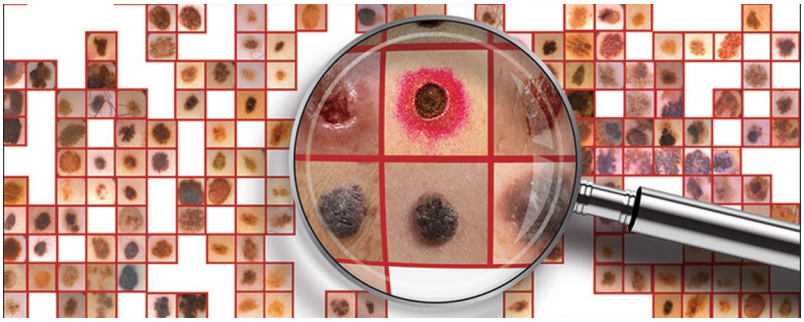
TÉCNICO LISBOA



Explainable Artificial Intelligence for Skin Cancer Detection: A Prototype-Based Deep Learning Architecture with Non-Expert Supervision

Miguel Joaquim Nobre Correia

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Carlos Jorge Andrade Mariz Santiago
Prof. Ana Catarina Fidalgo Barata

Examination Committee

Chairperson: Prof. Pedro Filipe Zeferino Aidos Tomás
Supervisor: Prof. Carlos Jorge Andrade Mariz Santiago
Member of the Committee: Prof. Mário Alexandre Teles de Figueiredo

November 2023

Declaration

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

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Acknowledgments

Wow... I can't believe I'm finally here. It all happened so quickly, but I've made it - completing my master's degree and embarking on a new chapter of my life. It has been many years of study, dedication, and effort to reach this goal, but I didn't do it alone.

First and foremost, I want to express my gratitude to my incredible twin brother Ricardo, with whom I had the pleasure of sharing all my school years, and even the same course and university. Undoubtedly, he has been a fundamental support for me, and I wouldn't have made it without him. Thank you for always being there to help me and making challenging times more joyful and fun, always seeing the optimistic side of things. Thank you for being the best brother anyone could have and my best friend.

Secondly, I want to thank my parents, Agostinho and Lurdes, for all the effort, hard work, dedication, and sacrifices they made so that my brother and I could study. Thank you for the love you've always given me and for always being ready to support me. Everything I have achieved would not have been possible without you. Thank you for instilling in me the good values and education I have today. You are the best parents I could wish for. Thirdly, I want to express my gratitude to Carmo, who has always been present in our family as long as I can remember, and has been like a second mother to me and my brother. Thank you for all the support and encouragement you've provided us with. All that I have achieved would not have been possible without these four pillars in my life, shaping me into who I am today. I would also like to thank my grandparents, the rest of my family, and friends for their unwavering support throughout.

Furthermore, I am immensely grateful to my thesis supervisors, Professor Carlos Santiago and Professor Catarina Barata. They have been exemplary teachers, offering immense help, cooperation, and guidance during my work. I deeply appreciate the knowledge they shared with me and for opening my eyes to the scientific development and research in the area I worked on. Undoubtedly, they are incredible and extraordinary teachers, and I will always be sincerely grateful for being part of my academic and professional experience.

Lastly, I want to extend special thanks to André, Diogo, Tiago, Rita, and Alceu, with whom I had the opportunity to share my entire thesis journey and who have been a tremendous help and great friends.

To each and every one of you, thank you so much for helping me close a fundamental chapter of my

life, allowing another one to begin.

Abstract

This thesis proposes an innovative approach to skin cancer detection using an interpretable prototypical part model. Our method addresses the limitations of existing black-box models in terms of interpretability. We apply this approach to two classification tasks: distinguishing between melanoma and nevus lesions, and classifying eight distinct skin lesion types. By analyzing prototypical parts that resemble diagnostic image features, our model provides explanations that directly impact its decision-making process. To enhance the clinical relevance of our approach, we incorporate non-expert supervision to guide the selection of relevant prototype areas within the lesion, excluding confounding factors beyond its boundaries. This supervision can be achieved using: 1) binary masks, obtained automatically using a segmentation network and 2) user-refined prototypes. We explore two different ways to integrate the first supervision, including modifying the loss function or integrating it directly into the model's forward process. In the binary scenario, we explore a novel approach: having prototypes only for the malignant class. This streamlines explanations by concentrating on comparisons with these prototypes. We further enhance this approach by introducing a new loss component to boost diversity among prototypes within the same class. Our findings reveal that our interpretable model, though initially constrained compared to black-box models on validation sets, narrows this gap in test set generalization. Particularly, interpretable scenarios using prototype-level supervision surpass black-box models in both approaches on test sets, highlighting the role of interpretability in enhancing model generalization.

