automatic analysis of whole body bone scintigraphy. *MEDECINE NUCLEAIRE*, 21(2):56–62, 1997. ISSN: 0928-1258.

[13] Samuel W. K. Chan, K. S. Leung, and W. S. Felix Wong. An expert system for the detection of cervical cancer cells using knowledge-based image analyzer. *Artificial Intelligence in Medicine*, 8(1):67–90, 1996. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55,https://www.sciencedirect.com/science/article/pii/0933365795000216, DOI: https://DOI.org/10.1016/0933-3657(95)00021-6.

[14] DM Wells and J Niederer. A medical expert system approach using artificial neural networks for standardized treatment planning. *INTERNATIONAL JOURNAL OF RADIATION ONCOLOGY BIOLOGY PHYSICS*, 41(1):173–182, 1998. ISSN: 0360-3016, DOI: 10.1016/S0360-3016(98)00035-2.

[15] S Dreiseitl, L Ohno-Machado, H Kittler, S Vinterbo, H Billhardt, and M Binder. A comparison of machine learning methods for the diagnosis of pigmented skin lesions. *JOURNAL OF BIOMEDICAL INFORMATICS*, 34(1):28–36, 2001. ISSN: 1532-0464, DOI: 10.1006/jbin.2001.1004,.

[16] S Dreiseitl, H Kittler, H Ganster, and M Binder. Classifying pigmented skin lesions with machine learning methods. In H Malmgren, M Borga, and L Niklasson, editors, *ARTIFICIAL NEURAL NETWORKS IN MEDICINE AND BIOLOGY*, PERSPECTIVES IN NEURAL COMPUTING, pages 174–179. Artificial Neural Networks Med & Biol; Fdn Knowledge & Competence Dev, 2000. ISSN: 1431-6854.

[17] Kva K. Lee. Machine learning framework for classification in medicine and biology. In WJ VanHoeve and JN Hooker, editors, *INTEGRATION OF AI AND OR TECHNIQUES IN CONSTRAINT PROGRAMMING FOR COMBINATORIAL OPTIMIZATION PROBLEMS, PROCEEDINGS*, volume 5547 of *Lecture Notes in Computer Science*, pages 1–7, 2009. ISSN: 0302-9743.

[18] Rahil Garnavi, Mohammad Aldeen, and James Bailey. Computer-aided diagnosis of melanoma using border- and wavelet-based texture analysis. *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, 16(6):1239–1252, 2012. ISSN: 1089-7771, DOI: 10.1109/TITB.2012.2212282.

[19] Rahil Garnavi, Mohammad Aldeen, and James Bailey. Computer-aided diagnosis of melanoma using border- and wavelet-based texture analysis. *IEEE Transactions on Information Technology in Biomedicine*, 16(6):1239–1252, Nov 2012. DOI: 10.1109/TITB.2012.2212282, ISSN: 1558-0032.

[20] M. Emre Celebi and Azaria Zornberg. Automated quantification of clinically significant colors in dermoscopy images and its application to skin lesion classification. *IEEE Systems Journal*, 8(3):980–984, 2014.

[21] Omar Abuzaghleh, Buket D. Barkana, and Miad Faezipour. Noninvasive real-time automated skin lesion analysis system for melanoma early detection and prevention. *IEEE JOURNAL OF TRANSLATIONAL ENGINEERING IN HEALTH AND MEDICINE*, 3, 2015. ISSN: 2168-2372, DOI: 10.1109/JTEHM.2015.2419612.

[22] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning, 2012. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, http://arxiv.org/abs/1112.6209.

[23] Farhad Mortezapour Shiri, Thinagaran Perumal, Norwati Mustapha, and Raihani Mohamed. A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU, 2023. 10.48550/arXiv.2305.17473.

