TÉCNICO LISBOA

Automatic Prediction of BRAF Mutation in Melanoma Using Deep Learning

Simão Campos Gonçalves

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Dra. Ana Catarina Fidalgo Barata
Prof. Jorge dos Santos Salvador Marques

Examination Committee

Chairperson: Prof. João Manuel de Freitas Xavier
Supervisor: Dra. Ana Catarina Fidalgo Barata
Member of the Committee: Prof. João Miguel Raposo Sanches

November 2022

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First, I would like to thank my parents and siblings for their friendship, care, and support over all these years. They have always been there for me in the good and bad moments. I would also like to acknowledge my grandparents, aunt, and uncle for their active presence in my life.

To my supervisors, Dra. Ana Catarina Fidalgo Barata and Professor Jorge dos Santos Salvador Marques, whose guidance and trust was fundamental. Thank you very much.

A special thanks to Dra. Ana Catarina Fidalgo Barata for the dedication and support, which were vital throughout the dissertation. I would like to give also special thanks to Rita Verdelho and António Gama for their friendship and support during these last two semesters.

To my friends from Odemira, Lisboa, Técnico, Erasmus, and all the other important people I met along the way, thank you. I am grateful to have you all in my life.

Lastly, I would like to thank Instituto Superior Técnico for making my engineering education possible.

To all of you, from the bottom of my heart, thank you.

Abstract

Melanoma is the deadliest form of skin cancer. The treatment of metastatic melanoma patients depends on the mutational state of the BRAF gene. Thus, it is crucial to timely assess this gene's status to select an adequate treatment. Nowadays, to infer the BRAF status, a biopsy is performed on the lesion area, and after that, PCR analysis is executed on the extracted DNA. This process is efficient; however, it is slow and depends on the experience of specialized personnel. It is essential to complement the existing diagnostic techniques. Previous works have shown that dermoscopic images of melanomas convey relevant information about the BRAF mutational status. The objective of this thesis is to explore faster, more automated, and less human-dependent ways of predicting BRAF status for melanoma patients. Regarding the BRAF data, only a few labeled *ex vivo* dermoscopic images are available for this work. Therefore, three convolutional neural networks are pre-trained on a related task (benign/melanoma classification task) using a larger *in vivo* dataset. Some versions of the related task pre-training use *ex vivo* data to perform domain adaptation, attempting to mitigate the shift between the *in vivo* and *ex vivo* data. The pre-trained architectures are used to extract features from the BRAF dataset, and classification algorithms are employed to predict the mutational status, most inspired by few-shot learning. The results obtained in this work overcome the current state-of-the-art results, proving that the proposed deep learning approaches are a promising venue of research for BRAF status prediction.

Keywords

Deep Learning, Few-Shot Learning, Domain Adaptation, Melanoma, BRAF, Dermoscopy

Resumo

Melanoma é o tipo de cancro de pele mais fatal. O tratamento dos pacientes com melanoma metastizado depende do estado mutacional do gene BRAF. Portanto, é crucial aferir o estado deste gene de modo a selecionar um tratamento adequado. Atualmente, para avaliar o estado da BRAF, é realizada uma biópsia e em seguida, um PCR ao ADN extraído. Este processo é eficiente. No entanto, é lento e depende da experiência de pessoal especializado. Consequentemente, é essencial complementar as técnicas de diagnóstico existentes. Trabalhos anteriores mostram que imagens dermatoscópicas de melanomas contêm informações relevantes sobre o estado mutacional da BRAF. O objetivo desta tese é explorar formas mais rápidas, mais automatizadas e menos dependentes de humanos de prever o estado da BRAF no melanoma. Relativamente aos dados da BRAF, apenas algumas imagens dermatoscópicas *ex vivo* anotadas estão disponíveis neste trabalho. Portanto, três redes neurais convolucionais são pré-treinadas numa tarefa relacionada (tarefa de classificação benigno/melanoma) usando um conjunto de dados *in vivo* maior. Algumas versões do pré-treino na tarefa relacionada usam dados *ex vivo* para realizar adaptação de domínio, tentando atenuar o "shift" entre os dados *in vivo* e *ex vivo*. As arquiteturas pré-treinadas são usadas para extrair informação do dataset da BRAF. Recorrem-se a algoritmos de classificação para prever o estado do gene, maioritariamente inspirados em aprendizagem com poucos dados. Os resultados obtidos neste trabalho superam os resultados do estado da arte, comprovando que as abordagens de aprendizagem profunda propostas são um caminho promissor para a previsão do estado da BRAF.

Palavras Chave

Aprendizagem Profunda, Aprendizagem com poucos dados, Adaptação de Domínio, Melanoma, BRAF, Dermatoscopia

Contents

List of Figures

List of Tables

Acronyms

AUC	Area Under the Receiver Operating Characteristic
BACC	Balanced Accuracy
BWV	Blue-Whitish Veil
BRAF	V-Raf murine sarcoma viral oncogene homolog B
BRAF+	Mutated BRAF
BRAF-	Non-Mutated BRAF
CNN	Convolutional Neural Network
CORAL	Correlation Alignment
DA	Domain Adaptation
DANN	Domain-Adversarial Neural Network
DL	Deep Learning
FN	False Negative
FP	False Positive
FSL	Few-Shot Learning
GRL	Gradient Reversal Layer
HER2	Human Epidermal Growth Factor Receptor 2
HER2+	Mutated HER2
HER2-	Non-Mutated HER2
HE	Hematoxylin and Eosin
ISIC	International Skin Imaging Collaboration
LR	Logistic Regression
LRR	Low-Rank Representation
MIA	Melanoma Institute Australia

MIABID	Medical Image Assisted Biomarkers' Discovery
MICCAI	Medical Image Computing and Computer Assisted Intervention
ML	Machine Learning
MLP	Multilayer Perceptron
MSKCC	Memorial Sloan Kettering Cancer Center
KNN	K-Nearest Neighbors
PR	Precision
SE	Sensivity
SF	Summarized Features
SLN	Sentinel Lymph Node
SLN+	Mutated SLN
SLN-	Non-Mutated SLN
SP	Specificity
ROI	Region of Interest
RT	Related Task
TN	True Negative
TP	True Positive
WSI	Whole Slide Image

1

Introduction

Contents

1.1 Motivation

A significant part of deaths related to skin cancer is associated with melanoma. Usually, to treat a melanoma patient, surgical excision is proposed. However, when the tumor is metastatic, other treatment options must be explored. Chemotherapy, immunotherapy, and targeted therapies are possible options for these cases [1]. Target therapies are one type of treatment that has emerged in the last years, as well as immunotherapy. These treatments allow acting in specific genetic mutations within the cells, promoting personalized therapies. Nowadays, the type of treatment indicated to a patient who suffers from metastatic melanoma varies according to V-Raf murine sarcoma viral oncogene homolog B (BRAF) status in the cancerous cells [2]. Figure 1.1 summarizes the approved treatment protocol for metastatic melanoma.

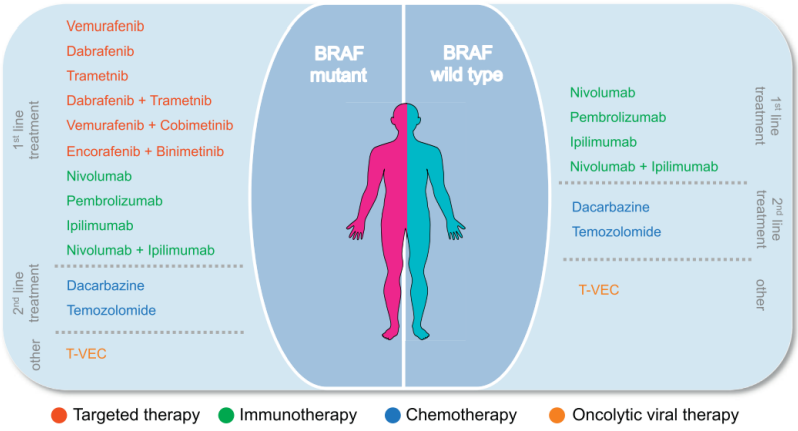


Figure 1.1: Metastatic melanoma - Treatment protocol [3].

As the figure shows, the BRAF status influences the first-line treatment, i.e., the recommended treatment for a diseased person. When a patient presents Mutated BRAF (BRAF+), the recommended first-line treatment consists of target drugs to act in BRAF (monotherapy) or a combination of drugs that act in both the BRAF and MEK genes. MEK is a gene that cooperates with BRAF, and some studies suggest that when targeted together with the BRAF gene results in higher rates of tumor responses [4]. A better survival outcome is observed when drugs targeting both genes are combined. When a patient presents a wild type BRAF (Non-Mutated BRAF (BRAF-)), the first-line treatment consists only of immunotherapy.

Throughout the years, it has been noticed that the treatment protocol is too general; therefore some patients develop immunity against the drugs used in the assigned treatment. Besides, there are also patients who, despite not becoming immune to the treatment, manifest adverse side effects. Problems arising from the treatment protocol not being personalized enough are also relevant. However, they are not the focus of the present work since protocol improvement is not intended. This thesis focuses on

detecting the mutational status of the BRAF gene in an automated way so that each patient can be guided to adequate treatment, given the current protocol.

1.2 Problem Definition

1.2.1 Current Approach to Predict BRAF Status

BRAF is a gene that regulates three essential aspects of a cell: proliferation, survival, and differentiation. This human gene appears mutated in around half of the cases related to melanoma. Also, a change in this gene increases the tumorous cells' propensity to expand [2]. The four most common melanoma subtypes are superficial spreading melanoma, nodular melanoma, lentigo melanoma, and acral lentiginous melanoma. BRAF+ is mostly incident in the superficial spreading and nodular subtypes. However, according to [5], these mutations are not enough to cause melanoma, as they have been detected in both benign and dysplastic nevi.

BRAF status is relevant to the treatment of metastatic melanoma patients because the protocol varies in conformity with the presence/absence of mutation in this gene. Currently, to evaluate the mutational status of BRAF, the most common approach is followed by an excision biopsy. DNA extracted from cells of the skin lesion is evaluated through PCR analysis which is an efficient but slow procedure and requires the work of specialized personnel. The analysis depends on the pathologist's experience, as the pathologist must select the most relevant part of the lesion to be submitted to the PCR analysis. Different parts of the lesion might lead to different outcomes. Therefore, more than one PCR might be needed to confirm the prediction.

The search for alternative ways to check on the mutational status of the BRAF gene is justified because it is crucial to find faster, more automatized, and less human-dependent ways of inferring this gene's status.

1.2.2 Dermoscopy: A Viable Solution?

Some medical diagnostic techniques allow the examination of skin lesions in a non-invasive way. One example is dermoscopy, a technique that uses skin surface microscopy.

Figure 1.2(a) exemplifies a dermoscopy examination, and figure 1.2(b) the type of image obtained from this analysis. The dermoscopic image was extracted from an open-source dataset available at the International Skin Imaging Collaboration (ISIC).

Dermoscopy allows obtaining high-resolution images of the whole lesion area; also, it has been noticed that BRAF+ melanomas are associated with particular histopathologic and clinical features. Since dermoscopy acts as a bridge between these two types of features, dermoscopic features are

expected to be associated with the mutations.

Following [6], in most cases, genetic mutations induce morphological changes in the cells and the tumorous tissue. For instance, streaks and Blue-Whitish Veils (BWVs) in dermoscopy appear associated with BRAF+ melanomas [7]. Thus, dermoscopic images emerge as promising data for BRAF mutation prognosis as they may convey relevant information for this task.

(a) **(b)**

Figure 1.2: Skin screening performed by a dermatologist [8] (a); Dermoscopic image of a benign skin lesion located in the torso region [9] (b).

1.2.3 Proposal: Approach Based on Deep Learning

This thesis aims to automatize the inspection of dermoscopic images and to reduce the subjectivity associated with human analysis. To this end, the use of Deep Learning (DL) methods to extract relevant information (features) for BRAF status prediction from dermoscopic images is proposed. These features can then be processed by other Machine Learning (ML) algorithms to predict the presence/absence of mutation. DL methods require a high volume of data to achieve reliability. Unfortunately, the only available dermoscopic dataset with BRAF information is small. However, Few-Shot Learning (FSL) strategies intend to overcome this problem, making it a topic worth of exploring. For this thesis, FSL is related to classification tasks; therefore, in the present work, FSL is studied just on the scope of few-shot classification tasks. Besides, for this research, *in vivo* as well as *ex vivo* dermoscopic images are present so it will be interesting to explore if a model can perform well in the different domains. Some data processing techniques address issues involving the different nature of datasets. One well-known strategy to address this matter is Domain Adaptation (DA).

1.3 Objectives and Contributions

The main objective of this work is to both accelerate and automatize BRAF status prediction using DL based approaches, reducing the dependence on human expertise. Fulfilling this objective will complement the existing ways to assess BRAF status and will speed up the process of selecting the most appropriate treatment for each patient which, consequently, increases the survival expectation. Other

scientific objectives are stated, namely to study if techniques based on FSL will help overcoming the lack of data and if DA methods will be helpful in a context were more than one domain is involved.

The investigation conducted in this thesis contributed in part for the research paper "Predictive Biomarkers in Melanoma Detection of BRAF Mutation using Dermoscopy" [10] which was present in the Medical Image Computing and Computer Assisted Intervention (MICCAI) 2022 workshop on Medical Image Assisted Biomarkers' Discovery (MIABID).

1.4 Document Organization

This document is structured as follows: Chapter 1 explains the motivation for predicting BRAF status and gives an insight into the current procedures to detect the mutations and what is intended with the investigation conducted in this work. Chapter 2 presents a literature review of previous works related to this thesis's topic. Chapter 3 supplements the necessary theoretical background to fully understand the methods used. Chapter 4 explains the proposed approaches to address BRAF status prediction. Chapter 5 informs the reader about the available resources to conduct the investigation and the set-up for the conducted experiments. Besides, it also informs the reader about the evaluation metrics considered and the implementation challenges faced in the present work. Chapter 6 contains the results and the conclusions drawn from this investigation. Finally, chapter 7 gives the final remarks and opens the doors to further investigation.

2

State of the Art Review

Contents

This chapter starts by presenting some literature review about the prediction of cancer patient prognosis. Articles concerning Sentinel Lymph Node (SLN) and genetic mutations (Human Epidermal Growth Factor Receptor 2 (HER2) and BRAF) are analyzed. Then, a section summing up the flaws and the advantages of the methods used in the reviewed literature is introduced to tackle BRAF status prediction in melanomas.

2.1 State of the Art

2.1.1 Automatic Prediction of SLN Mutation

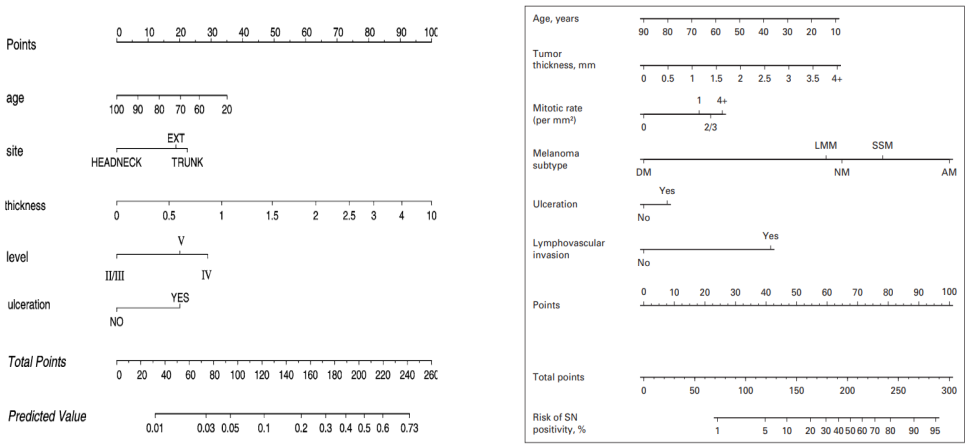
The high fatality ratios associated with melanoma have raised an urge to identify prognostic factors that can increase melanoma patients' life expectancy. One of the most important prognostic factors is the local lymph nodes status, specifically the SLN, which is closest to the lesion area.

Research conducted on the impact of Mutated SLN (SLN+) on the life expectancy of the victims has shown that these patients have fewer survival chances than patients with Non-Mutated SLN (SLN-) [11]. To increase the life expectancy of the diseased, lymphadenectomy is recommended for anyone presenting SLN+.

The most common procedure for analyzing the SLN status is a biopsy. Nevertheless, some patients present severe conditions, and their integrity could be compromised if submitted to a biopsy. Gradually, investigation with the end of predicting the status of the SLN in a less invasive way has been arising.

The gross part of the studies tries to build models to predict the positivity of the SLN. This prediction is made by gathering the maximum relevant clinical information from the patient (like age and gender) and metadata for the melanoma itself, for instance, site, thickness (Breslow thickness), histologic subtype, mitotic rate, Clark level, and ulceration. Two examples of this are the researches conducted in [12] and [13], which resort to simple linear models called nomograms to predict the positivity of the SLN. Nomograms establish an easy and graphical way to approximate equation solutions. They have become an important tool in oncology as they use widely available clinicopathologic information from a patient for tasks like outcome prediction and decision making.

The objective of Lo *et al.* [12] was to develop a nomogram (figure 2.1(b)) that could overcome a previous nomogram introduced in 2005 (figure 2.1(a)). According to [12], patient selection for biopsy is based on national and international established guidelines. The suggested nomogram requires six clinicopathological parameters: patient age, tumor thickness, mitotic rate, melanoma subtype, lymphovascular invasion, and ulceration — one more covariable than the nomogram proposed in [13].



(a) **(b)**

Figure 2.1: Nomograms to predict the SLN status: Memorial Sloan Kettering Cancer Center (MSKCC) nomogram [13] (a); Melanoma Institute Australia (MIA) nomogram [12] (b).

Using labeled data for SLN status from two different institutes, it was possible to train a model that took into account all the available metadata for the patient and the melanoma (patient age, Clark level, tumor site, tumor thickness, tumor histological subtype, mitotic rate, ulceration, and lymphovascular invasion). A multivariable logistic regression was estimated using the purposeful variables selection method [14], which only selects the most relevant variables for a model. In the end, the generated model only considered six covariables dropping Clark level and tumor site. The resulting model presented an Area Under the Receiver Operating Characteristic (AUC) of 73.9%, better than the 67.7% attained by the model of [13] for the same dataset. Another extraordinary result is that this new nomogram for a Sensivity (SE) of 95% would exclude 22.1% of the SLN- population from biopsy, a better result than the guidelines and the predecessor nomogram. This work has shown that some metadata might lose relevance when other clinical data is added to a model. Besides, it has also shown that there are other clinicopathological parameters that, when put together, can generate a more robust model for SLN+ prediction.

The use of these nomograms is very intuitive. A vertical line from each parameter is drawn until it reaches the "Points" mark. After this is done for each parameter, the total amount of points is calculated. A new line is drawn, this time from the "Points" mark until the "Predicted Value" mark in the case of the MSKCC nomogram or until the "Risk of SN positivity, %" mark in the case of the nomogram proposed by MIA.

The interest in finding biomarkers that justify the presence of a mutation has grown lately, and ML techniques respond to this interest, acting as a useful tool. Some of the models that address the SLN status are obtained based on DL techniques. These models resort to a Convolutional Neural Network

(CNN) to extract relevant features from visual data, such as stained histologic sections images, to learn digital biomarkers that might indicate the presence of lymph node mutation.