Keywords

Skin Cancer; Prototypes; Human Feedback; Interpretability.

Resumo

Esta tese propõe uma abordagem inovadora para a deteção de cancro da pele através de um modelo interpretável baseado em protótipos de partes. O método aborda as limitações dos modelos caixa-preta em termos de interpretabilidade. Aplicamos esta abordagem a duas tarefas de classificação: distinguir entre lesões de melanoma e nevus e classificar oito tipos distintos de lesões da pele. Ao analisar protótipos de partes que se assemelham a características de diagnóstico da imagem, o modelo fornece explicações que afetam diretamente o seu processo de tomada de decisão. Para melhorar a relevância clínica, incorporamos supervisão não especializada para orientar a seleção de áreas de protótipos relevantes dentro da lesão, excluindo fatores de confusão para além dos seus limites. Isto pode ser alcançado usando: 1) máscaras binárias, obtidas automaticamente utilizando uma rede de segmentação, e 2) protótipos refinados pelo utilizador. Exploramos duas maneiras de integrar a primeira supervisão, incluindo a modificação da função de perda ou a sua integração direta no processo de avanço do modelo. No cenário binário, exploramos uma nova abordagem: ter protótipos apenas para a classe maligna, simplificando as explicações em comparações com apenas esses protótipos. Aprimoramos esta abordagem introduzindo uma nova componente de perda para aumentar a diversidade intra-classe entre protótipos. Os resultados mostram que o modelo interpretável, apesar de inicialmente ter desempenho inferior aos modelos de caixa-preta nos conjuntos de validação, destaca-se na generalização nos conjuntos de teste, especialmente com supervisão ao nível dos protótipos. Isto ressalta a importância da interpretabilidade na melhoria da generalização do modelo.

Palavras Chave

Cancro da Pele; Protótipos; Feedback Humano; Interpretabilidade.

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
AMs	Activation Maps
BA	Balanced Accuracy
CAM	Class activation mapping
CAV	Concept Activation Vector
CAVs	Concept Activation Vectors
CNN	Convolutional Neural Network
CNNs	Convolutional Neural Networks
CrsEnt	cross entropy loss
DL	Deep Learning
Grad-CAM	Gradient-weighted Class Activation Mapping
IAIA-BL	Interpretable AI Algorithm for Breast Lesions
KDL	Knowledge Distillation Loss
MEL	Melanoma
NNs	Neural Networks
NV	Nevus
ProtoPNet	Prototypical Part Network
SGD	Stochastic Gradient Descent
TCAV	Testing with Concept Activation Vectors
XAI	eXplainable Artificial Intelligence

1

Introduction

Contents

1.1 Motivation

Skin cancer is a significant global concern, with a staggering number of 150,000 new melanoma cases diagnosed in 2020 alone [17]. In fact, even a single instance of sunburn every two years can triple the chances of developing melanoma skin cancer when compared to individuals who have never experienced sunburn [18].

The early detection of skin cancer is vital for effective treatment and enhancing patients' quality of life. To aid in the diagnosis of skin cancer, dermoscopy has emerged as a valuable non-invasive medical imaging technique that provides high-resolution images of skin lesions [19], see fig. 1.1. However, despite its usefulness, accurately diagnosing melanoma remains a challenge. This challenge has prompted dermatologists to seek assistance from computer systems during the initial diagnosis process.

In recent years, there have been notable advancements in computer-assisted systems for skin cancer diagnosis [20]. However, this task remains complex due to the similarities between malignant and non-malignant lesions. While there is little variation between classes, there exists significant variability within the same class in terms of lesion size, shape, color, and texture [21]. Moreover, dermoscopy images often contain artifacts such as hairs, rulers, and veins, which can obstruct relevant features and be misinterpreted by computer-aided systems during the detection process [22].