[24] Leiyu Chen, Shaobo Li, Qiang Bai, Jing Yang, Sanlong Jiang, and Yanming Miao. Review of image classification algorithms based on convolutional neural networks. *Remote sensing*, 13(22), 2021. DOI: 10.3390/rs13224712.

[25] What is deep learning? | IBM. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, https://www.ibm.com/topics/deep-learning.

[26] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures, 2023. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, http://arxiv.org/abs/1807.10108.

[27] Lequan Yu, Hao Chen, Qi Dou, Jing Qin, and Pheng-Ann Heng. Automated melanoma recognition in dermoscopy images via very deep residual networks. *IEEE Transactions on Medical Imaging*, 36(4):994–1004, 2017. DOI: 10.1109/TMI.2016.2642839.

[28] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks, 2016. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55, http://arxiv.org/abs/1606.00373.

[29] Wanshun Wong. What is residual connection?, 2021. https://towardsdatascience.com/what-is-residual-connection-efb07cab0d55.

[30] Rajamanickam Thamizhamuthu and Subramanian Pitchiah Maniraj. Deep learning-based dermoscopic image classification system for robust skin lesion analysis. *TRAITEMENT DU SIGNAL*, 40(3):1145–1152, 2023. ISSN: 0765-0019, DOI: 10.18280/ts.400330.

[31] Lidia Talavera-Martinez, Pedro Bibiloni, and Manuel Gonzalez-Hidalgo. Hair segmentation and removal in dermoscopic images using deep learning. *IEEE ACCESS*, 9:2694–2704, 2021. ISSN: 2169-3536, DOI: 10.1109/ACCESS.2020.3047258.

[32] Dalal Bardou, Hamida Bouaziz, Laishui Lv, and Ting Zhang. Hair removal in dermoscopy images using variational autoencoders. *SKIN RESEARCH AND TECHNOLOGY*, 28(3):445–454, 2022. ISSN: 0909-752X, DOI: 10.1111/srt.13145.

[33] Ranpreet Kaur, Hamid GholamHosseini, and Roopak Sinha. Hairlines removal and low contrast enhancement of melanoma skin images using convolutional neural network with aggregation of contextual information. *BIOMEDICAL SIGNAL PROCESSING AND CONTROL*, 76, 2022. ISSN: 1746-8094, DOI: 10.1016/j.bspc.2022.103653.

[34] Douglas de A. Rodrigues, Roberto F. Ivo, Suresh Chandra Satapathy, Shuihua Wang, Jude Hemanth, and Pedro P. Reboucas Filho. A new approach for classification skin lesion based on transfer learning, deep learning, and IoT system. *PATTERN RECOGNITION LETTERS*, 136:8–15, 2020. ISSN: 0167-8655, DOI: 10.1016/j.patrec.2020.05.019.

[35] Conor Wall, Fraser Young, Li Zhang, Emma-Jane Phillips, Richard Jiang, and Yonghong Yu. Deep learning based melanoma diagnosis using dermoscopic images. In Z Li, C Yuan,

J Lu, and EE Kerre, editors, *DEVELOPMENTS OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES IN COMPUTATION AND ROBOTICS*, volume 12 of *World Scientific Proceedings Series on Computer Engineering and Information Science*, pages 907–914. Fern Univ; TH Koln Univ Appl Sci; Univ Technol Sydney; SW Jiaotong Univ; Shunde Polytechn; Minnan Normal Univ; Natl Assoc Non Class Log & Computat China, 2020. ISBN: 978-981-122-333-4.

[36] Wiem Abbes and Dorra Sellami. Deep neural networks for melanoma detection from optical standard images using transfer learning. In J Watrobski, W Salabun, C Toro, C Zanni-Merk, RJ Howlett, and LC Jain, editors, *KNOWLEDGE-BASED AND INTELLIGENT INFORMATION & ENGINEERING SYSTEMS KSE 2021)*, volume 192 of *Procedia Computer Science*, pages 1304–1312. KES Int, 2021. ISSN: 1877-0509.