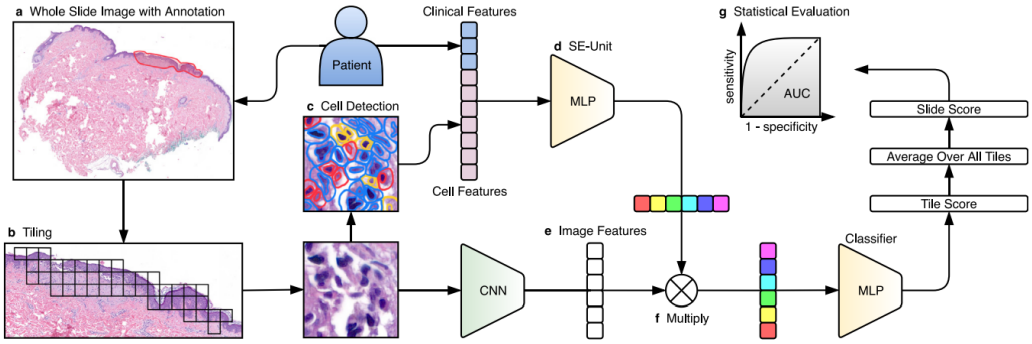


Figure 2.2: Pipeline for SLN status prediction [15].

Brinker *et al.* [15] proposed to use a DL approach to address this question (figure 2.2 presents the pipeline developed in [15]). In their work, they tried various feature combinations (clinical features, cell features, and image features), each of these obtained in different manners. The idea was to use a pipeline to extract melanoma features from prior melanoma tiles. The used tiles were obtained by tiling Hematoxylin and Eosin (HE) stained Whole Slide Images (WSIs) that were annotated by a bioinformatician; therefore, Regions of Interest (ROIs) for the tumor were marked a priori. Each one of the used tiles maintained the label of the corresponding WSI.

The tiles were fed to a CNN that followed a ResNet architecture (ResNeXt50 [16]) which was pretrained on the ImageNet-1k dataset [17], and the clinical and cell features were processed in parallel, entering a Multilayer Perceptron (MLP) that worked as a squeeze and excitation unit [18]. The three types of features (clinical features, cell features, and image features) could then be combined and proceed to another MLP that classified a given tile as SLN+ or SLN-. The prediction for a WSI was made by averaging the prediction of all its tiles.

Despite the efforts, the study shown by these authors only achieved an AUC of $61.8\% \pm 0.2\%$ and it was when only the image features were being considered; besides, this AUC value had an associated SE of just $48.2\% \pm 14.2\%$. These results were not relevant enough to support the transference of the model to clinical practice.

2.1.2 Automatic Prediction of Genetic Mutations

The techniques referred to in subsection 2.1.1 can be extended to other types of prognostic indicators; thus, it is important to understand how these techniques work and how they can be adapted to address other problems.

As stated by [15], SLN+ appears to be associated with BRAF+; therefore, lymph node status might

as well give an upper hand when studying specific gene mutations since both problems seem to be related.

Regarding genetic mutation prediction models, two factors are highly relevant. The first is if the proposed model can accurately predict a specific gene mutation. The second is if, given the genetic mutation, the model can clarify whether a specific treatment targeting the mutated gene would benefit the patient.

Farahmand *et al.* [19] proposed a model to classify the HER2 in breast cancer as Mutated HER2 (HER2+) or Non-Mutated HER2 (HER2-). Besides, the authors presented a model that can also predict the response of a patient with HER2+ to the trastuzumab treatment. The techniques described in this article have some similarities to the ones used for SLN status prediction in [15], giving evidence that the same DL approaches can be widely used for different pathologies. In [19], two strategies were compared, a full learning strategy and a transfer learning strategy. Full learning strategies imply that the training process is entirely based on the training dataset of the given problem, the so called "training from scratch". In contrast, transfer learning is a method where a model can be trained on a different but related domain or task than the one in which it will be evaluated [20]. WSIs with annotated ROIs were used, like in [15], but in this case, for the HER2-/HER2+ tumor relevant areas. A CNN (Inception-V3 architecture [21]) was trained just on tiles from the WSIs in the full training approach, and in the transfer learning approach, the network was first trained on the ImageNet dataset and then fine-tuned for the target dataset. Three different training schemes were used. The first not taking into account the ROIs, having all the tiles inherited the labels from the corresponding WSI. The second training scheme considered the annotated ROIs, and only these parts of the WSIs were tiled and labeled according to the original WSI. The last training scheme considered the annotated region but also some areas outside the ROI for tiling. In this case, there were three different labels, "HER2-" and "HER2+" (inherited from the corresponding WSI) and "other" for the tiles out of the ROIs.

The model was evaluated on an independent dataset from a different data center. The evaluation was performed on tile level and slide level. The slide level evaluation was obtained by averaging the results of all the tiles belonging to the same WSI and comparing it to a well-defined threshold value, similar to what is done in [15].

The obtained results show that both the full learning and transfer learning approaches achieve similar results. The model which results from the last training method (using three different labels) surpasses the models which resulted from the other training methods, achieving a classifier that makes possible AUCs of 0.88 for "HER2+" and "HER2-" and 0.87 for "other". These AUCs were obtained on slide level for the independent dataset.

As for the treatment response, a similar methodology was used to build the classifier. The training of this classifier considered a dataset only with HER2+ samples from patients who were submitted to the

trastuzumab treatment afterward. The classification was split into “responders” and “non-responders”, obtaining an AUC of 0.80. The authors also tried to use the previous HER2-/HER2+ classifier to make predictions on treatment response, but the achieved results were poor.

The work in [19] shows that ROIs are useful when considering a CNN based model since they help highlight relevant features for the mutated genes. Also, it shows that better ROI estimation by the CNN is achieved when considering a three-way classifier. Lastly, it reveals that the use of two independent models for the tasks of genetic mutation identification and treatment response prediction is a better practice than using the same model.

The conducted image analysis on the previous work was performed on HE slides, although there are also studies that analyze dermoscopic images to predict mutations. The work of Armengot-Carbó *et al.* [7] is one example.

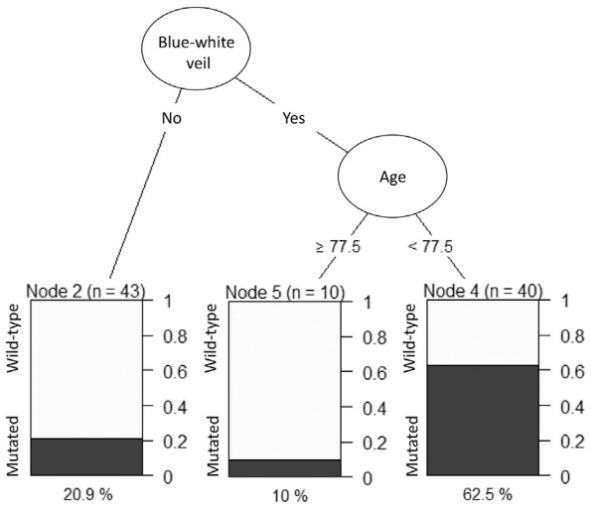


Figure 2.3: Decision tree for BRAF mutational status prediction [7].

In [7], the authors developed a classification tree to predict the BRAF mutational status in melanoma (figure 2.3). The classifier considered only two variables: patient age and the presence/absence of BWVs in the dermoscopic images. It attained an accuracy of 73.1%. In their work, the main objective was to relate BRAF status with dermoscopic features, which is still a poorly explored subject. In the conducted study, dermoscopic images from cutaneous melanomas were considered, as well as clinical and histopathological data. The used method consisted of two independent observers of 93 dermoscopic images, a dermoscopic feature from these images would only be considered valid when both observers agreed. Statistical tests were conducted on the studied variables, leading to the decision tree classifier. 47 dermoscopic images from the whole dataset were randomly selected to check on the model reliability through cross-validation. From this work, it was concluded that the dermoscopic features more frequent on BRAF+ melanomas were streaks, exophytic papillary structures, and BWVs. Thus, provid-

ing evidence that some dermoscopic features relate to BRAF mutational status. Furthermore, BRAF+ was more incident in younger patients, and the BWV structure was the one feature that seemed to relate more with the mutation as it was the one that kept bigger statistical significance after performing multivariate analysis.

2.2 Discussion on the Reviewed Literature

The review presented in 2.1 introduced a model that does not require any visual data to predict SLN status [12]. This article suggests that mutation detection can be approached using just clinicopathologic parameters from a patient.

The review also covered two DL approaches, one for SLN status [15] and another for HER2 status [19]. The work presented in the HER2 article achieved a reasonable AUC value of 0.88 for HER2-/HER2+ classification, while the work on SLN status only achieved, in the best scenario, an AUC of $61.8 \pm 0.2\%$ for SLN-/SLN+ classification. Both works used tiling of HE stained WSIs, and even though the work in [15] used a greater number of slides for training (415), the number of SLN+ and SLN- tiles was imbalanced (150 SLN+, 265 SLN-) while in [19] the training set consisted of 188 WSIs but with more balanced data (93 HER2+, 95 HER2-). When a model is being trained, the training data should be more balanced like in [19]; otherwise, the model will have a bigger tendency to classify unseen data in the class with more training examples which can lead to errors. Besides, in [15], the network only receives tiles from the annotated area, and according to [19], this might result in the network not learning features that associate with non-mutation.

Lastly, an article about BRAF status prediction in melanomas was introduced [7]. This article strongly supports the work which will be developed in this thesis because it proves that dermoscopic images may present structures that relate to BRAF status. In this article, with only the patients' age and information about the presence/absence of BWVs, it was possible to build a decision tree to predict BRAF mutational status with an accuracy of 73.1%. It is also mentioned that ulceration could be related to the BRAF status since it is a result of a study conducted by [22]. In [7], BWVs were a prevalent factor over the other dermoscopic features. In this work, the validity of dermoscopic features depended on the evaluation of two independent observers. This strategy could be substituted by a DL model, which would extract the relevant features from the dermoscopic images. Hence, the search for better models that use dermoscopic features and other relevant clinical data is legitimized.

In the reviewed literature, the DL approaches present in [15] and [19] rely on the use of HE images, while the work developed in this thesis will explore dermoscopic images, so the important aspects to be noted in an image will not be the actual cellular morphology but the lesion morphology at the macroscopic level. Features like ulceration can be noticed in both types of images. However, ulceration can

be more perceptible in dermoscopic images. This said, the image processing will not be exactly equal to the processing used in these works, mainly because the type of data is different. Instead of tiling the image, the mutation will be predicted from the whole dermoscopic image.

3

Theoretical Background

Contents

In this chapter, an overview about CNNs is supplemented. It is intended for the reader to understand the importance of these networks as information extractors. Additionally, a study on FSL techniques is exhibited as the available dermoscopic BRAF dataset is small. Lastly, DA methods are presented since, for this work, datasets from different domains will be used.

3.1 Convolutional Neural Networks

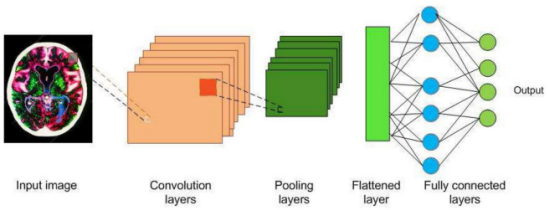


Figure 3.1: Simple CNN model [23].

CNNs are one type of DL architecture that achieved brilliant results on image-related tasks such as image segmentation and classification over the past years. This class of artificial neural networks started gaining popularity when the AlexNet architecture [24] surpassed all the other image classification models on the ImageNet challenge in 2012. Figure 3.1 illustrates a simple CNN architecture working as a feature extractor and classifier.

The use of these networks in the scope of medical research is becoming more common. There is some recent medical investigation that makes use of this type of network ([6], [15], [19]). Moreover, approaches that use this type of network are among the best in many medical image understanding challenges, and one example is the MICCAI biomedical challenge [23]. The excellent performance of CNNs for these types of tasks is due to their ability to recognize different patterns in images, which ultimately leads to learning relevant image features.

CNNs receive images as inputs. For a CNN, an image is seen as a numeric matrix that is usually three-dimensional (height, width, and depth). The height and the width define the 2D spatial dimension of the input, while the depth dimension has to do with the image's color. For instance, when the input is an RGB image, the depth dimension is equal to 3, where each of the three channels encodes the intensity of a specific color on the pixels of the input image.

During training, the networks learn how to extract relevant information (features) for the task through a succession of internal representations of the inputs. To output a label, there are specific building blocks for the CNN, which convert the extracted information to the probability of the input belonging to a particular class.

CNNs combine different building blocks, such as convolutional layers, pooling layers, and fully-

connected layers, which are its core components.

3.1.1 Convolutional Layers

Convolutional layers are the main component blocks of a CNN model and are responsible for the feature extraction procedure. A set of learnable kernels, also called filters, parameterize these layers. Kernels are three-dimensional ($H \times W \times D$), where "D" is the input volume's depth [23]. Despite having the same depth, kernels are generally spatially smaller than the input.

In these layers, the input data is convolved with each one of the kernels, producing 2D activation maps (also known as feature maps), one for each kernel. The convolution operation consists in sliding the filters along the input, performing dot products on every spatial location [25]. To perform the convolution operation, one must define the stride (quantifies the kernel movement) and the padding (a technique used to ensure that the center of each kernel overlaps with the outermost element of the input volume) [26].

Figure 3.2 was selected from [26] to illustrate a convolution operation; here, the input tensor of dimension 5×5 is padded with zeros (zero-padding), and a stride of 1 is used.

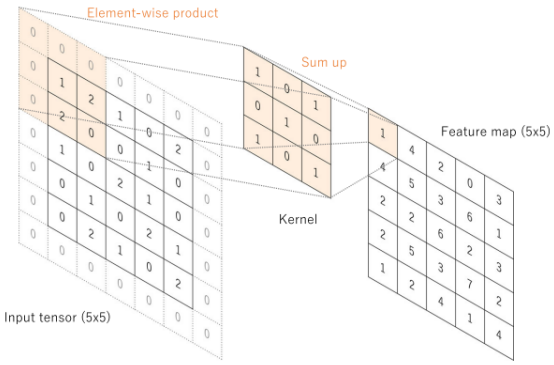


Figure 3.2: Convolution operation [26].

The output of a convolutional layer is obtained by stacking the resultant 2D activation maps along the third dimension. Each neuron of the activation image is the result of a dot product between a filter and the input at a specific spatial position, so this region is only connected to a small portion of the input, which is called the receptive field size of the neuron, which in turn has the same spatial dimensions as the kernel [27].

During the learning process, the network's filters are updated to become susceptible to certain patterns in the input volume, producing activation images when these patterns are present [25].

3.1.2 Pooling Layers

Downsampling the spatial dimension of the input is possible thanks to pooling layers. By doing so, the parameters a network must learn are reduced. Besides, reducing the dimensionality of the activation maps conveys translation and rotational invariance [23].

Max pooling and average pooling are the most basic pooling operations. In max pooling, patches of the activations are extracted; from these patches, only the maximum value is kept, and the rest is discarded [26]. For the average pooling, instead of selecting the maximum value for each patch, the average value is computed [28]. Figure 3.3 illustrates the max pooling and average pooling operations.

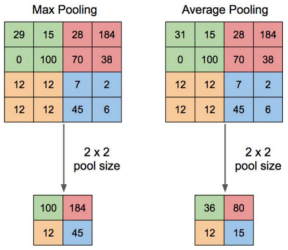


Figure 3.3: Max pooling and average pooling. Filter size: 2×2; Stride: 2 [28].

More extreme pooling operations such as global average pooling and global max pooling are used to reduce the feature maps' information to a spatial size of 1×1, keeping only the depth dimension ($1 \times 1 \times D$). The difference between global average pooling and global max pooling is that in the first operation, the information of a feature map is averaged [29] whereas, in the second operation, the largest value of the whole feature map area is kept [30]. The global average pooling operation is usually only performed once, right before the fully-connected layer [26]. The same goes for the global max pooling.

3.1.3 Fully-Connected Layers

Fully-connected layers possess neurons that directly connect to the units in the two adjacent layers [23]. In these building blocks, every input is connected to all the outputs by learnable weights. They receive 1D inputs; therefore, the information must be flattened before proceeding into the fully-connected layers (check figure 3.1).

In a CNN, for a classification task, after the feature extraction procedure, a fully-connected layer can map the feature vectors to the output. In this situation, the CNN extracts and classifies the information, making the whole process end-to-end. It is then typical that the last fully-connected layer presents the same number of units at the output as the number of classes present for the classification task [26].

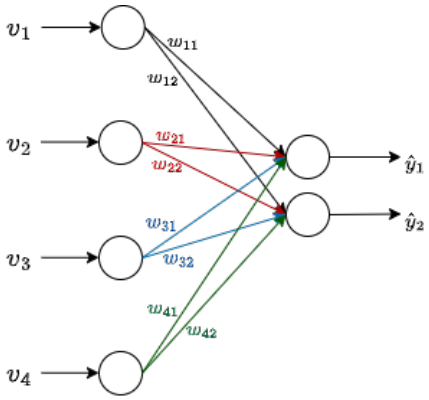


Figure 3.4: Fully-connected layer connecting a flattened information vector to the output.

Consider the fully-connected layer in figure 3.4. In this figure, the layer connects a one-dimensional input feature vector ($v \in \mathbb{R}^{4\times1}$) to the output. The fully-connected layer is parameterized by a weight matrix $W \in \mathbb{R}^{4\times2}$. The operation performed by the fully-connected layer is a weighted sum between the input vector and the layer's weights, followed by a non-linear activation function σ as explicit in (3.1).

$$\hat{y} = \sigma(W^\top v) \iff \begin{bmatrix}\hat{y}_1\\ \hat{y}_2\end{bmatrix} = \sigma\left(\begin{bmatrix}w_{11} & w_{21} & w_{31} & w_{41}\\ w_{12} & w_{22} & w_{32} & w_{42}\end{bmatrix}\begin{bmatrix}v_1\\ v_2\\ v_3\\ v_4\end{bmatrix}\right). \tag{3.1}$$

The result of the weighted sum is often called logits, i.e., a raw vector of non-normalized probabilities. To obtain normalized probabilities a non-linear activation function like the sigmoid or the softmax is applied.

3.2 Supervised Training of a CNN for a Classification Task

Training comprises two steps, the forward pass, and the backward pass. The forward pass consists of predicting the class for input images through a succession of internal representations. The internal representations are obtained by the set of operations performed by the network, given the present weights for the convolutional and fully-connected layers [26]. For multi-class classification problems, the last fully-connected layer is usually followed by a softmax activation function to convert the output logits values to normalized probabilities. Following a similar notation to the one in [31], the softmax function is mathematically expressed as

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{3.2}$$

where K is the number of classes, p_i is the probability of an input image belonging to class i, z_i is the unnormalized value that corresponds to probability p_i. The outcome of a softmax function is a vector $p = (p_1, ..., p_K)$ which contains the probability distribution of an input image on the K classes present in

the classification task.