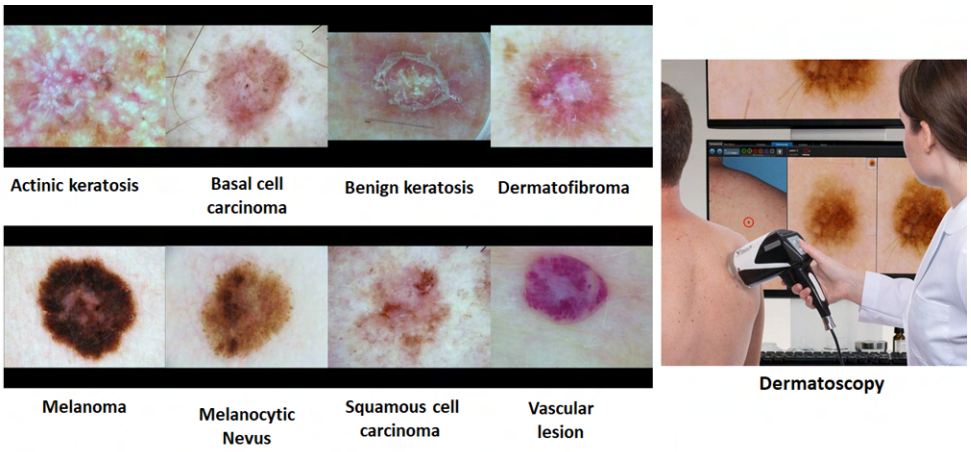


Figure 1.1: Illustration showcasing various types of skin lesions from the ISIC 2019 dataset [1–3] on the left, along with an exemplification of the dermatoscopy process on the right.

Furthermore, researchers have increasingly gravitated towards the utilization of Neural Networks (NNs) in the realm of automatic skin cancer diagnosis [23]. The growing popularity of NNs can be attributed to their remarkable capacity to tackle intricate problems by adeptly identifying pertinent features and establishing meaningful relationships within the data. However, it is important to note that the NNs used in these tasks often take the form of multi-layered deep neural networks. While these models achieve impressive results, they can become complex and difficult to interpret, even with meticulous

analysis. In essence, they are often viewed as black boxes, leaving users, including doctors, without a clear understanding of the decision-making process behind the model's output [24]. This lack of transparency can be concerning, particularly in high-risk situations, as the responsibility for decision-making is delegated to a non-human entity.

Moreover, in the context of algorithms relying on human data to make decisions, it is of utmost importance to take into account the right to explanation as outlined in Europe's General Data Protection Regulation [25]. Recent emphasis has been placed on the necessity to employ eXplainable Artificial Intelligence (XAI) techniques and address concerns related to the use of Deep Learning (DL) models characterized by their black-box nature [26]. There exists a crucial query that demands an answer from medical practitioners: **why did the model arrive at this particular decision?**

1.2 Problem Formulation

XAI techniques have been developed to address the lack of transparency in black-box models. However, many existing methods of explanation follow a post-hoc approach, to analyze the decision-making process of the model after it has been trained. This approach can lead to explanations that do not accurately capture the internal computations of the model, lacking in detail or coherence [27]. An alternative solution to this problem is the development of interpretable models that inherently possess transparency. Interpretable models offer the advantage of enabling a comprehensive understanding of the internal path within the model, while providing explanations that are easy to comprehend and exerting influence on the decision-making process [27].

A prime example of an interpretable model is the Prototypical Part Network (ProtoPNet) [10]. This architecture introduces a method for understanding the process behind image classification by comparing different parts of an image with prototypical parts that are characteristic of specific classes. The comparison between image parts and prototypical parts produces similarity scores, and the final model decision is determined by these scores. This classification approach leads to a transparent, natural, intuitive, and human-like explanation inherent to the model, accurately reflecting its internal computational process.