[37] S. Georgakopoulos, V, K. Kottari, K. Delibasis, V. P. Plagianakos, and I Maglogiannis. Detection of malignant melanomas in dermoscopic images using convolutional neural network with transfer learning. In G Boracchi, L Iliadis, C Jayne, and A Likas, editors, *ENGINEERING APPLICATIONS OF NEURAL NETWORKS, EANN 2017*, volume 744 of *Communications in Computer and Information Science*, pages 404–414, 2017. ISSN: 1865-0929.

[38] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1), 2019. DOI: 10.1186/s40537-019-0197-0.

[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. ISSN: 1076-9757, https://www.jair.org/index.php/jair/article/view/10302, DOI: 10.1613/jair.953.

[40] J Moreno, Daniel Rodriguez, M. Sicilia, José Riquelme, and Y Ruiz. SMOTE-i: mejora del algoritmo SMOTE para balanceo de clases minoritarias. *Jornadas de Ingeniería del Software y Bases de Datos, Vol. 3, No. 1, 2009*, 2009. ISSN: 1988–3455, https://www.researchgate.net/publication/229045207_SMOTE-I_mejora_del_algoritmo_SMOTE_para_balanceo_de_clases_minoritarias .

[41] María José Basgall, Waldo Hasperué, Marcelo Naiouf, Alberto Fernández, and Francisco Herrera. SMOTE-BD: An exact and scalable oversampling method for imbalanced classification in big data. *Journal of Computer Science and Technology*, 18(3):e23, 2018. ISSN: 1666-6038, 1666-6046, https://journal.info.unlp.edu.ar/JCST/article/view/1122, DOI: 10.24215/16666038.18.e23.

[42] Tschandl P., Rosendahl C., and Kittler H. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, 2018. https://challenge.isic-archive.com/data/#2019, Sci. Data 5, 180161 DOI.10.1038/sdata.2018.161 (2018).

[43] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic), 2017. arXiv:1710.05006, https://challenge.isic-archive.com/data/#2019.

[44] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Allan C. Halpern, Susana Puig, and Josep Malvehy. Bcn20000: Dermoscopic lesions in the wild, 2019. arXiv:1908.02288, https://challenge.isic-archive.com/data/#2019.

[45] Fatimazahra Belharar. Enhancing classification accuracy for imbalanced image data using SMOTE, 2023. https://medium.com/@fatimazahra.belharar/enhancing-classification-accuracy-for-imbalanced-image-data-using-smote-41737783a

[46] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. http://jmlr.org/papers/v18/16-365.html.

[47] Cory Maklin. Synthetic minority over-sampling TEchnique (SMOTE), 2022. https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c.

[48] Rubiales Alberto. Funciones de error con entropia: Cross entropy y binary cross entropy, 2021. https://rubialesalberto.medium.com/funciones-de-error-con-entropia-cross-entropy-y-binary-cross-entropy-8df8442cdf35

[49] Vitaly Bushaev. Stochastic gradient descent with momentum, 2017. Medium, https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d.

[50] Luis Velasco. Optimizadores en redes neuronales profundas: un enfoque práctico, 2020. Medium, https://velascoluis.medium.com/

optimizadores-en-redes-neuronales-profundas-un-enfoque-pr%C3%A1ctico-819b39a3eb5.

[51] Mingxing Tan. EfficientNet: Rethinking model scaling for convolutional neural networks. version: 5, http://arxiv.org/abs/1905.11946.

[52] Vardan Agarwal. Complete architectural details of all efficientnet models, 2020. Medium, https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. http://arxiv.org/abs/1512.03385.

[54] resnet101 — torchvision main documentation, 2024. https://pytorch.org/vision/main/models/generated/torchvision.models.resnet101.html.

[55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. http://arxiv.org/abs/1409.1556.

[56] Papers with code - vision transformer explained, 2024. https://paperswithcode.com/method/vision-transformer.