Following the forward pass, the backward pass updates the network's parameters, i.e., the layer's weights. The update is based on the network's performance on the forward pass. A cost function, known as the loss function, is computed to assess the model's performance. A classification loss compares the predictions with the ground truth labels, penalizing misclassifications. One example of a classification loss function is the cross-entropy.

$$L_{cross-entropy} = -\sum_{i=1}^{K} q_i log(p_i). \tag{3.3}$$

In (3.3), p_i, and K have the same meaning as in (3.2), and q_i is the ground truth label for the input image (q is an one-hot encoded vector of dimension K where all coefficients are equal to zero except for the coefficient that corresponds to the correct class).

The model's parameters are optimized so that the loss is minimized. The optimization can be performed by algorithms such as gradient descent [26].

3.3 Few-Shot Learning

ML, in particular DL, has shown to be effective when a large amount of data is available for training. However, when the datasets are small, ML methods struggle to generalize. Modern learning approaches, such as FSL, try to overcome this problem. This learning mechanism uses prior knowledge to obtain good performance on a target task given only a few labeled data during the training stage.

As previously referred, in the context of this thesis, there will be very few labels regarding the BRAF status in the dermoscopic images of the melanomas so, supervised FSL can be useful to obtain an adequate classifier.

FSL differs from traditional supervised learning. In the traditional approach, a model is trained on a large dataset, and then on the testing stage, the model is presented with unseen examples and classifies those examples in one of the classes seen during the training stage. In FSL, the objective is different; instead of having as the main goal the classification of unseen data in known classes (base classes), the objective can be to learn a model capable of classifying data in novel classes, i.e., classes not present during the training stage, given only a few labeled examples. Another objective of FSL can also be to learn a model that can perform classification tasks similar to the ones of the training stage but with fewer data.

In this chapter, data-based FSL is presented [32]. After that, the concept of meta-learning is introduced. Following meta-learning, to close this FSL review, metric-based methods are discussed.

3.3.1 Data-Based FSL

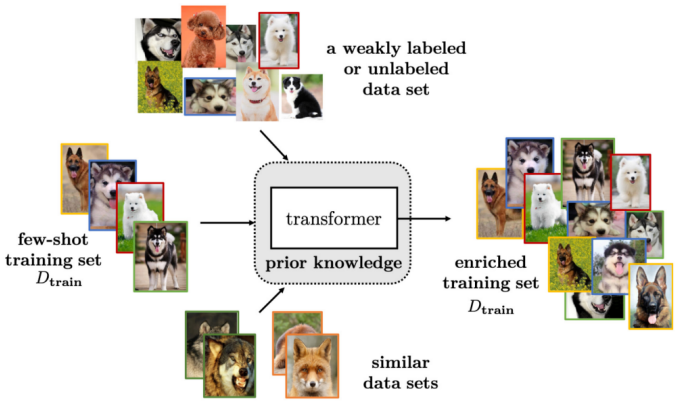


Figure 3.5: FSL problem solved by enlarging the few-shot training set [32].

The following methods are based on enriching the few-shot training set with new data, which will not be the central strategy in this thesis so, other methods will be explained in greater detail.

Data-based FSL (figure 3.5) works on a basic principle. These methods consist in enlarging the few-shot training set D_{train} with more data so that standard supervised learning techniques can be considered as a valid approach for the problem [32]. To achieve this, samples from the training set or even from another dataset can be transformed and used as new training examples. The use of hand-crafted transformations such as rotations, shearing, or translations is, in the context of FSL, associated with data pre-processing. In these cases, invariance is introduced manually. The use of hand-crafted transformations has some drawbacks, including the lack of adaptability of implemented transformations to other datasets and the incapability of humans to introduce all the possible invariance to a model [32]. Therefore, to augment the data, other techniques might be more appropriate.

To enrich the few-shot training set, other image transformation techniques can be used. The synthesized images can be either obtained from datasets similar to the one used on the few-shot classification task [33] or even from large datasets that can be weakly labeled or even unlabeled for the target label on the few-shot classification task [34]. In figure 3.5, "transformer" refers to the transformation operation used on the supplemented data.

3.3.2 Meta-Learning

Meta-learning is present in a gross part of FSL algorithms. Meta-learning is a learning mechanism that comprehends two stages, a meta-training stage and a meta-testing stage. In the meta-training stage, training can be divided into various episodes, for instance, several classification events where a meta-learner trained on a few examples from various classes tries to classify an unseen sample in one of

those classes [32]. The objective is for the meta-learner to improve its capabilities of classifying unseen samples in base classes by acquiring knowledge from different classification tasks [35]. In the meta-testing stage, other classification tasks are presented to the meta-learner. Here, a few labeled images from novel classes are shown to the model to conclude if it can perform well in these situations. Using a similar notation to the one present in [32], each meta-training episode i consists of a training set D^i_{train} (also called support set) and a test set D^i_{test} (also called query set) where the classifier performance is measured. A meta-learner learns from various episodic tasks T_i. These episodic tasks are often N-way-K-shot classification tasks, i.e., the support sets consist of N classes, each class with K examples [35]. Figure 3.6 exemplifies a meta-learning framework. In this framework, there are several meta-training tasks, each one with a support set consisting of six images, two images per base class, and a query set with three images that match the seen classes. The testing stage presents various testing tasks where the test samples from the query set must be classified in one of the three novel classes present on the support set. This framework represents a 3-way-2-shot-classification problem.

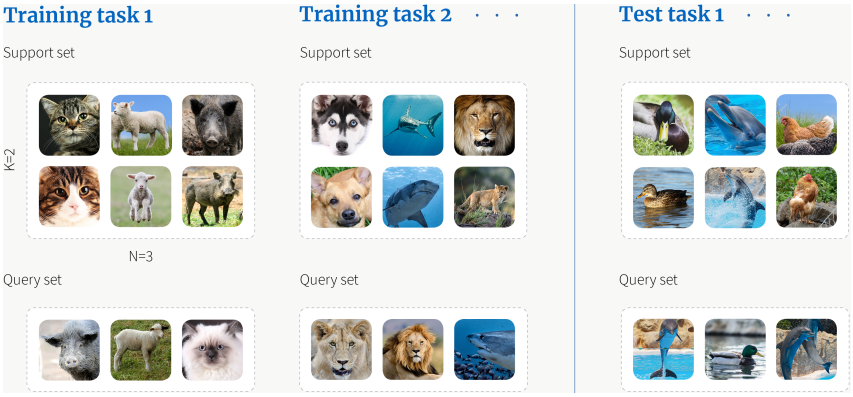


Figure 3.6: Meta-learning framework: 3-way-2-shot classification episodes [35].

3.3.3 Metric-Based FSL Algorithms

Metric-based FSL algorithms use distance metrics to compare the similarity of two images. In this subsection, a model that implements a linear classifier is presented, followed by an adaptation of this model, which is metric-based. These aforementioned models do not make use of meta-learning. Two metric-based classifiers that rely on meta-learning are also presented to close the subsection.

In [36], the authors propose two different models to solve the FSL problem, the "baseline model" and the "baseline++ model", respectively. These two metric-based methods are described using an adapted version of the notation in the mentioned work.

Figure 3.7 pictures the baseline model. In this model, the authors proposed to train a CNN network on a large labeled dataset with various classes. The objective was to use this CNN to extract features

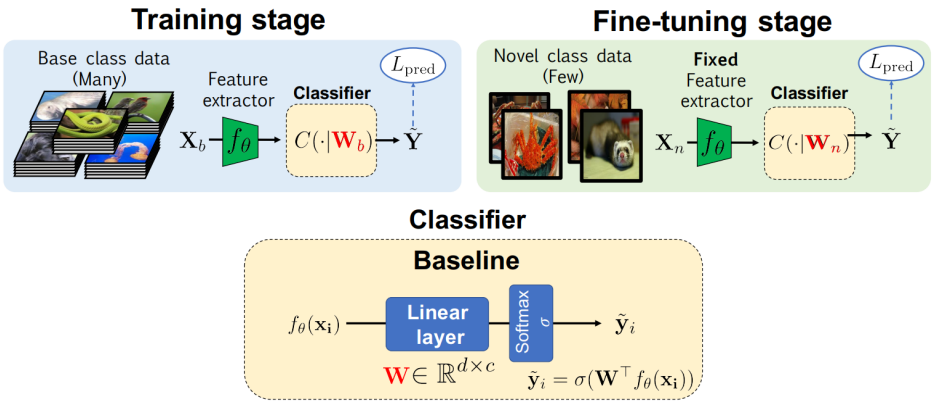


Figure 3.7: Baseline model (Using the nomenclature of [36]).

from the input images. In the proposed model, the feature extractor (f_θ) is parametrized by the network's parameters θ, and the extracted features from the input images are represented by $f_\theta(x_i)$. The extracted features enter a classifier ($C(.|W_b)$) parametrized by a weight matrix W_b. The weight matrix W_b consists of c column vectors $w_k, k \in \{1, ..., c\}$ of dimension d. The k_{th} column vector corresponds to class k. The classifier performs the linear operation $W_b^\top f_\theta(x_i)$ and then applies the softmax function (σ), converting the result to the probability of the image being classified in each of the base classes present during training. The training is based on minimizing the cross-entropy loss (L_{pred}). To fulfill a few-shot classification task, where examples from novel classes are present to the network, the authors resort to transfer learning. The network's layers were frozen, and the classifier was retrained. In sum, the network's weights were kept the same, and only the parameters of the weight matrix (now W_n as it refers to novel classes) were refined for the new classification task.

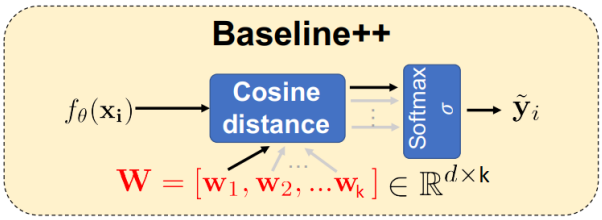
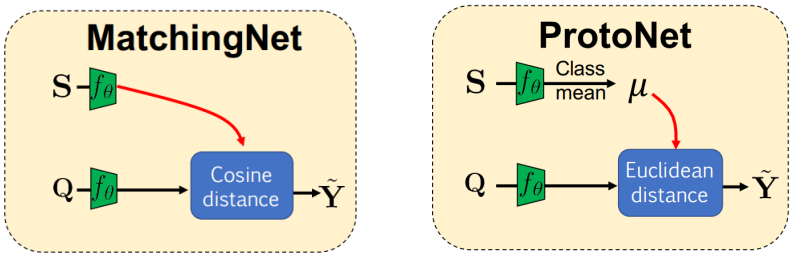


Figure 3.8: Baseline++ classifier [36].

The baseline++ model differs from the baseline model only in the way that the classifier is designed (figure 3.8). In this alternative model, the authors proposed to use a similarity function. During training, for each vector of features $f_\theta(x_i)$, the cosine similarity is computed concerning the weight vectors w_k, obtaining k similarity scores, one for each class. Each similarity value is obtained through

$$s_{i,k} = \frac{f_\theta(x_i)^\top w_k}{||f_\theta(x_i)|| \cdot ||w_k||}. \tag{3.4}$$

The usage of this similarity function allows the reduction of intra-class variation during the training, which might be relevant when there are domain differences in data.



(a) **(b)**

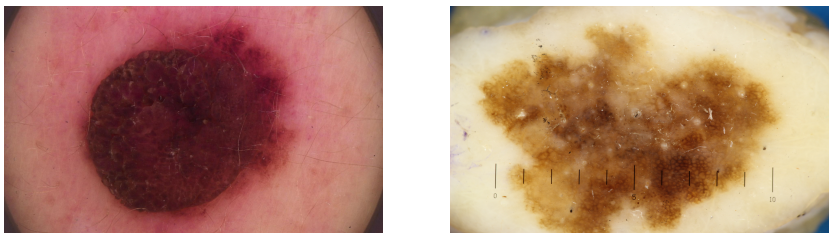
Figure 3.9: MatchingNet (a) and ProtoNet (b) [36].

The MatchingNet [37] (figure 3.9(a)) and the ProtoNet [38] (figure 3.9(b)) use distance metric-based classifiers that are put together with the concept of meta-learning, producing FSL algorithms. The idea behind the MatchingNet classifier is similar to the baseline++ classifier. As previously explained, in meta-learning, the available data is sampled and divided into several classification tasks. On each task, there will be a few examples of some base classes on the support set. In the case of a MatchingNet, for each task, the features of the images on the support set are compared to those on the query set by the cosine similarity. On top of that, the average cosine similarity is calculated for each class. In the case of a ProtoNet, the idea is to use the feature vectors extracted from the support set to compute class prototypes and then to measure the euclidean distance from each class prototype (μ_k) to the features extracted from each query set image ($f_\theta(x_i^Q)$),

$$d(f_\theta(x_i^Q), \mu_k) = ||(f_\theta(x_i^Q) - \mu_k)||_2. \tag{3.5}$$

3.4 Domain Adaptation

In the context of this thesis, DA techniques will be used because one of the challenges in this work is to build a model which can perform well on datasets that come from different domains (*in vivo* and *ex vivo*). Figure 3.10(a) is one example of an *in vivo* dermoscopic image, and figure 3.10(b) is one example of an *ex vivo* dermoscopic image.



(a) (b)

Figure 3.10: *In vivo* dermoscopic image of a benign lesion (a) [9]; *Ex vivo* dermoscopic image of a melanoma (b) (private *ex vivo* dataset).

When it comes to medical imaging, the majority of ML methods consider that the data distribution in the training and test sets is the same, which is deceptive most of the time. This assumption results in worse performance for a model. The increase of test error as the distribution difference between the training and test sets accentuates is called the domain shift problem. The domain shift problem is associated with models which, for instance, analyze different types of images such as magnetic resonances and tomographies or with models which analyze only one type of image but from different data centers (multi-site data). In this last situation, the model is affected because the datasets are obtained using different scan technologies and/or methods. DA techniques are useful because they allow to minimize the distribution difference between different but related domains [39].

DA can be seen as a particular case of transfer learning. Transfer learning is, according to [20], the concept in which a model is trained in a specific domain (source domain) or task and evaluated in a different domain (target domain) or task. The domain is defined as a feature space together with a marginal probability distribution and a task as a group of labels together with a predictive function learned in the training process [40]. In the DA case, the task is the same in both the source and the target domain, but the marginal distribution of features within the domains differs [41]. In [39], DA procedures are categorized as either shallow or deep and additionally as supervised, semi-supervised and unsupervised, in conformity with the availability of labels in the target domain. Subsections 3.4.1 and 3.4.2 present distinct methods to solve the domain shift problem, following the taxonomy in [39].

3.4.1 Shallow Models

Traditionally, shallow models are ML models that work on human-engineered features and use conventional ML methods. There are two common techniques in the scope of shallow DA. These two strategies are instance re-weighting and feature transformation [39].

Figure 3.11 illustrates an example of instance re-weighting on a binary classification problem which is solved with a support vector machine.

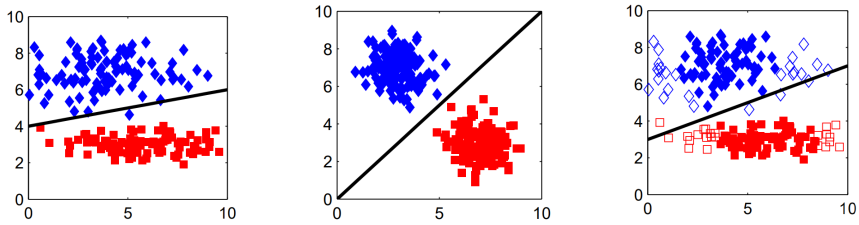


Figure 3.11: Instance re-weighting technique: Source domain (a); Target domain (b); Re-weighted source domain (c) [42].

In figures 3.11(a) and 3.11(b), it is noticeable that the data distributions are quite different. To fix this, instance re-weighting takes place. Instance re-weighting allows assigning different weights to the samples in the source domain according to their relevance in the target domain. The more relevant the sample, the larger the weight. In figure 3.11(c), the relevant samples are represented by filled markers having a larger weight assigned to them while the other samples are down-weighted, represented as unfilled markers. This procedure results in an attenuation of the domain shift. A classifier trained in the re-weighted source domain (figure 3.11(c)) is expected to have a better performance in the target domain.

In the feature transformation strategy, a feature space common to both the source and target domain is built. This space has the objective of reducing the shift between the data distribution of the domains. A model trained in this new feature space is, therefore, less affected by the domain shift problem. One of the techniques used in this context is, for instance, Low-Rank Representation (LRR). Methods based on LRR do not require labeled data in the target domain; thus, these techniques are worth exploring for medical-related problems since image labels in this area are scarce. LRR of the data opposes techniques that learn domain-invariant features. Take equation (3.6) present in [43], which formulates a simple LRR when adaptation between only one source domain and one target domain is intended.

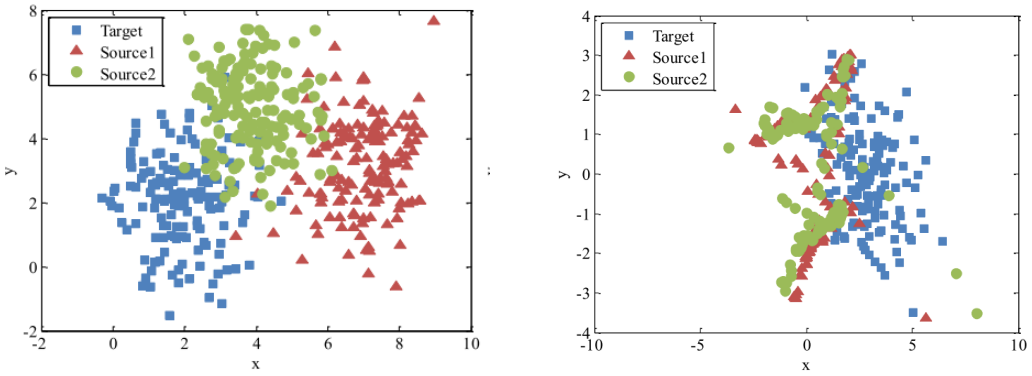
$$PX_S = X_T Z + E. \tag{3.6}$$

Using a similar notation to the one in [43], $X_S \in \mathbb{R}^{d \times n_s}$ is the source domain, where d represents the dimension of the features and n_s the number of samples in this domain. $P \in \mathbb{R}^{d \times d}$ is the transformation matrix that maps the source domain to the target domain. The target domain is represented by $X_T \in \mathbb{R}^{d \times n_t}$, where n_t is the number of samples in this domain. $Z \in \mathbb{R}^{n_t \times n_s}$ is a coefficient matrix, and $E \in \mathbb{R}^{d \times n_s}$ is the error matrix. This equation shows that the data in the source domain can be linearly represented by the data in the target domain, reducing the domain shift. Equation (3.6) can be formulated as the optimization problem (3.7), also present in [43].

$$\begin{aligned} \min_{P,Z,E} \quad & rank(Z) + \alpha||E||_1 \\ \text{s.t.} \quad & PX_S = X_T Z + E \end{aligned} \quad . \tag{3.7}$$

This way, the problem formulation is more robust against data corruption. The parameter α controls the trade-off between the rank of matrix Z and the norm of the error matrix E.

If the problem in question is a classification problem, after the LRR for the data is obtained, a classifier can be trained on the transformed space. Figure 3.12 illustrates the results of using a feature transformation strategy.



(a) **(b)**

Figure 3.12: Feature transformation: Domains before the feature transformation (a); Domains after the feature transformation (b) [43].