However, in its classical form, issues arise regarding the assurance of prototype quality and diversity. The prototypes learned by the model, which serve as the basis for explanations, can sometimes be representative of confounding factors, biases, and artifacts present in the images, rather than capturing the most relevant characteristics of the classes they represent. This implies that both the decision-making process and the corresponding explanation, relying solely on these prototypes, may be compromised by factors that, for instance, hold less clinical significance in the context of skin cancer diagnosis. Consequently, it becomes essential to employ techniques that oversee the quality of the prototypes, ensuring

they potentially represent features of greater clinical relevance for informed decision-making. In fig. 1.2, the application of the ProtoPNet framework can be observed within the context of melanoma skin lesion classification.

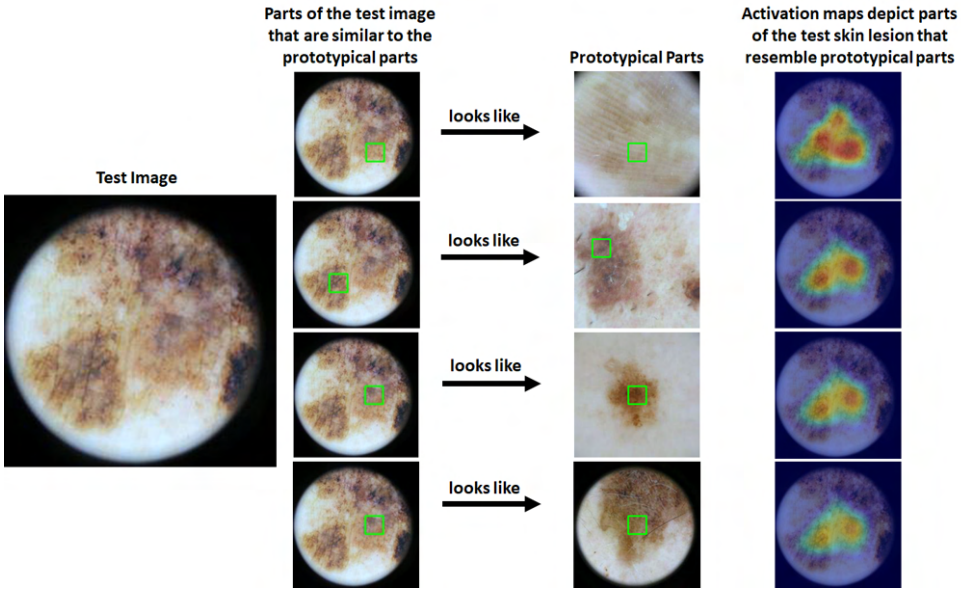


Figure 1.2: Example of a correctly classified melanoma image, when compared to prototypes of similar skin lesion parts from the same class.

1.3 Objectives and Thesis Contributions

This thesis aims to build upon the foundation of ProtoPNet [10] and apply it to a binary problem of classifying skin lesions as melanoma or nevus, as well as extend it to a multi-class problem encompassing eight types of skin lesions, as depicted in fig. 1.1. Additionally, in the binary problem, it is also explored the approach of using only prototypes for the malignant class and making the decision entirely dependent on the comparison with these prototypes, rather than being influenced by prototypes from the other class as well. The aim is to make the explanation even more straightforward and concise.

However, the primary focus of this thesis is to incorporate supervision techniques that contribute to improving the quality of the learned prototypes. These prototypes should represent parts within the skin lesion and its boundary that are potentially of greater clinical relevance, as opposed to potential artifacts located outside the lesion's boundary. Two of the techniques are based on segmentation masks, drawing inspiration from the fine annotation loss present in current literature [11]. Another technique, which can be used alternatively, draws from the remembering loss of ProtoPDebug [13]. This technique provides an opportunity for a human user to input examples of prototypes that they consider more valid and diagnostically relevant.

Furthermore, an analysis was performed using a dataset with annotations from 5 experts regarding a total of 31 dermatological features. This analysis aimed to understand if the prototypes learned by the model, using one of the supervision techniques that ensures the prototypes are within the boundary of the lesion, can represent any concept belonging to the medical lexicon of dermatology.

Finally, this thesis demonstrates the ability to have an interpretable model built on the foundation of prototypes, which potentially holds greater clinical relevance. Moreover, it's worth highlighting that despite facing challenges in competing with black-box models on the validation set, which shared a similar distribution to the training set, interpretable models with non-expert supervision at the prototype level exhibited remarkable generalization capabilities when tested on test sets associated with different hospital domains and distributions. This underscores their superior performance in real-world scenarios, where their adaptability truly shines.