In the example of figure 3.12, there are two source domains and one target domain, each one with different data distribution (figure 3.12(a)). After feature transformation is applied, the data distribution in the three domains becomes more similar (figure 3.12(b)).

3.4.2 Deep Models

Image analysis progress in the past years is thanks to CNNs, which show to be capable of learning low-level image features. Unfortunately, CNNs also face the domain shift problem. One of the ideas behind DA with deep models is to extract image features with CNNs and then process these features using shallow DA methods. Another idea is to transfer a model from the source to the target domain using a technique called fine-tuning.

Fine-tuning consists of training a CNN in one source domain with labeled data and then retraining the last layers of the network using some labeled data from the target domain.

Figure 3.13 schematizes a network employed by Ghafoorian *et al.* [44] for the magnetic resonance of brain lesions segmentation. In their work, the network was trained in a source domain, and then i

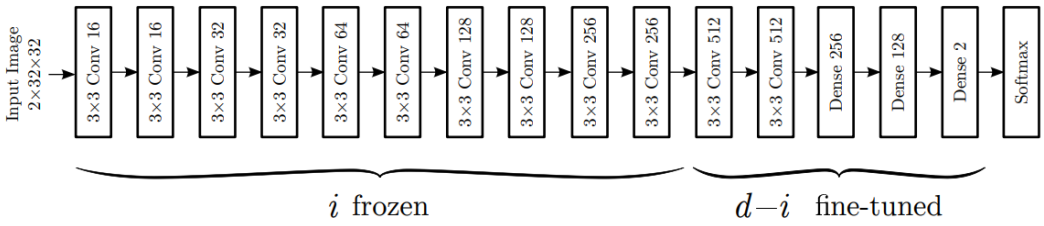


Figure 3.13: CNN architecture proposed by Ghafoorian *et al.* [44].

layers were frozen, letting $d - i$ layers available to be retrained with some labeled samples of the target domain.

Image analysis in medicine has one major issue: the lack of labeled data. This lack of labeled data is primarily due to the time it takes to annotate images and the money it costs, so techniques like fine-tuning might not do the trick in these situations. The field associated with techniques that perform DA when no labeled data in the target domain is available for training is designated unsupervised DA.

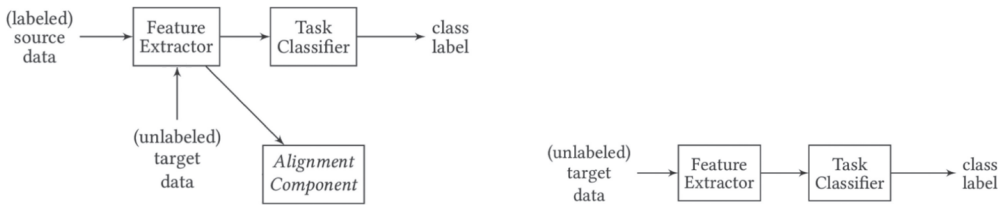


Figure 3.14: DA method with invariant feature learning: Training stage (a); Testing stage (b). (Images adapted from [20]).

A group of techniques that can address unsupervised DA aims to align features, more precisely, to learn domain-invariant features (figure 3.14). The objective is to learn features that have the same distribution independently of the domain they belong to. A classifier trained with domain-invariant features in the source domain will most likely perform well in the target domain as the features in which it was trained match the ones in the other domain [20].

The architecture presented in figure 3.14 represents a general description of these methods. Implementations of this architecture differ in the "Alignment Component" block and in the feature extraction procedure. As shown in figure 3.14, during the training stage, labeled source data and unlabeled target data enter the "Feature Extractor" block. The extraction can be performed by means of a CNN. The extracted features from the source and target domains enter the "Alignment Component" block to minimize the domain shift. Also, the source features are used to train a classifier. The source features will be domain-invariant if the alignment is successful, implying that the classifier is trained on domain-invariant features. After the training is complete, the feature extractor can be used on the unlabeled target data

to extract relevant features from this dataset; plus, the task classifier can predict the labels for the target data (figure 3.14(b)).

Strategies that **minimize divergence** are one option for the "Alignment Component" block. For instance, **Correlation Alignment (CORAL)** is a method that was first proposed by Sun *et al.* in [45]. The method aims to align second-order statistics of the domains by re-coloring whitened source features using the covariance of the target distribution [45]. A whitened feature vector is called this way because it behaves as white noise, having a covariance matrix equal to the identity matrix, meaning that the features of the vector are uncorrelated. When the authors of [45] claim that there is a re-coloring of whitened source features using the covariance of the target distribution, they mean, in other words, that source features which are uncorrelated to other source features (null covariance) will be modified so that they exhibit interdependence with the remaining features present in the feature vector. To re-arrange the features interdependence, CORAL loss (3.8) is introduced,

$$L_{CORAL} = \frac{1}{4d^2}\|C_S - C_T\|_F^2. \tag{3.8}$$

By measuring the distance between the second-order statistics of features extracted from images of the source and target domains, it is possible to compute the CORAL loss. In (3.8), d is the dimension of the extracted features, $\|.\|_F^2$ is the squared Frobenius norm, and C_S and C_T are the covariance matrices of the source and target domains respectively, obtained by

$$C_S = \frac{1}{n_S - 1}\left(D_S^\top D_S - \frac{1}{n_S}(\mathbf{1}^\top D_S)^\top(\mathbf{1}^\top D_S)\right), \tag{3.9}$$

and

$$C_T = \frac{1}{n_T - 1}\left(D_T^\top D_T - \frac{1}{n_T}(\mathbf{1}^\top D_T)^\top(\mathbf{1}^\top D_T)\right). \tag{3.10}$$

In (3.9), n_S is the number of feature vectors extracted from the source domain. In other words, it can be the number of source images that go through the network in one batch. D_S is a matrix where each row corresponds to a feature vector of dimension d. Analogously, in (3.10), n_T is the number of feature vectors extracted from the target domain, and D_T is a matrix where each row corresponds to a feature vector of dimension d. In both expressions, $\mathbf{1}$ corresponds to a column vector where all elements are equal to one.

During the training two losses are minimized, the classification loss and the CORAL loss

$$L_{TOTAL} = L_{CLASS} + \sum_{i=1}^{t} \lambda_i L_{CORAL}^i. \tag{3.11}$$

The classification loss is also computed during training, being obtained by training the task classifier on the source domain features. In (3.11), t stands for the number of layers in which the CORAL loss is

computed, and the hyperparameters λ_i work as trade-offs between the classification and the adaptation tasks.

Instead of minimizing a divergence, another option is to resort to **adversarial training strategies** to perform feature alignment. For example, the "Alignment Component" block of figure 3.14(a) can consist of a domain discriminator (domain classifier).

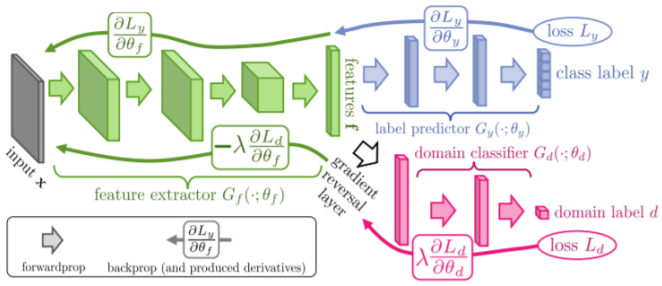


Figure 3.15: DA method with invariant feature learning using a domain classifier as the alignment component [46].

Figure 3.15 illustrates the architecture proposed by Ganin *et al.* [46] to perform unsupervised DA. This architecture is designated by **Domain-Adversarial Neural Network (DANN)** and it is composed by three different blocks (very similar to the architecture of figure 3.14(a) but now using an adversarial approach). The first block is a feature extractor which receives an input x. This input consists of $N = n + n'$ images, where n images are from the source domain ($x_i, i \in \{1, ..., n\}$) and n' images are from the target domain ($x_i, i \in \{n+1, ..., N\}$). The feature extractor ($G_f(:, \theta_f)$), parametrized by the network's weights θ_f, together with the label predictor ($G_y(:, \theta_y)$), parametrized by θ_y, form a regular feed-forward network. The label predictor receives only the portion of features that correspond to source domain images and predicts their class label. The last component of this architecture is a domain classifier ($G_d(:, \theta_d)$), parametrized by θ_d. The domain classifier receives as input features extracted from both domains and aims to classify the features as source domain features or as target domain features. If the training intention were to minimize the error associated with the label predictor and the error associated with the domain classifier, then the domain classifier would force the feature extractor to extract dissimilar features across domains.

Since the objective is to obtain indistinguishable features while maintaining good performance on the label predictor, a Gradient Reversal Layer (GRL) is introduced, connecting the feature extractor to the domain classifier. The GRL layer is responsible for the adversarial training. During the forward pass, this layer acts as a simple identity transformation. However, during the backward pass, it multiplies the gradients by a negative constant before sending them to the feature extractor, which results in competition between the feature extractor (which tries to induce in error the domain classifier) and the domain classifier (which tries to correctly predict the domain labels). Taking on the expressions present in [46], the DANN training is mathematically expressed as follows:

Consider the prediction loss of a source example ($x_i, i \in \{1, ..., n\}$),

$$L_y^i(\theta_f, \theta_y) = L_y(G_y(G_f(x_i; \theta_f); \theta_y), y_i), \tag{3.12}$$

and the domain loss of a source or target sample ($x_i, i \in \{1, ..., N\}$),

$$L_d^i(\theta_f, \theta d) = L_d(G_d(G_f(x_i; \theta_f); \theta_d), d_i). \tag{3.13}$$

The DANN training is conducted with the objective of optimizing (3.14)

$$E(\theta_f, \theta_y, \theta_d) = \frac{1}{n} \sum_{i=1}^{n} L_y^i(\theta_f, \theta_y) - \lambda \left(\frac{1}{n} \sum_{i=1}^{n} L_d^i(\theta_f, \theta_d) + \frac{1}{n'} \sum_{i=n+1}^{N} L_d^i(\theta_f, \theta_d) \right), \tag{3.14}$$

where λ actuates as the trade-off between the domain classification and the label prediction tasks. Since the objective is to minimize the prediction loss and to maximize the domain classification error, the optimization of (3.14) must lead to a saddle point that satisfies

$$(\hat{\theta}_f, \hat{\theta}_y) = \underset{\theta_f, \theta y}{\operatorname{argmin}} E(\theta_f, \theta_y, \hat{\theta}_d), \tag{3.15}$$

and

$$\hat{\theta}_d = \underset{\theta_d}{\operatorname{argmax}} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d). \tag{3.16}$$

In the best-case scenario, the training leads to a domain classifier that achieves 50% accuracy and a label predictor that attains good prediction results. In the end, the feature extractor prevents the domain classifier from correctly predicting the domain labels while still allowing a discriminatory representation for the prediction task. If the domain classifier cannot distinguish between the input features' domains, then the domains are aligned, and the shift is minimized.

4

Proposed Approach

Contents

This chapter starts by exposing a general view of the problem in question so that the reader can fully understand the following sections. Secondly, the pre-training methodologies for the CNNs are covered, followed by a section that explains the algorithms used to address the BRAF status prediction.

4.1 Outline

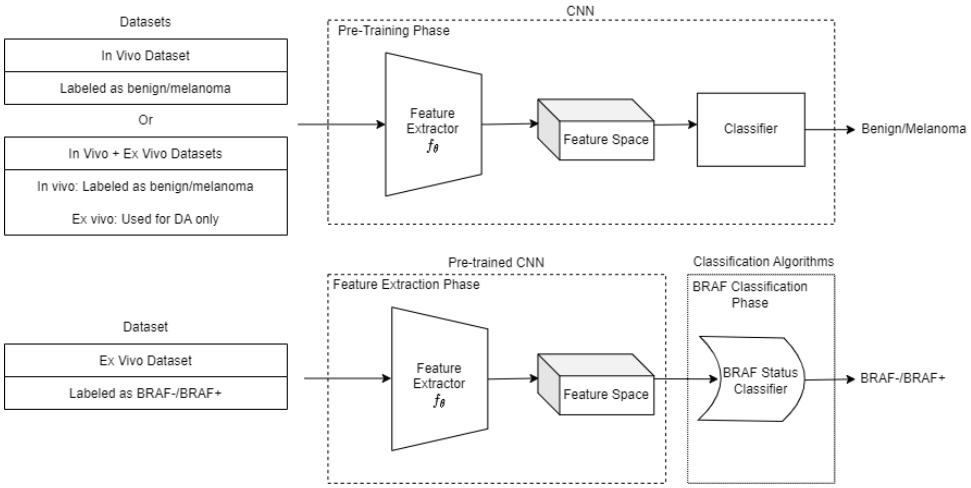


Figure 4.1: Proposed pipeline to address BRAF status prediction.

The main objective of this thesis is to predict the mutational status of the BRAF gene in melanoma lesions using faster and more automated methods than the widely used PCR analysis, complementing this procedure. An approach that aims to employ techniques based on DL to analyze dermoscopic images is suggested. The pipeline in figure 4.1 outlines the proposed approach to tackle the described objective.

Two different dermoscopic datasets are available for this work, one is *in vivo*, and the other is *ex vivo*. The *in vivo* data is vast, and the lesions are labeled as benign/melanoma, while the *ex vivo* images are scarce and correspond to melanomas labeled for the BRAF status (BRAF-/BRAF+) and a few benign lesions. In this thesis, it is intended to predict the mutational status of the BRAF gene. However, the only dataset conveying this information is both small and *ex vivo*. Therefore, there is a need to overcome the few data problem and most likely a domain shift problem, and this is where techniques inspired by FSL and DA will intervene.

As figure 4.1 outlines, the general idea is to pre-train CNNs on learning relevant information that can help identify the BRAF mutational status ("Pre-Training Phase"). In this phase, one option is to use the largest dataset (*in vivo*) alone. The other option is to use both datasets (the *in vivo* and the *ex vivo*) with the constraint of only using this last dataset to perform DA. At this stage, it is not intended for the model

to learn to classify the *ex vivo* data since it is scarce and will be needed later on to train/validate different classification algorithms.

After pre-training the CNN models, they will be used as feature extractors ("Feature Extraction Phase") for the *ex vivo* data, and after that, algorithms (most inspired by FSL) will be used to classify the *ex vivo* data as BRAF-/BRAF+.

The following sections explain the different phases of the pipeline in greater detail.

4.2 Pre-Training Phase

The "Pre-Training Phase" is where the CNNs learn to extract features from different images to obtain knowledge to perform the BRAF classification task. The main goal here is to attain CNNs which can extract relevant information from the input images, i.e., to build a discriminative latent space for the pretended task.

Two different approaches are considered in the "Pre-Training Phase". Both approaches are based on the concept of transfer learning. In the first approach, the CNN is trained on the *in vivo* data for a task closely related to the BRAF detection problem (the benign/melanoma classification task). This strategy is called **Related Task (RT)**. The RT pre-training strategy is kept for the second approach, but DA techniques are added. For this end, the *ex vivo* data is also considered. This second approach is named **RT & DA**, as it uses DA techniques together with the RT training. This approach encloses two strategies according to the selected DA method: the CORAL strategy (**RT & CORAL**) and the DANN strategy (**RT & DANN**).

4.2.1 Related Task

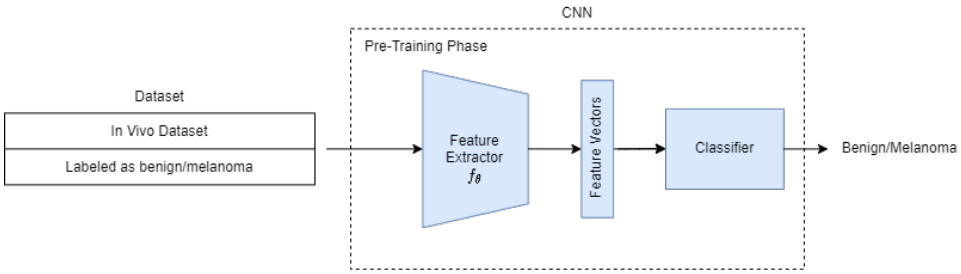


Figure 4.2: Pre-training on a RT (● : *In vivo*).

In this strategy (figure 4.2) the CNNs are trained for a task which is closely related to the BRAF mutational status prediction task. Here, the networks are trained with *in vivo* skin lesion images, learning to classify an image as benign or melanoma. Later on, the knowledge acquired in this task will be used

for predicting the BRAF status as there is evidence that some features are related to both tasks, as discussed in section 2.1.2.

4.2.2 Related Task & Domain Adaptation

Given the fact that the pre-trained models are used as feature extractors on *ex vivo* data, like the "Feature Extraction Phase" of figure 4.1 suggests, the pre-training approach proposed in subsection 4.2.1 might suffer from a domain shift problem. Therefore, a modified set-up, similar to the one in subsection 4.2.1, is suggested here. In this approach, a joint training technique is proposed.

With this method, the idea of pre-training the networks on the RT is maintained, adding on an "Alignment Component" block that minimizes the shift between the extracted features of the two different types of images (recall figure 3.14(a)). The type of architecture used for DA varies in conformity with the choice for the "Alignment Component" block. In this thesis, two strategies are considered. One is **divergence-based** (CORAL loss minimization, **RT & CORAL strategy**), and the other one is based on **adversarial training** (DANN architecture, **RT & DANN strategy**).

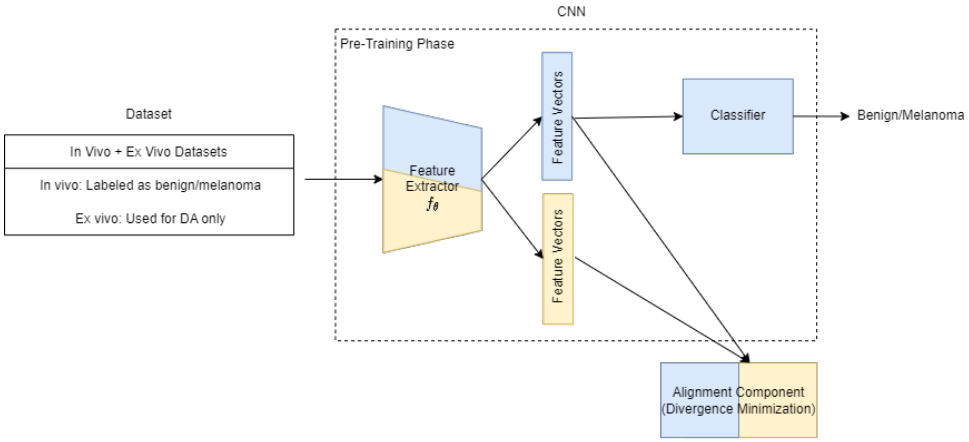


Figure 4.3: RT & CORAL pre-training strategy (● : *In vivo*, ● : *Ex vivo*).

- **RT & CORAL Strategy**: Figure 4.3 illustrates this methodology, which is based on the CORAL loss minimization method proposed by Sun *et al.* [47]. The feature extractor extracts features of images from both domains. The *in vivo* features enter the "Classifier" block to train the model on the RT. The *ex vivo* features enter the "Alignment Component" block together with the *in vivo* features. In this block, the CORAL loss is computed. During backpropagation the training objective is to both minimize the classification loss on the RT and to minimize the CORAL loss, which measures the divergence across the *in vivo* and *ex vivo* domains.

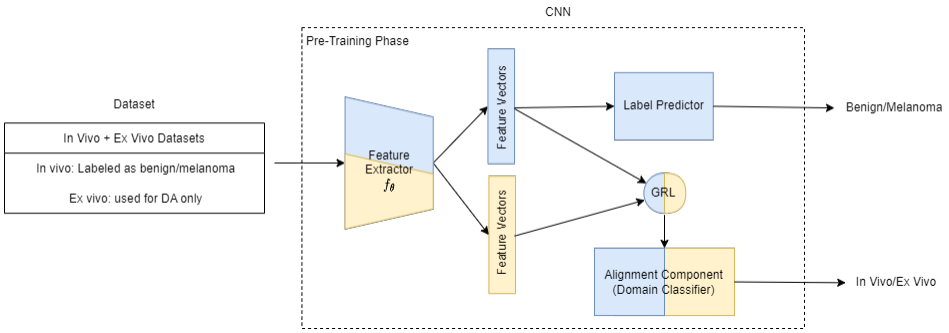


Figure 4.4: RT & DANN pre-training strategy (● : *In vivo*, ● : *Ex vivo*).

- **RT & DANN Strategy**: The scheme of this methodology is present in figure 4.4 and is based on the DANN architecture introduced by Ganin *et al.* [46]. A feature extractor receives the two types of images available for this research, the *in-vivo* data and the *ex-vivo* data. The output of this extractor is a set of feature vectors for the *in vivo* data and another for the *ex vivo* data. The extracted data must be, first of all discriminative enough to perform the RT and secondly needs to be domain-invariant. To achieve both goals, at the output of the feature extractor there is a bifurcation, one path leads the *in vivo* features to the RT classifier while the other path leads both *in vivo* and *ex vivo* features to the "Alignment Component" block which consists of a domain classifier. The domain classifier is trained to distinguish the two domains, and the benign/melanoma classifier block ("Label Predictor") is trained to minimize the classification error on the *in vivo* data. A GRL connecting the "Feature Extractor" to the "Alignment Component" block ensures that the adversarial training takes place. The feature extractor is forced to learn domain-invariant but discriminative features that can trick the domain classifier (minimizing the shift) and produce correct classifications for the RT at the same time.

4.3 Feature Extraction and BRAF Classification Phases

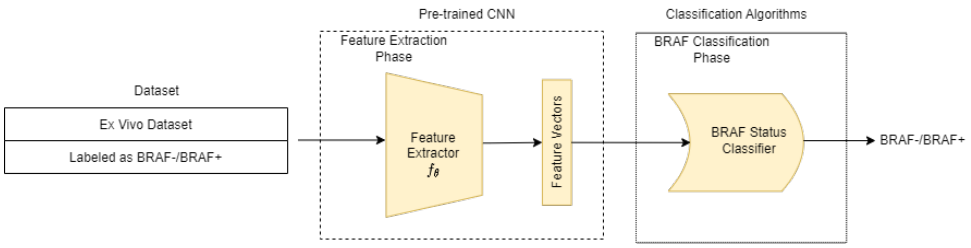


Figure 4.5: Feature extraction & BRAF status classification (● : *Ex vivo*)

After the pre-training stage, the networks can be used as feature extractors on the *ex vivo* data la-

beled for the BRAF status ("Feature Extraction Phase"). The extracted features are then processed by external classification algorithms during "BRAF Classification Phase". Figure 4.5 illustrates this procedure.

4.3.1 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm [48] is one of the simplest algorithms used for classification. The idea is to compare a test sample (test feature extracted from *ex vivo* data) to the training samples (training features also extracted from the *ex vivo* dataset). The label assigned to the test example will be the same as the most prevalent class of the nearest k training examples.

In the conducted research, the comparison between feature vectors is made by taking two different distance metrics into account: The euclidean distance (3.5) and the cosine similarity (3.4).

4.3.2 Prototype-Based Classifiers

The prototype-based classifiers are inspired in the FSL techniques presented in chapter 3, more specifically, they are inspired on the metric-based classification algorithms present on the MatchingNet [37] and on the ProtoNet [38].

ProtoNet Classifier With Euclidean Distance (P1) This is the classifier proposed in [38]. In this thesis, the feature extractor extracts the features from the *ex vivo* images. After this, the BRAF- and the BRAF+ features are gathered into two different groups. By averaging the features of each group, two prototypes are computed, one for the BRAF- class and another for the BRAF+ class. The euclidean distance between a test feature and each class prototype is calculated to predict the class for the test example. The label assigned to the sample is given according to the spatially closer prototype.

ProtoNet Classifier With Cosine Similarity (P2) As referred by Snell *et al.* [38], any distance metric is permissible, so instead of using the euclidean distance as the comparison metric, here, the cosine similarity between test samples and each class prototype is computed. After computing the cosine similarity between a test feature and each class prototype, the label assigned to the test example is given according to the prototype it resembles the most (the one to which the test feature vector exhibits a higher similarity value).

MatchingNet Classifier With Cosine Similarity (M1) This is the standard classifier proposed by Vinyals *et al.* [37]. In this strategy, after the features of the *ex vivo* dataset are extracted and divided into two different groups, the BRAF- and the BRAF+ group, the cosine similarity between a test example and each one of the samples in the two groups is computed. This leads to N similarity values with respect to the BRAF+ class and M similarity values with respect to the BRAF- class. To achieve a single similarity value per class, the average cosine similarities are computed by averaging the similarity values of each

class group. In the end, the class assigned to the test example is given according to the highest average similarity value.

MatchingNet Classifier With Euclidean Distance (M2) The idea behind this technique is the same as in the MatchingNet classifier with cosine similarity, only the comparison metric changes. Instead of computing the average cosine similarity per class, now the average euclidean distance per class is determined, and the label given to a test sample is in accordance with the group of features which is, on average, spatially closer.

4.3.3 Logistic Regression

For this approach, a Logistic Regression (LR) [49] is trained on the extracted features from the *ex vivo* data. A test sample can then be fed to the LR equation to predict the mutational status of the patient associated with the extracted features.

The LR classifier is often used in binary classification tasks. In this thesis, the LR is used to model the a posteriori probabilites of the BRAF+

$$P(y = 1|v) = \pi = \frac{1}{1 + e^{-v^\top \beta}}, \tag{4.1}$$

and BRAF- classes,

$$P(y = 0|v) = 1 - \pi = \frac{e^{-v^\top \beta}}{1 + e^{-v^\top \beta}}. \tag{4.2}$$

In (4.1) and (4.2), vector $v \in \mathbb{R}^{d+1}$ denotes the extracted feature vector (an extra element of value 1 is considered because of the offset value β_0). Vector v multiplies with a vector of regressor coefficients β. The logit operation

$$logit(\pi) = ln\left(\frac{\pi}{1-\pi}\right) = v^\top \beta = \beta_0 + v_1\beta_1 + ... + v_d\beta_d, \tag{4.3}$$

in this case defines the odds ratio between the probability of having a BRAF+ and the probability of having a BRAF- sample, given the extracted features.

Besides the classical LR model, others are considered, namely LR with a regularization parameter. The regularizers contemplated for this strategy are the $L1$ norm ($L1$ penalty) and the $L2$ norm ($L2$ penalty).

5

Experimental Set-Up

Contents

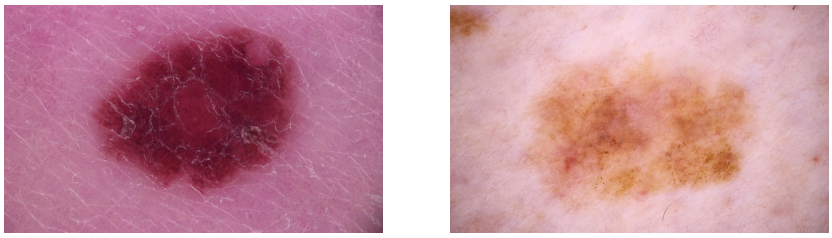
Chapter 5 introduces the datasets used for the experiences. After explaining the pre-processing applied to the data, the pre-training configurations are presented. Finally, the metrics used to evaluate the performance of the proposed methodologies are introduced, followed by the formulation of the implementation challenges for this thesis.

5.1 Datasets

For this work, dermoscopic images from different data centers are available. The largest dataset is public and consists of *in vivo* skin lesions (ISIC 2020 dataset [9]). The smallest dataset is private and contains 138 *ex vivo* dermoscopic images of skin lesions.

5.1.1 Public *In Vivo* Dataset

The ISIC 2020 dataset [9] consists of $33,126$ *in vivo* dermoscopic images of skin lesions, of which $32,542$ are benign lesions and 584 are melanomas. The malignant diagnoses were performed using histopathology, while the benign diagnoses were performed either through expert agreement, longitudinal follow-up, or histopathology. The images that are part of this dataset come from the Hospital Clínic de Barcelona, the Medical University of Vienna, the MSKCC in New York City, the MIA in Sydney, the University of Queensland in Brisbane, and the University of Athens Medical School. Thus, this dataset presents multi-source data and the images have different colors, sizes, and aspect ratios. Figures 5.1(a) and 5.1(b) are drawn from the ISIC 2020 dataset and correspond to a benign lesion and a malignant lesion.



(a) **(b)**

Figure 5.1: *In vivo* dermoscopic images from the ISIC 2020 dataset [9]: Benign lesion (a); Malignant lesion (b).

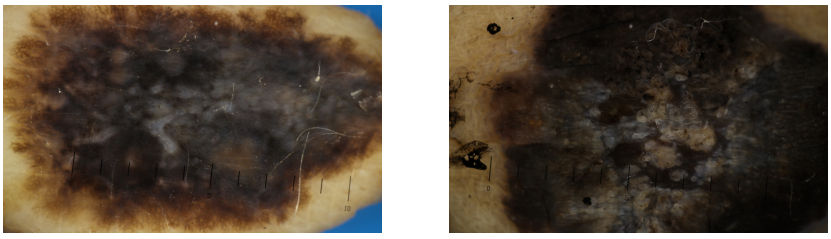
5.1.2 Private *Ex Vivo* Dataset

The private *ex vivo* dataset is small, comprehending 138 *ex vivo* dermoscopic images of skin lesions. Of these lesions, only 69 are melanomas and labeled for BRAF status. This portion of the dataset will be

denoted the "BRAF dataset". The remaining 69 images from the 138 are benign and unlabeled for BRAF status.

The 69 images that comprehend the BRAF dataset correspond to 43 melanoma patients, meaning that patients may present more than one image for the same lesion, depending on its size. Of the 43 patients, 23 are men (53.49%), and 20 are women (46.51%). The average age of the patients is 69.60 ± 11.45 years old. This dataset is also imbalanced because there are 31 BRAF- patients $(48$ images) and 12 BRAF+ patients $(21$ images). Figure 5.2(a) corresponds to a BRAF- melanoma drawn from the BRAF dataset and 5.2(b) corresponds to a BRAF+ melanoma drawn from the same dataset.

The private *ex vivo* dataset is used for two purposes. The first purpose is to perform BRAF status classification, and for this, only the BRAF dataset portion is considered. The other purpose is to perform DA; to do so, the total amount of 138 images is taken into account.



(a) **(b)**

Figure 5.2: *Ex vivo* dermoscopic images of melanomas: BRAF- melanoma (a); BRAF+ melanoma (b).

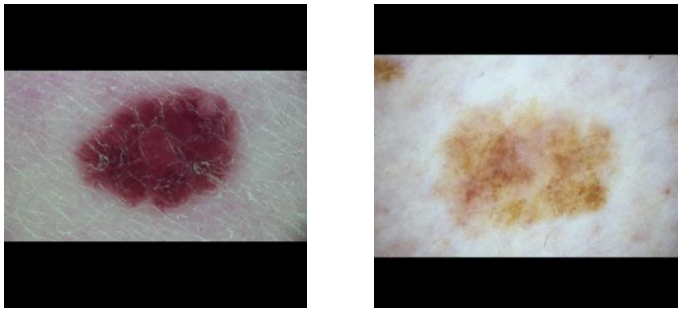
5.1.3 Pre-Processing

From the $33,126$ images that comprehend the ISIC 2020 dataset, 425 are duplicates. The duplicates are discarded and the remaining data is split into two different sets. A training set containing $26,041$ benign lesions and 464 malignant lesions $(80\%$ of the whole dataset), and a validation set containing $6,079$ benign lesions and 117 malignant lesions $(20\%$ of the whole dataset).

As mentioned in subsection 5.1.1, the ISIC 2020 dataset contains images from different data centers. Usually, different centers operate under different illumination conditions and have disparate image acquisition devices; thus, the resulting data from the various centers present different coloring. The change in color produced by the aforementioned factors results in a performance downgrade for the computer-aided systems, which intend to either extract information of dermoscopic images or classify them [50]. To attenuate this effect, before training or validating the CNN architectures using the ISIC 2020 dataset, the Shades Of Gray algorithm [51] (color constancy algorithm) is applied.

Besides the color change, the acquired images present different resolutions and sizes. On top of the Shades of Gray transformation, all the images are resized to a 300×300 resolution before being fed to

the CNN architectures and padded with two horizontal black borders to preserve the original aspect ratio as well as the information. Figures 5.3(a) and 5.3(b) illustrate the pre-processed versions of the images in figure 5.1.



(a) **(b)**

Figure 5.3: Transformed *in vivo* dermoscopic images from the ISIC 2020 dataset [9]: Pre-processed benign lesion (a); Pre-processed malignant lesion (b).

As for the *ex vivo* data, the same resizing and padding operation applied to the ISIC 2020 dataset is used. The Shades Of Gray operation is not applied to the private *ex vivo* dataset, mostly because the skin that appears in the images is no longer vascularized, so the lesion color is very similar from image to image. Also, a gross part of the images presents the same blue background. Figure 5.4 shows the transformed version of the images in figure 5.2.

(a) **(b)**

Figure 5.4: Transformed *ex vivo* dermoscopic images of melanomas: Pre-processed BRAF- melanoma (a); Pre-processed BRAF+ melanoma (b).

5.2 Pre-Training Approaches Implementation

5.2.1 Architectures

Three different CNN architectures are chosen to perform the feature extraction procedure. The three networks come from three different families and present different depths. The selected architectures are the ResNet-18 [52], the EfficientNet-B2 [53], and the Inception-V3 [21].

5.2.2 Common Configurations

For all the proposed pre-training approaches some configurations are common, namely:

- Training set transformation: On-line data augmentation consisting of random horizontal and vertical flips as well as a random erasing in the images with a probability of 50% (scale = $(0.02, 0.33)$, ratio = $(0.3, 3.3)$);
- Feature extractor: ImageNet initialization for all the architecture's layers. The last fully-connected layer is replaced by a randomly initialized lesion classifier;
- Lesion classifier architecture: A simple fully-connected layer with the same number of input units as the extracted features' size and two output units. The fully-connected layer introduces a dropout with probability $p = 0.3$ (for the ResNet-18 and the EfficientNet-B2 architectures);
- Lesion classification loss: Categorical cross-entropy with 0.02 penalty for misclassifications in the benign class and 0.98 penalty for misclassifications in the melanoma class;
- Optimizer: Adam optimizer [54];
- Initial learning rate: 5×10^{-7} (ResNet-18), 1×10^{-6} (Inception-V3 and EfficientNet-B2).

5.2.3 RT Pre-Training Configuration

- Training duration: 100 epochs with an early-stop of patience = 30;
- Batch-size: 32 for the ResNet-18 and Inception-V3 architectures and 16 for the EfficientNet-B2 architecture.

5.2.4 RT & CORAL Pre-Training Configuration

- Domain loss: CORAL loss with a lambda value of 100 as this loss is initially approximately 100 times smaller than the classification loss;

- Training duration: 100 epochs controlled by the *in vivo* dataset's size;
- Batch-size: 64 for the ResNet-18, 32 for the Inception-V3, and 20 for the EfficientNet-B2.

5.2.5 RT & DANN Pre-Training Configuration

- Domain classifier architecture: A MLP composed of three layers. The input layer and the hidden layer present, at the input and at the output, the same number of units. This number of units equals the number of features in a feature vector. Between these two layers there is a dropout with probability $p = 0.3$. The output layer has an input size equal to the feature vectors' dimension and contains one output unit;
- Domain classification loss: Binary cross-entropy;
- Training duration: 100 epochs controlled by the *in vivo* dataset's size;
- Batch-size: 64 for the ResNet-18, 16 for both the EfficientNet-B2 and the Inception-V3.

5.3 Computational Environment

All the experiments were conducted using the Python programming language. The framework used to manipulate DL architectures was the $1.12.1$ version of Pytorch [55]. Other Python libraries such as Sklearn and Numpy were also frequently used to assess classification performance. The training of DL architectures was performed on a desktop with an Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz, 3601 Mhz, 4 Core(s), 8 Logical Processor(s) and a GeForce GTX 1060 6GB NVIDIA GPU.

5.4 Evaluation Metrics

5.4.1 Specificity

Specificity (SP) is the True Negative (TN) rate, i.e., the number of negative examples correctly classified as negative by the model divided by the total number of negative cases. This ratio is computed as shown in (5.1). It is obtained by dividing the correctly classified negative cases (TN) by the sum of cases the model classified wrongly as positive (the False Positive (FP) samples) with the TN cases,

$$SP = \frac{TN}{TN + FP}. \tag{5.1}$$

5.4.2 Sensivity

SE (also called recall) is the true positive rate, i.e., the number of positive examples correctly classified as positive by the model divided by the total number of positive cases. Meaning that the samples correctly classified as positive (True Positive (TP)) are divided by the sum of cases that the model classified wrongly as negative (the False Negative (FN) cases) with the TP cases. This ratio is calculated as exhibited in (5.2).

$$SE = \frac{TP}{TP + FN}. \tag{5.2}$$

5.4.3 Balanced Accuracy

The RT and the BRAF classification task are binary classification problems that use imbalanced data. Therefore, the use of Balanced Accuracy (BACC) to evaluate the models' performance is useful as it considers both the SP and SE values.

$$BACC = \frac{SE + SP}{2}. \tag{5.3}$$

5.4.4 Precision

Precision (PR) evaluates the reliability of a classification model to classify positive cases. As shown in (5.4), this metric is computed by dividing the TP cases by the total number of classifications in the positive class, the TP and the FP cases,

$$PR = \frac{TP}{TP + FP}. \tag{5.4}$$

5.4.5 F_1 Score

F_1 score is a metric that combines PR and SE. It is given by the harmonic mean of the two aforementioned metrics, as exemplified in (5.5).

$$F_1 = \frac{2 \times (PR \times SE)}{PR + SE}. \tag{5.5}$$

5.5 Implementation Challenges

5.5.1 Generalization Problem

The few available data to perform the BRAF status classification is a major issue. In this thesis, transfer learning from a RT is employed to empower the feature extractors without using more specialized classi-

fication pre-training approaches on the *ex vivo* data. Using the BRAF data for classification tasks rather than just for DA in the pre-training approaches would consume data that could not be used afterward for training and validating the BRAF classification algorithms.

To train the proposed BRAF classification algorithms that act on the information extracted by the CNNs, a leave-one-out cross-validation [56] is applied. This way, the obtained results are more robust. The algorithms are trained with the information extracted from 42 of the 43 BRAF patients and validated on the case that is left out. This process is repeated 43 times, one time per patient. In the end, the results per patient are collected, and the evaluation metrics computed.