In conclusion, a portion of the work conducted for this thesis has contributed to the acceptance of an article at the Workshop on Interpretability of Machine Intelligence in Medical Image Computing at MICCAI 2023, titled "XAI for Skin Cancer Detection with Prototypes and Non-Expert Supervision" [28]. Additionally, a second paper titled "Global and Local Explanations for Skin Cancer Diagnosis Using Prototypes" was accepted for presentation at the Eighth ISIC Skin Image Analysis Workshop at MICCAI 2023 [29]. This paper is also related to the themes and issues addressed in this thesis.

1.4 Document Organization

This document is structured into 6 chapters. Chapter 1 introduces the detection of skin cancer using computational systems and emphasizes the need for these systems to be transparent and reliable, instilling confidence in medical professionals. This requirement leads to the adoption of XAI methods, particularly interpretable models like ProtoPNet, which form the foundation of this thesis.

In chapter 2, we provide an explanation of XAI and its fundamental concepts. We then present various XAI techniques used in medical image analysis, focusing on the distinction between post-hoc techniques and the need for interpretable models. ProtoPNet is described in detail, along with examples of its application in medical image analysis. We discuss the improvements introduced and the challenges encountered, including its application to skin cancer detection, and how our approach differs from the current state-of-the-art.

Chapter 3 further explains the adopted approaches and methods in detail. In chapter 4, we present the datasets used, the experimental setup, and the evaluation metrics employed. Finally, chapter 5 showcases the results obtained, discussing the various prototype supervision techniques utilized in both binary and multiclass classification contexts. Ultimately, chapter 6 presents the conclusions drawn from this research and outlines potential avenues for future work.

2

State of the Art Review

Contents

2.1 eXplainable Artificial Intelligence (XAI)

The purpose of this section is to introduce the concept of XAI and provide the reader with a basic understanding of the different techniques that exist and examples of their application for medical image analysis. Furthermore, it is intended to explain how the approach discussed in this thesis, which is based on using the deep network architecture ProtoPNet [10], fits within XAI and how it differs from some of the other methods discussed.

2.1.1 Important XAI Concepts

XAI is a type of artificial intelligence that provides humans with an explanation or justification for the decisions made by the model. It is often used to try to make the decision-making processes of black-box models more transparent, as these models can be difficult for users to understand. As such, XAI is closely related to three key concepts: transparency, interpretability, and explainability.

A model is transparent when the method by which it obtains characteristics from the training data and assigns labels to the test data can be described by the approach itself [30]. Interpretability refers to the ability of humans to understand the path behind the decisions made by an Artificial Intelligence (AI) system [31]. Explainability refers to the ability of an AI system to provide a clear and understandable explanation of its decision-making process to humans.

Although explanation and interpretability may be considered synonymous, for some authors there is a distinction. Namely, an uninterpretable explanation may not correctly represent the model's path, pointing only to what data the model pays attention to and not how it uses it [27]. An interpretable model differs from an explainable model in that it provides a clear and reliable explanation of its internal calculations. This characteristic allows us to understand the logic behind the model's decisions. Additionally, if the model's explanation exhibits human-like behavior, it can be more intuitive and easier for people to understand. As a result, an interpretable model can present itself as a human-friendly explanation that is influential in decision-making [27].

2.2 XAI Techniques In Medical Image Analysis

To distinguish the different techniques used in XAI for medical image analysis, we followed the nomenclature adopted by van der Velden *et al.* [32], which is based on the work of Adadi and Berrada [33] and Murdoch *et al.* [31].

The methods of explanation for medical image analysis can be divided fundamentally into 3 main categories: visual explanation, textual explanation and example-based explanation [32]. As such, this section will present at least one example of each type of XAI. In particular, three types of visual explana-

tion are first presented: Gradient-weighted Class Activation Mapping (Grad-CAM) [34], Class activation mapping (CAM) [35] and prediction difference analysis [36]. The first two being backpropagation-based approaches and the latter being a perturbation-based approach, a distinction discussed later. Second, a textual explanation is presented: Testing with Concept Activation Vectors (TCAV) [7]. Finally, some introduction is given to ProtoPNet, as a representative of an example-based approach, as this is covered in more detail in section 2.3.

Furthermore, we can distinguish between XAI models whose explanation is inherent to the model (model-based explanation) and models whose explanation results from an analysis made after they have been trained (post hoc explanation). In other words, we can say that an explanation based on the model forces NNs to be interpretable, while a post hoc explanation is created after the model has been trained. Additionally, we can distinguish between an explanation that is specific to the model (model-specific explanation) and an explanation that is agnostic or independent of the model (model-agnostic explanation). Finally, we can distinguish between a local explanation and a global explanation. The first corresponds to an explanation for a given model output, and the second takes into account the entire model, often called an explanation at the dataset level.

2.2.1 Visual XAI

Visual XAI is undoubtedly the most commonly used type of explanation, subdividing into two types of approaches: backpropagation-based approaches and perturbation-based approaches [32]. In short, visual explanations present saliency maps that indicate the most relevant parts of the image for the decision made by the model.

Starting with backpropagation-based approaches, these were one of the initial approaches to introduce explanation in medical image analysis and in other areas, since they make it possible to visualize what Convolutional Neural Networks (CNNs)-based models have learned. Backpropagation is based on the calculation of the gradient of the classification result obtained for a given class with respect to the input image, allowing two visualization techniques. The first in which one generates an image that maximizes the output score for a given class and a second in which a saliency map is generated for a specific input image with respect to a given class [37]. Saliency maps basically identify the regions of the input image that have a higher weight, at the level of the pixels, for this image to be classified as belonging to a particular class [37]. These maps can only be created on CNNs that have been trained with image tags to solve a classification task. These approaches are post hoc explanations.

In the case of perturbation-based approaches, in order to understand which regions of the image are more relevant to the model decision, perturbations or changes are made to the input image. Interestingly, perturbation-based approaches are all post hoc, model-agnostic and local explanations. Consequently, although they bring the great advantage that they can be used in several types of models, the explanation

is obtained after training the model. Therefore, they do not represent a very transparent explanation in the sense that they do not translate the true internal path of the model. However, perturbation-based approaches still bring advantages in terms of their simple utility in not having to restructure a model in order to make it explainable, and as such there is no risk of having to sacrifice the model's predictive ability.

2.2.1.A Grad-CAM and CAM

Grad-CAM [34] is a method for generating visual explanations from CNN-based networks without requiring any architectural modifications or retraining. It is an extension of CAM [35] and uses gradients from the model output to identify the image regions that contribute the most to the output for a particular class. Grad-CAM can be applied to a wide range of CNNs, including those with fully-connected layers, structured outputs, multi-modal inputs, or reinforcement learning [34].

It is important to note that CAM is a technique for identifying the regions of an image that are most relevant for distinguishing between different classes. It does this by mapping the score for a particular class to the feature maps of the previous convolutional layer using global average pooling. The resulting weighted sum represents the regions of the image that contribute most to the output for that class. CAM allows to detect objects without the need to have additional information such as bounding boxes. However, CAM is restricted for CNNs with max pooling with a fully-convolutional architecture and that do not have any fully-connected layers [35]. CAM was used for example to show how using additional information beyond the dermoscopic image of the skin lesion, such as the age and location of the skin lesion on the body, allows the model to focus on the most important regions of the skin for disease detection [4], as can be seen in fig. 2.1.

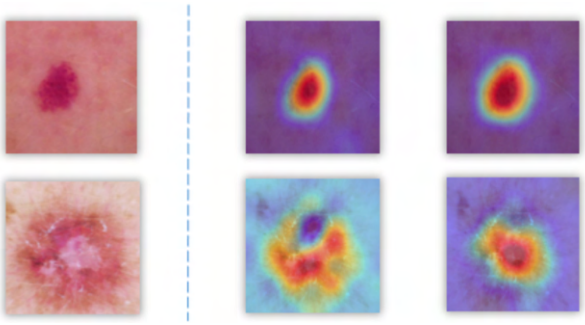


Figure 2.1: Example of CAM that allowed understanding that metadata, such as lesion location and age in addition to the skin lesion image allowed the model to focus on more important skin regions. Input images (left), Activation Maps (AMs) obtained without the use of metadata (middle) and AMs obtained when metadata was used (right) [4].