5.5.2 Multiple Images Per Patient Problem

Given that on the BRAF dataset, a patient can present more than one image for the same lesion and a single diagnosis is intended, two different methods are proposed.

One of the methods is a **Per-Image analysis**. In this procedure, given a patient with multiple images, a per-image diagnosis is performed, and after that, to obtain the final diagnosis, one of the following criteria is adopted: The One-Dominance criterion (1D), where a patient is considered as BRAF+ if at least one of the images is classified as BRAF+. Or the Majority Voting criterion (MV), where the final diagnosis is equal to the most voted class for the patient´s images.

The second method is a **Summarized Features (SF) analysis**, where after extracting the features of every image for a patient, the information is summarized in a single vector either by computing the mean vector or the max-wise vector across all of the feature vectors' values. The first criterion is denoted the Mean criterion (mean) and the second, Max criterion (max). After obtaining the SF vector, the BRAF classification algorithms are applied to the summarized information to obtain a single diagnosis per patient.

6

Experimental Results and Discussion

Contents

This chapter presents the most relevant results attained during the work. First, the best results for BRAF status prediction are exhibited. After that, the performance of the models resulting from the different pre-training approaches on the RT is inspected. A deeper analysis of the results obtained using the ResNet-18 as a feature extractor for the BRAF classification task is supplemented. To close this chapter, the algorithm of Armengot-Carbó *et al.* [7] is also applied to the available BRAF dataset to compare the obtained results with the existing state of the art.

6.1 BRAF Classification: Best Results Overview

Table 6.1 summarizes the best results obtained for BRAF status classification using the ResNet-18 and the EfficientNet-B2 as feature extractors. The results for the Inception-V3 architecture can be found in appendix C as they were similar to the ones obtained for the EfficientNet-B2 both for BRAF status prediction and for the RT. Besides, during the pre-training stage, for all approaches, the Inception-V3 architecture has exhibited convergence issues, experiencing a performance drop very early during training compared to the ResNet-18 and the EfficentNet-B2 architectures.

Table 6.1: Best results for BRAF status prediction.

Per-Image Analysis						
Classifier	Pre-Trained Network	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
LR_{L1} - 1D	EfficientNet-B2: ImageNet	75.3	83.9	66.7	64.0	61.5
P2 - MV	ResNet-18: RT	62.1	74.2	50.0	46.2	42.9
SF Analysis						
Classifier	Pre-Trained Network	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
LR_{L1} - Max	EfficientNet-B2: ImageNet	77.8	80.7	75.0	66.7	60.0
P2 - Max	ResNet-18: RT	60.5	71.0	50.0	44.4	40.0

The best results were obtained using the EfficientNet-B2 as extractor and the LR with L1 penalty as BRAF classification algorithm. For this architecture, the different proposed pre-training approaches did not benefit the BRAF classification performance since the best results were obtained by initializing the architecture with the weights resultant from a pre-training on the ImageNet dataset.

The ResNet-18 architecture, unlike the EfficientNet-B2, benefited from the proposed pre-training approaches. The best results for this architecture were obtained by considering the RT pre-training approach and a prototype-based classifier (P2 classifier).

Despite the data limitation, it is possible to achieve reasonable results for the BRAF status prediction. Furthermore, it is interesting to note that for the two architectures, the selected BRAF status classifier is the same in the per-image and SF analyses.

There may be several reasons why one architecture does not benefit from the pre-training ap-

proaches, and the other one does. The most plausible one has to do with the fact that the EfficientNet-B2 and the ResNet-18 architectures come from different families, each with different building principles. The EfficientNet family was developed to achieve more efficient and accurate networks. A common practice to achieve better results for CNNs is to scale them up by depth, width, or even resolution. However, the EfficientNet architectures have a different scaling-up procedure as they are scaled up, keeping a balanced depth, width, and resolution. Also, keeping a reduced number of learnable parameters. On the other side, the ResNet architectures are scaled up or down by just adjusting the number of layers (scaled in depth) [53]. The ResNet family also uses the concept of residual learning to mitigate the effect of accuracy saturation as the networks grow deeper [52].

6.2 Performance on the RT

During the pre-training stage, the networks were trained to extract features from dermoscopic images. In this stage, the networks were trained to predict whether a dermoscopic image was a benign lesion or a melanoma. The results obtained on the RT, for the different pre-training approaches (RT, RT & CORAL, and RT & DANN) can be observed in table 6.2 (ResNet-18 architecture), and in table 6.3 (EfficientNet-B2 architecture).

Table 6.2: ResNet-18 performance on the RT. For the *in vivo* performance, the networks were evaluated on the **ISIC 2020 validation set** ($6,196$ images: $6,079$ benign, 117 melanomas), whereas for the *ex vivo* performance, the networks were evaluated on the **private *ex vivo* dataset** (138 images: 69 benign, 69 melanomas).

***In Vivo* Performance**					
Pre-Train	Metrics				
	BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
RT	**75.6**	**87.9**	**63.3**	**15.9**	**9.1**
RT & CORAL	73.6	93.3	53.9	21.4	13.4
RT & DANN	73.1	91.6	54.7	18.4	11.1
***Ex Vivo* Performance**					
Pre-Train	Metrics				
	BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
RT	54.4	10.1	98.6	68.3	52.3
RT & CORAL	**59.4**	**52.2**	**66.7**	**62.2**	**58.2**
RT & DANN	51.5	98.6	4.4	8.2	75.0

Table 6.3: EfficientNet-B2 performance on the RT. For the *in vivo* performance, the networks were evaluated on the **ISIC 2020 validation set** ($6,196$ images: $6,079$ benign, 117 melanomas), whereas for the *ex vivo* performance, the networks were evaluated on the **private *ex vivo* dataset** (138 images: 69 benign, 69 melanomas).

***In Vivo* Performance**					
Pre-Train	Metrics				
	BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
RT	**76.3**	**91.8**	**60.7**	**20.8**	**12.5**
RT & CORAL	73.6	95.9	51.3	28.0	19.3
RT & DANN	71.1	95.1	47.0	23.4	15.6
***Ex Vivo* Performance**					
Pre-Train	Metrics				
	BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
RT	47.8	23.2	72.5	58.1	48.5
RT & CORAL	**59.4**	**43.5**	**75.4**	**65.0**	**57.1**
RT & DANN	55.8	100.0	11.6	20.8	100.0

The best performance on the benign/melanoma classification task for the *in vivo* dataset is obtained by the simple RT pre-training. However, when it comes to the private *ex vivo* dataset, the pre-training that adds CORAL loss minimization between the *in vivo* and *ex vivo* domains (RT & CORAL) attains the best performance, costing only a small performance drop on the *in vivo* data when compared to the RT pre-training case. For the *ex vivo* data, the RT & CORAL pre-trained network exhibits balanced values of SP and SE, contrary to what happens in the two other pre-training approaches where there is either a high SE value and a low SP value (on the RT case) or a high SP value and a low SE value (on the RT & DANN case).

To visually interpret the influence of the different pre-training approaches on the obtained feature spaces, UMAP [57] plots are presented. For the sake of simplicity, only the UMAPs obtained for the different pre-training approaches using the ResNet-18 architecture as the feature extractor are illustrated (the plots obtained using the other architectures were similar).

Figure 6.1 illustrates the UMAP obtained for the RT pre-training approach. The clusters in the figure show that even though the images from the ISIC 2020 validation dataset are all from *in vivo* lesions, shifts exist between the data extracted from the different data centers that constitute the ISIC 2020 data cohort. It is interesting to note that there is a superposition of the features extracted from the NYC center and the BNE center, as well as a superposition among the features extracted from the VIE and BCN centers. It is known that the NYC and the BNE centers use the same brand of screening devices. The same goes for the VIE and the BCN centers. Therefore, these reasons are the most credible arguments to justify the superpositions. As expected, the *ex vivo* data cluster is separated from the *in vivo* data clusters.

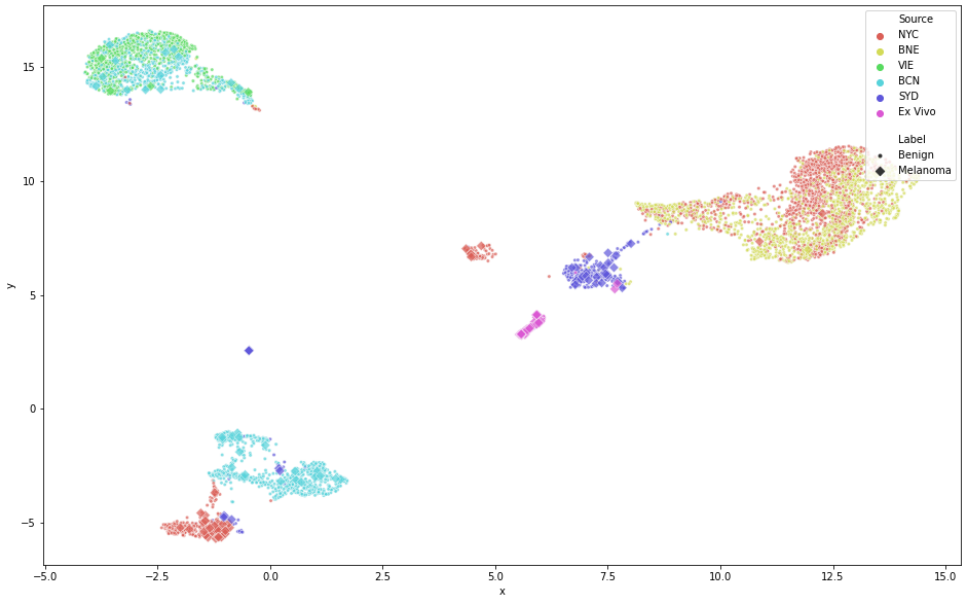


Figure 6.1: 2D UMAP plot of the features extracted from the **ISIC 2020 validation set** (5 sources: New York City (NYC), Brisbane (BNE), Vienna (VIE), Barcelona (BCN), and Sydney (SYD)), and from the **private *ex vivo* dataset**, using the **ResNet-18 architecture pre-trained on the RT** as feature extractor.

The main goal of the pre-training approaches RT & CORAL and RT & DANN is to minimize the occurrence of shifts between the *in vivo* and *ex vivo* data. The UMAPs obtained for these approaches are illustrated in figures 6.2 and 6.3.

The UMAP obtained by the ResNet-18 pre-trained on the RT & CORAL acting as a feature extractor (figure 6.2) does not exhibit a great difference compared to the UMAP of figure 6.1. This can lead to thinking that no domain alignment was performed between the *in vivo* and *ex vivo* data. Nevertheless, the UMAP plot might be misleading as the DA operation performed in this case was just a statistical alignment between the *in vivo* and *ex vivo* features. It is not surprising that this type of alignment passes unnoticed in an UMAP plot. Besides, it is always important to recall that the feature space learned by the ResNet-18 architecture is high dimensional (the extracted feature vectors have 512 dimensions). Therefore, a spatial reduction to two dimensions, such as the one performed by the UMAP transformation, might not contain all the information about the domain alignment.

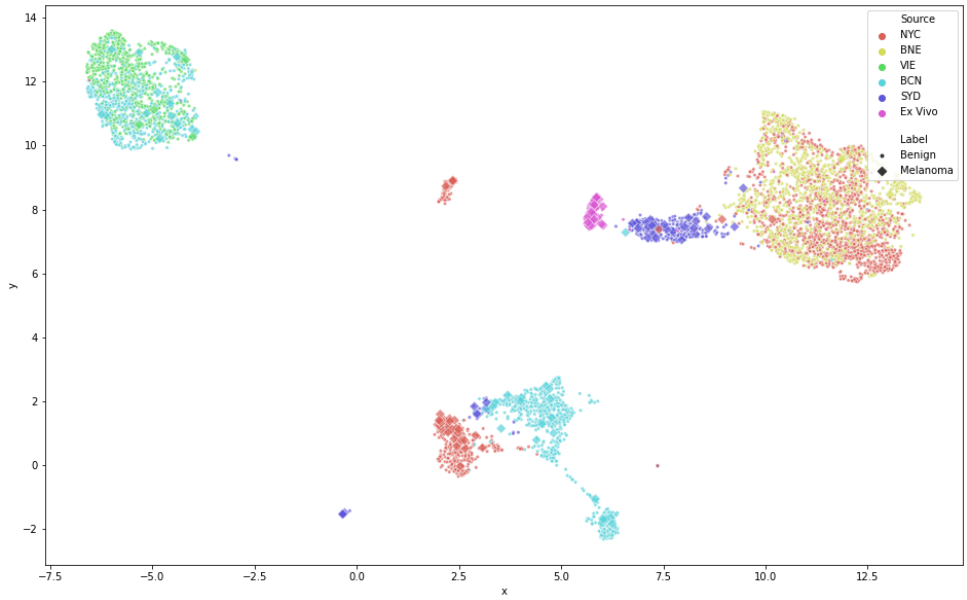


Figure 6.2: 2D UMAP plot of the features extracted from the **ISIC 2020 validation set** (5 sources: New York City (NYC), Brisbane (BNE), Vienna (VIE), Barcelona (BCN), and Sydney (SYD)), and from the **private *ex vivo* dataset**, using the **ResNet-18 architecture pre-trained on the RT & CORAL** as feature extractor.

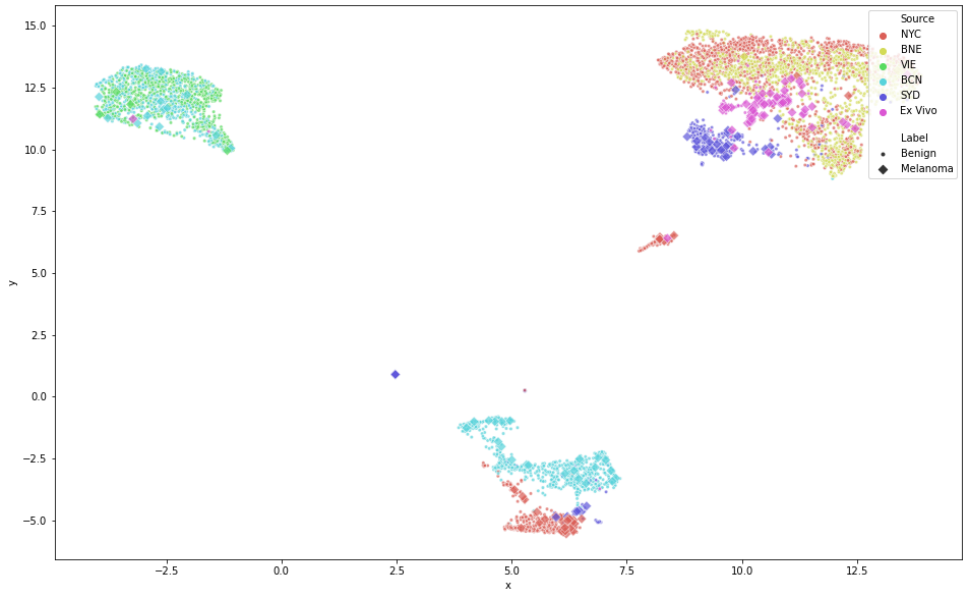


Figure 6.3: 2D UMAP plot of the features extracted from the **ISIC 2020 validation set** (5 sources: New York City (NYC), Brisbane (BNE), Vienna (VIE), Barcelona (BCN), and Sydney (SYD)), and from the **private *ex vivo* dataset**, using the **ResNet-18 architecture pre-trained on the RT & DANN** as feature extractor.

The UMAP obtained by considering the ResNet-18 architecture pre-trained on the RT & DANN (figure 6.3) gives clear evidence of domain alignment between the *in vivo* and *ex vivo* data. Unlike the RT & CORAL pre-training, the RT & DANN pre-training forces the extractor into learning features that can trick a domain classifier, resulting on the *ex vivo* data cluster translation observed in the UMAP of figure 6.3.

It is noticeable that the features extracted from the *ex vivo* data tend to align with those extracted from the centers that contain more images (the NYC and BNE centers). This is expected because the domain classifier received more data from these two centers during training than from any other. Besides, the gross part of the *ex vivo* melanomas align with *in vivo* benign lesions. This fact may be the reason behind the high SP and low SE values obtained for the *ex vivo* data when using the network pre-trained on the RT & DANN (table 6.2).

6.3 BRAF Classification Performance

The EfficientNet-B2 architecture initialized with the ImageNet weights led to better results on the BRAF classification task than models pre-trained on RT. Contrarily to the EfficientNet-B2, the ResNet-18 architecture benefited from the proposed pre-training approaches. Therefore, the best results for BRAF status classification achieved, for each pre-training approach, using this CNN architecture as a feature extractor are analyzed in this section. More detailed results for the ResNet-18 can be found in appendix A, as well as the results obtained using the EfficientNet-B2 (appendix B) and the Inception-V3 (appendix C) architectures.

6.3.1 KNN

For the KNN classifier, experiments on the extracted features were conducted considering the euclidean distance (Euc.) and the cosine similarity (Cos.). The number of neighbors considered in these experiments was either 3 (3N) or 5 (5N). The results obtained for the per-image analysis are shown in figure 6.4, and for the SF analysis, in figure 6.5.

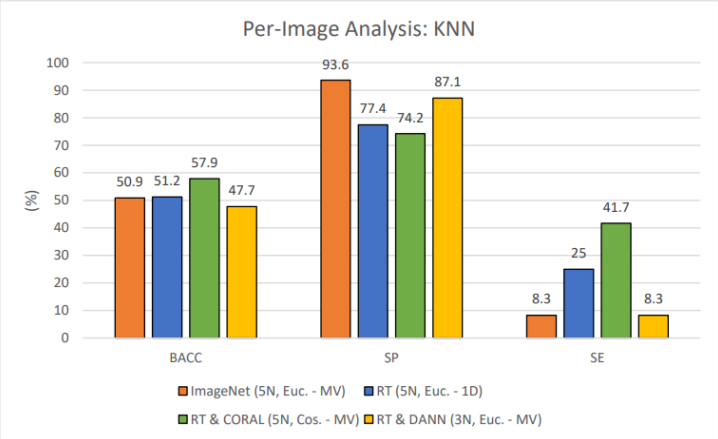


Figure 6.4: Per-Image analysis: **BACC**, **SP**, and **SE** values for the BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **KNN algorithm**.

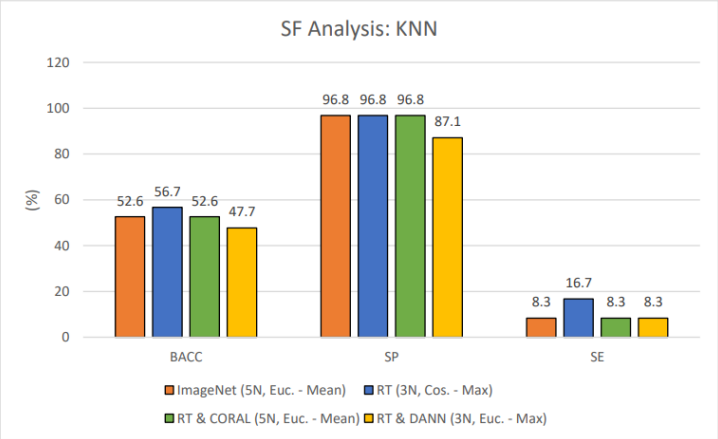


Figure 6.5: SF analysis: **BACC**, **SP**, and **SE** values for the BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **KNN algorithm**.

From figures 6.4 and 6.5, considering the BACC scores, one can conclude that the best result for the per-image analysis is attained when considering the RT & CORAL pre-training approach for the feature extractor, the KNN classifier configured with 5N, cosine similarity, and the MV criterion. The best SF analysis result is attained when considering the pre-training on the RT for the extractor, the KNN classifier configured with 3N, cosine similarity, and the Max criterion. Despite the best pre-training approach not being the same in the two types of analyses, the selected distance metric for the KNN is equal.

In general, the KNN classifier leads to biased results, exhibiting high SP and low SE values in both analyses.

6.3.2 Prototypes

For the prototype-based classifiers, the P1, P2, M1, and M2 classifiers introduced in chapter 4 were applied to the extracted feature vectors. The results obtained for the per-image analysis are presented in figure 6.6 and for the SF analysis in figure 6.7.

For the prototypes approach, it is evident that there is a preference for the MV criterion in the per-image analysis and a preference for the Max criterion in the SF analysis.

The RT pre-training put together with the P2 BRAF classifier achieves the highest BACC scores (62.1% for the per-image analysis and 60.5% for the SF analysis); besides, it displays balanced SE and SP values in both scenarios.

The RT & DANN pre-trained extractor also presents acceptable results in the two types of analyses despite being a slightly worse option when compared to the architecture initialized with the ImageNet weights for the SF analysis, which exhibits a better SP value. The RT & CORAL pre-trained ResNet-18 architecture exhibits the worse results for the prototype-based classifiers, manifesting a BACC value

lower than 50% in the per-image and the SF analyses.

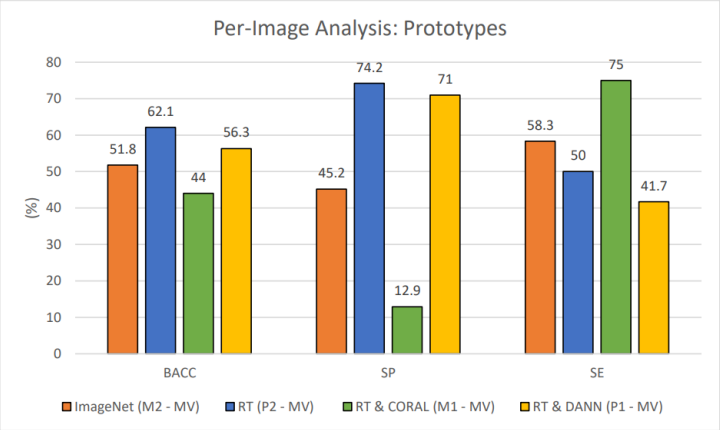


Figure 6.6: Per-image analysis: **BACC**, **SP**, and **SE** values for the BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **prototype-based algorithms**.

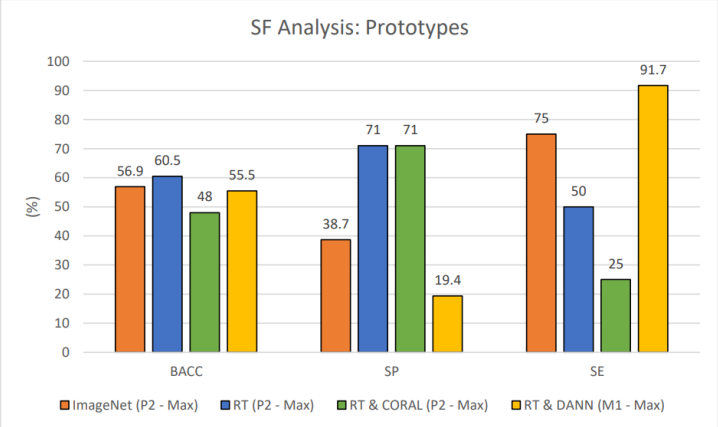


Figure 6.7: SF analysis: **BACC**, **SP**, and **SE** values for the BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **prototype-based algorithms**.

6.3.3 LR

The LR classifier was trained on the features extracted from the BRAF dataset. Three configurations for the LR were considered: LR with $L1$ penalty to compute sparse solutions when existent, reducing irrelevant coefficients to zero, i.e., removing unnecessary features for the BRAF classification, LR with $L2$ penalty, and classical LR with no regularization terms.

Figure 6.8 shows that there is an unanimous choice in terms of per-image analysis criterion since all the classifier configurations present the best results when MV is considered. As exhibited in figures 6.8 and 6.9, for all the pre-training approaches, the LR configuration for the per-image analysis coincides

with the configuration selected for the SF analysis. Moreover, for all the pre-training approaches, the regularized versions of the LR attained the best results.

The feature extractor pre-trained on the RT and the feature extractor pre-trained on the RT & CORAL exhibit the best results when the $L2$ penalty LR is considered. However, the networks pre-trained on the ImageNet dataset and on the RT & DANN approach attain the best results when the $L1$ penalty term is taken into account instead. This said, there is a high chance that the feature vectors extracted using these last two mentioned pre-training approaches contain information that is irrelevant for BRAF classification and when this information is discarded, the models perform well for this task. In fact, the RT & DANN pre-trained network used as feature extractor together with the LR classifier with the $L1$ penalty leads to a BACC of 57.9% on the per-image analysis and a BACC of 59.5% in the SF analysis, the best results achieved in these analyses with the LR classifier.

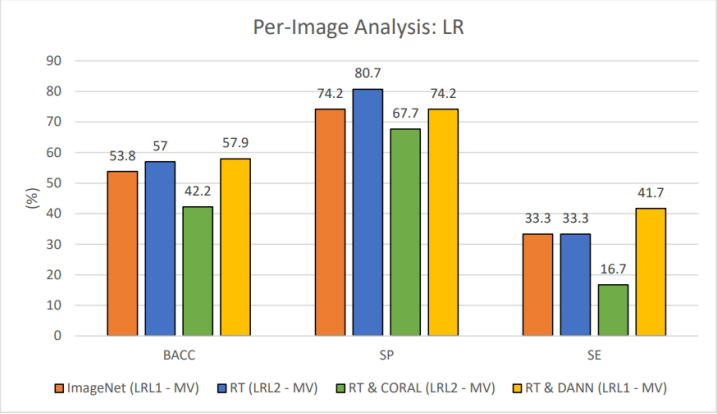


Figure 6.8: Per-image analysis: **BACC**, **SP**, and **SE** values for the BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **LR**.

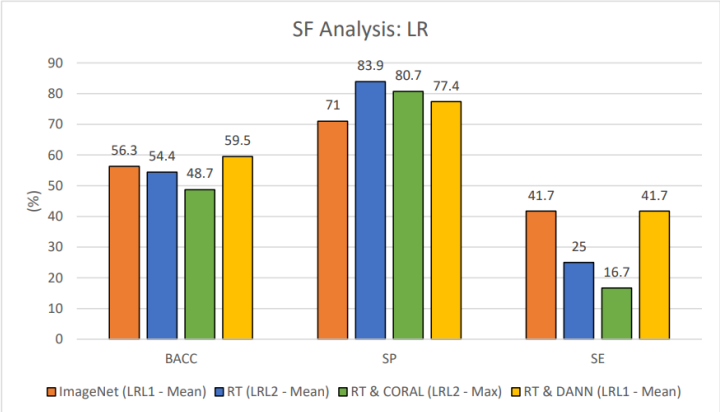


Figure 6.9: SF analysis: **BACC**, **SP**, and **SE** values for the BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **LR**.

6.3.4 Final Considerations on BRAF Classification

From the conducted experiments with the ResNet-18 as a feature extractor, it is observed that the KNN classifier is not the best option for classifying BRAF status. The BRAF dataset presents more BRAF- samples than BRAF+ samples, and the achieved results show that the KNN algorithm presents, in a general manner, tendency to classify the samples as BRAF-, being susceptible to imbalanced data. The results obtained exhibit high SP values and low SE values, making this classifier biased for all the pre-training approaches.

The prototype-based classifiers show promising results, especially the classifiers inspired by the ProtoNet (P1 and P2). Comparing the prototypes to the KNN and the LR classifiers, the prototype-based P2 classifier, together with the RT pre-training, achieved not only the most balanced results in terms of SE and SP but also the best results in terms of BACC for the per-image and the SF analyses. The LR classifier also achieved reasonable results, despite being less balanced in terms of SE and SP than the results obtained using the prototype-based classifiers.

Finally, it can be concluded that when a patient presents more than one image per lesion, it is possible to summarize the information in the multiple feature vectors and still achieve results similar to those obtained through the per-image analysis. In fact, the results achieved for the three families of classifiers show that in both types of analyses, there is a tendency for the best classifier configuration to be equal.

6.4 Comparison with the State of the Art

The authors of [7] built a tree classifier to predict BRAF status in melanoma given only two variables: patient age and the presence/absence of BWVs in the melanoma. Applying the predictor of [7] to the available BRAF dataset results in the decision tree of figure 6.10.

The performance of this classifier is summarized in table 6.4. In this table, the best results per pre-training approach for the ResNet-18 architecture are also presented, for the per-image analysis and for the SF analysis.

The classifier exhibited in figure 6.10 shows that the absence of BWVs is a good indicator for BRAF- melanomas. In fact, it discards 28 of the 43 patients. Of these 28 patients, around 71.4% manifest BRAF- melanoma. It can be seen in table 6.4 that this decision tree achieves a high SP value (77.4%). However, the SE value is low (around 16.7%). The tree classifier does not lead to the best results, the reason behind this may be related to the age distribution of the BRAF dataset (69.60 ± 11.45), which is different from the one of the dataset available in [7]. Besides, for the application of the decision tree classifier, the melanomas located in the palmoplantar and facial regions were not discarded, like in [7], since in this thesis, the BRAF dataset was already small.

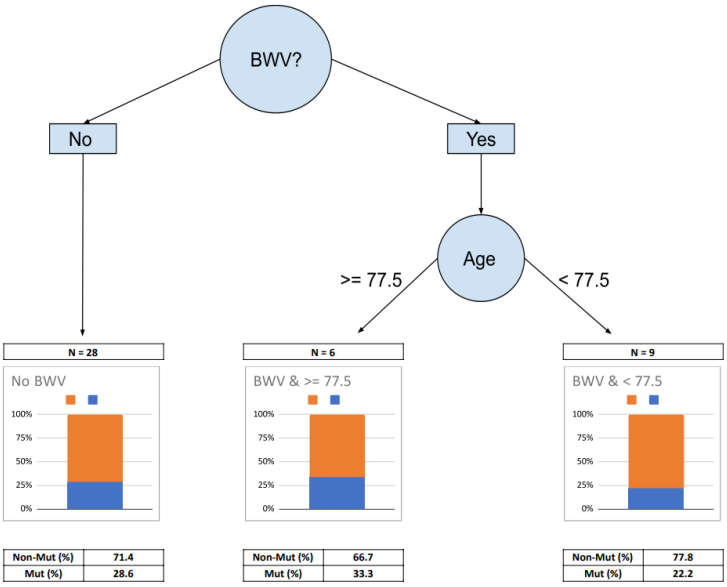


Figure 6.10: Decision tree classifier [7] (● : BRAF-, ● : BRAF+).

Table 6.4: State-of-the-art algorithm compared with the best results per pre-training approach using the **ResNet-18** as feature extractor for BRAF classification.

Analysis	**Pre-train**	**Braf Classif.**	Metrics				
			BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
Medical	-	Decision Tree [7]	47.1	77.4	16.7	19.1	22.2
Per-Image	ImageNet	LR_{L1} - MV	53.8	74.2	33.3	33.3	33.3
	RT	**P2 - MV**	**62.1**	**74.2**	**50.0**	**46.2**	**42.9**
	RT & CORAL	$KNN_{5N,Cos.}$ - MV	57.9	74.2	41.7	40.0	38.5
	RT & DANN	LR_{L1} - MV	57.9	74.2	41.7	40.0	38.5
SF	ImageNet	P2 - Max	56.9	38.7	75.0	45.0	32.1
	RT	**P2 - Max**	**60.5**	**71.0**	**50.0**	**44.4**	**40.0**
	RT & CORAL	$KNN_{5N,Euc.}$ - Mean	52.6	96.8	8.3	14.3	50.0
	RT & DANN	LR_{L1} - Mean	59.5	77.4	41.7	41.7	41.7

Comparing the best results obtained per pre-training approach in this investigation for the ResNet-18 architecture with the results obtained by employing the classifier developed in [7] (table 6.4), it can be concluded that the proposed approaches are viable, achieving better results than the ones obtained through the algorithm presented in the state of the art [7]. Moreover, the best results attained for BRAF status prediction in this thesis (recall table 6.1) indicate that the EfficientNet-B2 architecture pre-trained on the ImageNet dataset surpasses the state-of-the-art algorithm in every aspect.

7

Conclusions and Further Investigation

Contents

7.1 Conclusions

The findings of this study prove that resorting to DL methods to analyze dermoscopic data for BRAF status prediction can surpass the existent state-of-the-art results (algorithm proposed in [7]).

The leave-one-out cross-validation [56] proved to be a reliable approach in this study to evaluate the models, given the lack of data problem. Plus, the two types of analyses selected to deal with the multiple images per patient problem (the per-image and the SF analyses) tend to lead to the same behaviors for the ResNet-18 architecture, showing that it is possible to summarize the information for a patient presenting multiple data for the same lesion in a single information vector.

Unfortunately, the Inception-V3 and the EfficientNet-B2 architectures did not benefit from the proposed pre-training approaches for the BRAF status prediction task. So, the information learned by the feature extractor during pre-training is highly dependent on the selected architecture.

Regarding the classifiers adopted for this work, the prototype-based classifiers emerge as promising classification algorithms for their good performance on the BRAF classification task and simplicity.

7.2 Further Investigation

This thesis foments further research around BRAF status prediction using computer-aided methods. The results obtained for BRAF status prediction are promising. However, this work still has room for improvement. One limitation of the present work is that it does not explain the relation between the information present in the dermoscopic images and the BRAF status. Nevertheless, there are some ideas to study the aforementioned relation and to improve the work in this thesis:

- It would be interesting to visualize the activation maps of the CNNs to further understand what patterns they learn and why some architectures benefit from pre-training tasks closely related to the BRAF detection problem and some don't. Plus, it would also be interesting to correlate the patterns learned with the information retrieved by dermatologists;
- Since the *in vivo* data used for this thesis comes from multiple centers, there exist, as previously mentioned, intra-dataset domain shifts. Instead of just aligning the *ex vivo* data with the *in vivo* data, a total alignment between the different data centers of the ISIC 2020 dataset and the *ex vivo* data should be explored to check if there is any benefit following a total DA strategy;
- To increase the reliability of the proposed BRAF classification algorithms, new dermoscopic data should be used to test the algorithms. Other types of medical images like WSIs, which convey information about cellular morphology, could also be used to further study the proposed methodologies.

Bibliography

[1] L. E. Davis, S. C. Shalin, and A. J. Tackett, “Current state of melanoma diagnosis and treatment,” *Cancer Biology & Therapy*, vol. 20, no. 11, pp. 1366–1379, 2019.

[2] The American Cancer Society medical and editorial content team. Targeted Therapy Drugs for Melanoma Skin Cancer. Accessed 19-November-2021. [Online]. Available: https://www.cancer.org/cancer/melanoma-skin-cancer/treating/targeted-therapy.html

[3] I. Kozar, C. Margue, S. Rothengatter, C. Haan, and S. Kreis, “Many ways to resistance: How melanoma cells evade targeted therapies,” *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, vol. 1871, no. 2, pp. 313–322, 2019.

[4] Z. Eroglu and A. Ribas, “Combination therapy with braf and mek inhibitors for melanoma: latest evidence and place in therapy,” *Therapeutic advances in medical oncology*, vol. 8, no. 1, pp. 48–56, 2016.

[5] S. E. Fenton, J. A. Sosman, and S. Chandra, “B-Raf Mutated Melanoma,” in *Cutaneous Melanoma*, 2020, ch. 1.

[6] S. Kuntz, E. Krieghoff-Henning, J. N. Kather, T. Jutzi, J. Höhn, L. Kiehl, A. Hekler, E. Alwers, C. von Kalle, S. Fröhling, J. S. Utikal, H. Brenner, M. Hoffmeister, and T. J. Brinker, “Gastrointestinal cancer classification and prognostication from histology using deep learning: Systematic review,” *European Journal of Cancer*, vol. 155, pp. 200–215, 2021.

[7] M. Armengot-Carbó, E. Nagore, Z. García-Casado, and R. Botella-Estrada, “The association between dermoscopic features and BRAF mutational status in cutaneous melanoma: Significance of the blue-white veil,” *Journal of the American Academy of Dermatology*, vol. 78, no. 5, pp. 920–926, 2018.

[8] D. Carter. Skin cancer screening: 6 things your dermatologist wants you to know. Accessed 19-January-2022. [Online]. Available: https://www.mdanderson.org/cancerwise/skin-cancer-screening--5-things-your-dermatologist-wants-you-to-know.h00-159142878.html

[9] V. Rotemberg, N. Kurtansky, B. Betz-Stablein, L. Caffery, E. Chousakos, N. Codella, M. Combalia, S. Dusza, P. Guitera, D. Gutman, A. Halpern, B. Helba, H. Kittler, K. Kose, S. Langer, K. Lioprys, J. Malvehy, S. Musthaq, J. Nanda, O. Reiter, G. Shih, A. Stratigos, P. Tschandl, J. Weber, and P. Soyer, "A Patient-Centric Dataset of Images and Metadata for Identifying Melanomas Using Clinical Context," *Scientific Data*, vol. 8, no. 34, pp. 1–8, 2021.

[10] M. R. Verdelho, S. Gonçalves, L. Gonçalves, C. Costa, J. M. Lopes, M. M. M. Coelho, A. João, P. Soares, H. Pópulo, and C. Barata, "Predictive Biomarkers in Melanoma Detection of BRAF Mutation using Dermoscopy," in *Artificial Intelligence over Infrared Images for Medical Applications and Medical Image Assisted Biomarker Discovery*, 2022.

[11] G. Q. Phan, J. L. Messina, V. K. Sondak, and J. S. Zager, "Sentinel Lymph Node Biopsy for Melanoma: Indications and Rationale," *Cancer Control*, vol. 16, no. 3, pp. 234–239, 2009.

[12] S. N. Lo, J. Ma, R. A. Scolyer, L. E. Haydu, J. R. Stretch, R. P. M. Saw, O. E. Nieweg, K. F. Shannon, A. J. Spillane, S. Ch'ng, G. J. Mann, J. E. Gershenwald, J. F. Thompson, and A. H. R. Varey, "Improved Risk Prediction Calculator for Sentinel Node Positivity in Patients With Melanoma: The Melanoma Institute Australia Nomogram," *Journal of Clinical Oncology*, vol. 38, no. 24, pp. 2719–2727, 2020.

[13] S. Wong, M. Kattan, K. McMasters, and D. Coit, "A Nomogram That Predicts the Presence of Sentinel Node Metastasis in Melanoma With Better Discrimination Than the American Joint Committee on CancerStaging System," *Annals of surgical oncology*, vol. 12, no. 4, pp. 282–288, 2005.

[14] Z. Bursac, C. H. Gauss, D. K. Williams, and D. W. Hosmer, "Purposeful selection of variables in logistic regression," *Source Code for Biology and Medicine*, vol. 3, no. 17, pp. 1–8, 2008.