Grad-CAM was used to compare the performance of common methods used for melanoma classification to the performance of a model called MelaNet [5]. The results showed that the common methods

sometimes made incorrect predictions, even when the region of the skin lesion was highly activated. In contrast, for the wrong classifications made by the MelaNet model, the most activated regions were those surrounding the lesion, as shown in fig. 2.2. Therefore, Grad-CAM illustrates which regions of the image the model is focusing on when making a decision, but it does not provide any information about how the model is using that information internally.

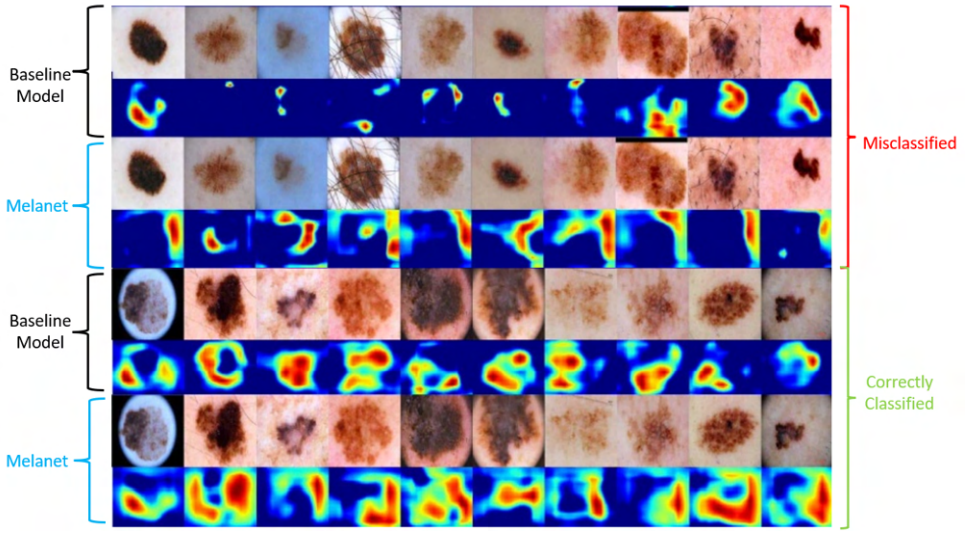


Figure 2.2: Grad-CAM maps for the correctly classified malignant cases by MelaNet and a baseline model, in the lower half of the figure and misclassified malignant cases in the top half of the picture [5].

2.2.1.B Prediction Difference Analysis

Zintgraf *et al.* [36] presents prediction difference analysis as a local visual explanation method at the pixel level. Each pixel is associated to a relevance score that is estimated by measuring the change in prediction when the feature, in this case the pixel, is omitted.

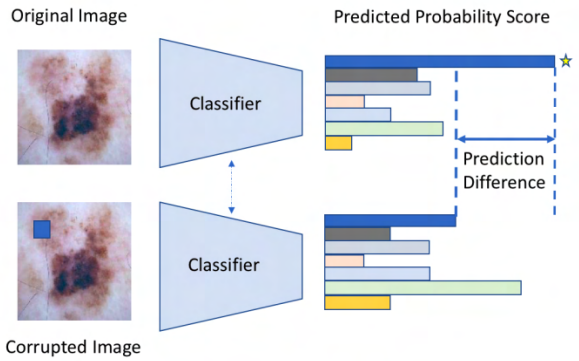


Figure 2.3: Example of the use of prediction difference analysis in the classification of a skin lesion by introducing a perturbation, in this case a blue square. Notice how the introduction of the perturbation caused the class with the highest probability to no longer be the one represented by the blue color, but the one represented by the green color [6].

The authors found that, while requiring more resources and time, this method provides clearer explanations for model decisions in the medical field, including distinguishing HIV patients from healthy patients using MRI analysis. For example, Li *et al.* [6] visualize how the model responds to a specific corrupted input image in order to explain a local decision, which allows the identification of key features without retraining the model. This example is illustrated in fig. 2.3. In this case, the key features correspond to the biomarkers present in the region of the image referring to a skin lesion, and through prediction difference analysis it is possible to understand the importance of these features in the model decision process.