[15] T. J. Brinker, L. Kiehl, M. Schmitt, T. B. Jutzi, E. I. Krieghoff-Henning, D. Krahl, H. Kutzner, P. Gholam, S. Haferkamp, J. Klode, D. Schadendorf, A. Hekler, S. Fröhling, J. N. Kather, S. Haggenmüller, C. von Kalle, M. Heppt, F. Hilke, K. Ghoreschi, M. Tiemann, U. Wehkamp, A. Hauschild, M. Weichenthal, and J. S. Utikal, "Deep learning approach to predict sentinel lymph node status directly from routine histology of primary melanoma tumours," *European Journal of Cancer*, vol. 154, pp. 227–234, 2021.

[16] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5987–5995.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[18] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[19] S. Farahmand, A. Fernandez, F. Ahmed, D. Rimm, J. Chuang, E. Reisenbichler, and K. Zarringhalam, “Deep learning trained on hematoxylin and eosin tumor region of interest predicts HER2 status and trastuzumab treatment response in HER2+ breast cancer,” *Modern Pathology*, vol. 35, no. 1, pp. 44–51, 2022.

[20] G. Wilson and D. J. Cook, “A Survey of Unsupervised Deep Domain Adaptation,” *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 51, pp. 1–46, 2020.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[22] C. Bombonato, S. Ribero, F. C. Pozzobon, J. A. Puig-Butille, C. Badenas, C. Carrera, J. Malvehy, E. Moscarella, A. Lallas, S. Piana, S. Puig, G. Argenziano, and C. Longo, “Association between dermoscopic and reflectance confocal microscopy features of cutaneous melanoma with BRAF mutational status,” *Journal of the European Academy of Dermatology and Venereology*, vol. 31, no. 4, pp. 643–649, 2017.

[23] D. R. Sarvamangala and R. V. Kulkarni, “Convolutional neural networks in medical image understanding: a survey,” *Evolutionary Intelligence*, vol. 15, no. 1, pp. 1–22, 2022.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[25] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” 2015. [Online]. Available: https://arxiv.org/abs/1511.08458

[26] R. Yamashita, M. Nishio, R. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[27] Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, “Computer Vision for Human–Machine Interaction,” in *Computer Vision for Assistive Healthcare*, 2018, ch. 5, pp. 127–145.

[28] M. Yani, S. S. M. T. Budhi Irawan, and S. T. M. T. Casi Setiningsih, “Application of Transfer Learning using Convolutional Neural Network Method for Early Detection of Terry’s Nail,” in *Journal of Physics: Conference Series*, vol. 1201, no. 1, 2019, p. 012052.

[29] M. Lin, Q. Chen, and S. Yan, "Network In Network," 2013. [Online]. Available: https://arxiv.org/abs/1312.4400

[30] H. Kim and Y.-S. Jeong, "Sentiment Classification Using Convolutional Neural Networks," *Applied Sciences*, vol. 9, no. 11, p. 2347, 2019.

[31] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[32] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a Few Examples: A Survey on Few-Shot Learning," *ACM Comput. Surv.*, vol. 53, no. 63, pp. 1–34, 2020.

[33] E. G. Miller, N. E. Matsakis, and P. A. Viola, "Learning from One Example Through Shared Densities on Transforms," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, 2000, pp. 464–471.

[34] T. Pfister, J. Charles, and A. Zisserman, "Domain-Adaptive Discriminative One-Shot Learning of Gestures," in *Computer Vision – ECCV 2014*, 2014, pp. 814–829.

[35] W. Zi, L. S. Ghoraie, and S. Prince. Tutorial # 2: few-shot learning and meta-learning I. Accessed 19-January-2022. [Online]. Available: https://www.borealisai.com/en/blog/tutorial-2-few-shot-learning-and-meta-learning-i/

[36] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A Closer Look at Few-shot Classification," in *International Conference on Learning Representations*, 2019.

[37] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 3637–3645.

[38] J. Snell, K. Swersky, and R. Zemel, "Prototypical Networks for Few-shot Learning," in *Advances in Neural Information Processing Systems*, 2017.

[39] H. Guan and M. Liu, "Domain Adaptation for Medical Image Analysis: A Survey," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 3, pp. 1173–1185, 2021.

[40] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[41] H. Daumé and D. Marcu, "Domain Adaptation for Statistical Classifiers," *J. Artif. Int. Res.*, vol. 26, pp. 101–126, 2006.

[42] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer Joint Matching for Unsupervised Domain Adaptation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1410–1417.

[43] M. Wang, D. Zhang, J. Huang, P.-T. Yap, D. Shen, and M. Liu, “Identifying Autism Spectrum Disorder With Multi-Site fMRI via Low-Rank Domain Adaptation,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 3, pp. 644–655, 2020.

[44] M. Ghafoorian, A. Mehrtash, T. Kapur, N. Karssemeijer, E. Marchiori, M. Pesteie, C. R. G. Guttmann, F.-E. de Leeuw, C. M. Tempany, B. van Ginneken, A. Fedorov, P. Abolmaesumi, B. Platel, and W. M. Wells, “Transfer Learning for Domain Adaptation in MRI: Application in Brain Lesion Segmentation,” in *Medical Image Computing and Computer Assisted Intervention − MICCAI 2017*, 2017, pp. 516–524.

[45] B. Sun, J. Feng, and K. Saenko, “Return of Frustratingly Easy Domain Adaptation,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2058–2065.

[46] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-Adversarial Training of Neural Networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2030–2096, 2016.

[47] B. Sun and K. Saenko, “Deep CORAL: Correlation Alignment for Deep Domain Adaptation,” in *Computer Vision – ECCV 2016 Workshops*, 2016, pp. 443–450.

[48] T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997.

[49] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, “An Introduction to Logistic Regression Analysis and Reporting,” *The journal of educational research*, vol. 96, no. 1, pp. 3–14, 2002.

[50] C. Barata, M. E. Celebi, and J. S. Marques, “Improving Dermoscopy Image Classification Using Color Constancy,” *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 1146–1152, 2015.

[51] G. D. Finlayson and E. Trezzi, “Shades of Gray and Colour Constancy,” in *Proc. 12th Color Imag. Conf.: Color Sci. Eng. Syst., Technol., Appl.*, 2004, pp. 37–41.

[52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[53] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *International conference on machine learning*, 2019, pp. 6105–6114.

[54] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[55] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[56] "Leave-One-Out Cross-Validation," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., 2010, pp. 600–601.

[57] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," 2018. [Online]. Available: https://arxiv.org/abs/1802.03426

A

Detailed Results for the ResNet-18 Architecture

Table A.1: Best results per BRAF classifier using the **ResNet-18** as feature extractor on the BRAF dataset.

Per-Image Analysis							
Classifier	Configuration	Pre-Train	Metrics				
			BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
KNN	5N,Cos. - MV	RT & CORAL	57.9	74.2	41.7	40.0	38.5
Prototypes	**P1 or P2 - MV**	**RT**	**62.1**	**74.2**	**50.0**	**46.2**	**42.9**
LR	L1 Pen. - MV	RT & DANN	57.9	74.2	41.7	40.0	38.5
SF Analysis							
Classifier	Configuration	Pre-Train	Metrics				
			BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
KNN	3N,Cos. - Max	RT	56.7	96.8	16.7	26.7	66.7
Prototypes	**P2 - Max**	**RT**	**60.5**	**71.0**	**50.0**	**44.4**	**40.0**
LR	L1 Pen. - Mean	RT & DANN	59.5	77.4	41.7	41.7	41.7

Table A.2: BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **KNN algorithm**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
5 or 3 N,Euc. - MV *	ImageNet	50.9	93.6	8.3	13.3	33.3
5N,Euc. - 1D	RT	51.2	77.4	25.0	27.3	30.0
5N,Cos. - MV	**RT & CORAL**	**57.9**	**74.2**	**41.7**	**40.0**	**38.5**
3N,Euc. - MV	RT & DANN	47.7	87.1	8.3	11.8	20.0
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
3 or 5 N,Euc. or Cos. - Mean **	ImageNet	52.6	96.8	8.3	14.3	50.0
3N,Cos. - Max	**RT**	**56.7**	**96.8**	**16.7**	**26.7**	**66.7**
5N,Euc. - Mean	RT & CORAL	52.6	96.8	8.3	14.3	50.0
3N,Euc. - Max	RT & DANN	47.7	87.1	8.3	11.8	20.0

*An equal result is obtained when using a 5 neighbors KNN with cosine similarity and MV criterion on features extracted by the ResNet-18 with ImageNet initialization.
**An equal result is obtained when using a 5 neighbors KNN with cosine similarity and Max criterion on features extracted by the ResNet-18 with ImageNet initialization.

Table A.3: BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **Prototype-based algorithms**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
M2 - MV or 1D	ImageNet	51.8	45.2	58.3	38.9	29.2
P1 or P2 - MV	**RT**	**62.1**	**74.2**	**50.0**	**46.2**	**42.9**
M1 - MV or 1D	RT & CORAL	44.0	12.9	75.0	37.5	35.0
P1 - MV	RT & DANN	56.3	71.0	41.7	38.5	35.7
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
P2 or M1 - Max	ImageNet	56.9	38.7	75.0	45.0	32.1
P2 - Max	**RT**	**60.5**	**71.0**	**50.0**	**44.4**	**40.0**
P2 - Max	RT & CORAL	48.0	71.0	25.0	25.0	25.0
M1 - Max	RT & DANN	55.5	19.4	91.7	45.8	30.6

Table A.4: BRAF status classification using pre-trained **ResNet-18** architectures as feature extractors on the BRAF dataset and the **LR**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
L1 pen. - MV	ImageNet	53.8	74.2	33.3	33.3	33.3
L2 pen. - MV	RT	57.0	80.7	33.3	36.4	40.0
L2 pen. - MV	RT & CORAL	42.2	67.7	16.7	16.7	16.7
L1 pen. - MV	**RT & DANN**	**57.9**	**74.2**	**41.7**	**40.0**	**38.5**
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
L1 pen. - Mean	ImageNet	56.3	71.0	41.7	38.5	35.7
L2 pen. - Mean	RT	54.4	83.9	25.0	30.0	37.5
L2 pen. - Max *	RT & CORAL	48.7	80.7	16.7	20.0	25.0
L1 pen. - Mean	**RT & DANN**	**59.5**	**77.4**	**41.7**	**41.7**	**41.7**

*An equal result is obtained when using a LR with no penalty and the Max criterion on the features extracted by the ResNet-18 with RT & CORAL initialization.

B

Detailed Results for the EfficentNet-B2 Architecture

Table B.1: Best results per BRAF classifier using the **EfficientNet-B2** as feature extractor on the BRAF dataset.

Per-Image Analysis							
Classifier	Configuration	Pre-Train	Metrics				
			BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
KNN	5 or 3 N,Euc. - MV	ImageNet	51.9	87.1	16.7	22.2	33.3
Prototypes	M1 - MV	ImageNet	73.9	64.5	83.3	60.6	47.6
LR	**L1 Pen. - 1D**	**ImageNet**	**75.3**	**83.9**	**66.7**	**64.0**	**61.5**
SF Analysis							
Classifier	Configuration	Pre-Train	Metrics				
			BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
KNN	3N,Euc. - Mean	ImageNet	54.2	100.0	8.3	15.4	100.0
Prototypes	M1 - Mean	ImageNet	65.6	64.5	66.7	51.6	42.1
LR	**L1 Pen. - Mean or Max**	**ImageNet**	**77.8**	**80.7**	**75.0**	**66.7**	**60.0**

Table B.2: BRAF status classification using pre-trained **EfficientNet-B2** architectures as feature extractors on the BRAF dataset and the **KNN algorithm**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
5 or 3 N,Euc. - MV	**ImageNet**	**51.9**	**87.1**	**16.7**	**22.2**	**33.3**
3N,Euc. - MV	RT	49.3	90.3	8.3	12.5	25.0
5N,Cos. - MV	RT & CORAL	50.3	83.9	16.7	21.1	28.6
3N,Euc. - 1D	RT & DANN	42.9	77.4	8.3	10.0	12.5
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
3N,Euc. - Mean	**ImageNet**	**54.2**	**100.0**	**8.3**	**15.4**	**100.0**
5N,Cos. - Mean	RT	50.9	93.6	8.3	13.3	33.3
5N,Cos. - Max	RT & CORAL	49.3	90.3	8.3	12.5	25.0
5N,Euc. - Mean	RT & DANN	48.4	96.8	0.0	0.0	0.0

Table B.3: BRAF status classification using pre-trained **EfficientNet-B2** architectures as feature extractors on the BRAF dataset and the **Prototype-based algorithms**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
M1 - MV	**ImageNet**	**73.9**	**64.5**	**83.3**	**60.6**	**47.6**
P2 - MV	RT	49.6	74.2	25.0	26.1	27.3
M1 - 1D	RT & CORAL	54.3	41.9	66.7	42.1	30.8
P1 - MV	RT & DANN	49.9	58.1	41.7	33.3	27.8
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
M1 - Mean	**ImageNet**	**65.6**	**64.5**	**66.7**	**51.6**	**42.1**
M2 - Mean	RT	58.2	58.1	58.3	43.8	35.0
P1 - Max	RT & CORAL	60.5	71.0	50.0	44.4	40.0
P2 - Max	RT & DANN	46.4	67.7	25.0	24.0	23.1

Table B.4: BRAF status classification using pre-trained **EfficientNet-B2** architectures as feature extractors on the BRAF dataset and the **LR**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
L1 pen. - 1D	**ImageNet**	**75.3**	**83.9**	**66.7**	**64.0**	**61.5**
L2 pen. - 1D *	RT	54.4	83.9	25.0	30.0	37.5
L1 pen. - 1D	RT & CORAL	71.4	67.7	75.0	58.1	47.4
L1 pen. - 1D	RT & DANN	49.9	58.1	41.7	33.3	27.8
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
L1 pen. - Max **	**ImageNet**	**77.8**	**80.7**	**75.0**	**66.7**	**60.0**
L1 pen. - Max	RT	66.3	74.2	58.3	51.9	46.7
L1 pen. - Mean	RT & CORAL	66.3	74.2	58.3	51.9	46.7
L1 pen. - Max	RT & DANN	53.1	64.5	41.7	35.7	31.3

*An equal result can be obtained by considering the MV criterion instead of 1D.
**An equal result can be obtained by considering the Mean criterion instead of Max.

C

Detailed Results for the Inception-V3 Architecture

Table C.1: Inception-V3 performance on the RT. For the *in vivo* performance, the networks were evaluated on the **ISIC 2020 validation set** ($6,196$ images: $6,079$ benign, 117 melanomas), whereas for the *ex vivo* performance, the networks were evaluated on the **private *ex vivo* dataset** (138 images: 69 benign, 69 melanomas).

***In Vivo* Performance**					
Pre-Train	Metrics				
	BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
RT	**75.2**	**88.0**	**62.4**	**16.0**	**9.1**
RT & CORAL	67.4	95.5	39.3	21.0	14.3
RT & DANN	63.4	96.0	30.8	18.0	12.8
***Ex Vivo* Performance**					
Pre-Train	Metrics				
	BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
RT	49.3	13.04	85.5	63.0	49.6
RT & CORAL	**55.1**	**20.3**	**89.9**	**67.0**	**53.0**
RT & DANN	51.4	97.1	5.8	11.0	66.7

Table C.2: Best results per BRAF classifier using the **Inception-V3** as feature extractor on the BRAF dataset.

Per-Image Analysis							
Classifier	Configuration	Pre-Train	Metrics				
			BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
KNN	5N,Euc. - MV	ImageNet	51.9	87.1	16.7	22.2	33.3
Prototypes	P1 or P2 - 1D	ImageNet	63.0	67.7	58.3	48.3	41.2
LR	**L1 Pen. - 1D**	**ImageNet**	**64.7**	**71.0**	**58.3**	**50.0**	**43.8**
SF Analysis							
Classifier	Configuration	Pre-Train	Metrics				
			BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
KNN	5N,Cos. - Max	ImageNet	53.5	90.3	16.7	23.5	40.0
Prototypes	P2 - Mean	ImageNet	54.7	67.7	41.7	37.0	33.3
LR	**L1 Pen. - Max**	**ImageNet**	**72.7**	**87.1**	**58.3**	**60.9**	**63.6**

Table C.3: BRAF status classification using pre-trained **Inception-V3** architectures as feature extractors on the BRAF dataset and the **KNN algorithm**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
5N,Euc. - MV	**ImageNet**	**51.9**	**87.1**	**16.7**	**22.2**	**33.3**
5N,Cos. - MV	RT	42.9	77.4	8.3	10.0	12.5
5N,Cos. - 1D	RT & CORAL	48.7	80.7	16.7	20.0	25.0
5N,Euc. - MV	RT & DANN	46.1	83.9	8.3	11.1	16.7
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
5N,Cos. - Max	**ImageNet**	**53.5**	**90.3**	**16.7**	**23.5**	**40.0**
5 or 3 N,Euc. - Max	RT	50.9	93.6	8.3	13.3	33.3
5N,Cos. - Max	RT & CORAL	48.4	96.8	0.0	0.0	0.0
5N,Euc. - Mean	RT & DANN	50.9	93.6	8.3	13.3	33.3

Table C.4: BRAF status classification using pre-trained **Inception-V3** architectures as feature extractors on the BRAF dataset and the **Prototype-based algorithms**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
P1 or P2 - 1D	**ImageNet**	**63.0**	**67.7**	**58.3**	**48.3**	**41.2**
M1 - 1D	RT	49.3	90.3	8.3	12.5	25.0
M2 - MV or 1D	RT & CORAL	53.0	22.6	83.3	43.5	29.4
P2 - MV	RT & DANN	50.5	67.7	33.3	30.8	28.6
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
P2 - Mean	**ImageNet**	**54.7**	**67.7**	**41.7**	**37.0**	**33.3**
P1 - Mean	RT	41.5	58.1	25.0	21.4	18.8
M2 - Mean	RT & CORAL	53.0	22.6	83.3	43.5	29.4
M2 - Mean *	**RT & DANN**	**61.7**	**48.4**	**75.0**	**48.6**	**36.0**

*This result is probably an outlier as the results obtained with the other prototypes hardly achieve BACC values of 50%

Table C.5: BRAF status classification using pre-trained **Inception-V3** architectures as feature extractors on the BRAF dataset and the **LR**.

Per-Image Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
L1 pen. - 1D	**ImageNet**	**64.7**	**71.0**	**58.3**	**50.0**	**43.8**
L1 pen. - MV	RT	42.2	67.7	16.7	16.7	16.7
L1 pen. - MV	RT & CORAL	43.2	61.3	25.0	22.2	20.0
No pen. - MV	RT & DANN	41.9	83.9	0.0	0.0	0.0
SF Analysis						
Configuration	Pre-Train	Metrics				
		BACC(%)	SP(%)	SE(%)	F1(%)	PR(%)
L1 pen. - Max	**ImageNet**	**72.7**	**87.1**	**58.3**	**60.9**	**63.6**
L1 pen. - Mean or Max	RT	42.2	67.7	16.7	16.7	16.7
L1 pen. - Mean	RT & CORAL	52.2	71.0	33.3	32.0	30.8
No pen. - Max	RT & DANN	46.8	93.6	0.0	0.0	0.0