2.2.2 Textual XAI

Continuing to follow the work exposed by van der Velden *et al.* [32] one of the most used explanation categories in medical image analysis is textual explanation. As the name suggests, it involves the use of text to present and justify the decision made by a given model. This text can correspond to keywords, concepts or even a real report. In this section, TCAV is presented as a representative of this type of XAI.

2.2.2.A Testing with Concept Activation Vectors (TCAV)

Focusing on the analysis of TCAV [7], this is a type of explanation used in medical image analysis that is very interesting, because it provides a justification for the model decision based on user-defined concepts. It uses bidirectional derivatives to quantify how much a concept is present in the model and the sensitivity of the image or class relative to that concept.

Additionally, it is important to mention that this type of explanation is post hoc, model-agnostic, global (at the class level) and a local explanation. Although the explanation is not inherent to the model and does not have a direct consequence on the model decision, it brings the advantage over the visual approaches of being a very friendly explanation, since one can understand the model decision by checking which concepts are present in the image. Basically, following fig. 2.4 the user defines a concept and provides positive examples, P_C, and negative examples, N_C, to analyze its presence in the model and images. These examples can be external to the training set. Given a space of activations of a layer l of the model, TCAV estimates a vector that corresponds to the latent representation of that concept.

In order to find this vector it is necessary to define a linear classifier that distinguishes between the two sets P_C and N_C in the space of activations. Given that the model input is denoted by $x \in \mathbb{R}^n$ and that the activation function of layer l is represented by $f_l : \mathbb{R}^n \rightarrow \mathbb{R}^m$, the Concept Activation Vector (CAV), $v_C^l \in \mathbb{R}^m$, is defined as the vector orthogonal to the decision boundary that points toward the concept C. Through the use of bidirectional derivatives we can calculate the sensitivity of the model with respect to a concept C in a given layer l, for this we consider a given input example x belonging to class k. Thus the sensitivity of class k to concept C can be calculated as the bidirectional derivative

$S_{C,k,l}(x) = \nabla h_{l,k}(f_l(x)) \cdot v_c^l$, with $h_{l,k} : \mathbb{R}^m \to \mathbb{R}$, that is $h_{l,k}(f_l(x))$ corresponds to the score that the model assigns to x as belonging to class k.

TCAV uses these bidirectional derivatives to calculate the sensitivity of the model with respect to a class. The TCAV score can be defined as $TCAV_{Q_{C,k,l}} = |\{x \in X_k : S_{C,k,l}(x) > 0\}|/|X_k|$. Where $TCAV_{Q_{C,k,l}} \in [0, 1]$ corresponds to the fraction of elements in the input set X_k belonging to class k that are sensitive to the concept C. In fig. 2.4 it is possible to see how the concept *stripes* is defined, and how the sensitivity of the *zebra* class with respect to this same concept is calculated.

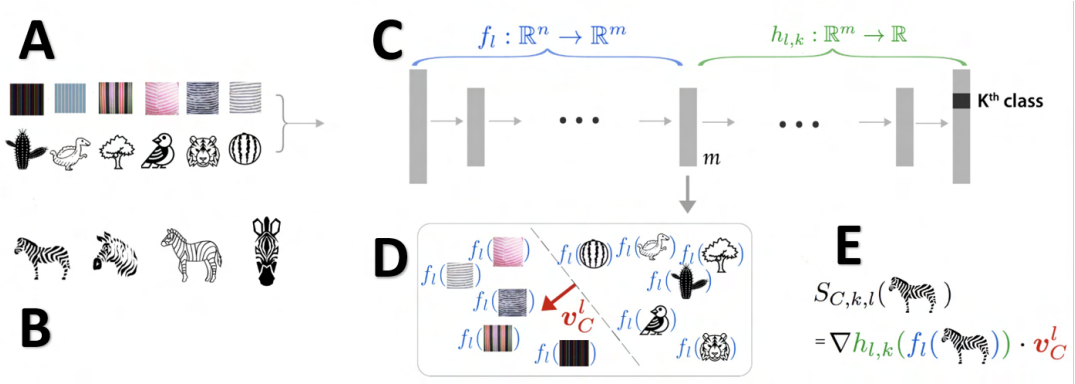


Figure 2.4: TCAV measures a model's sensitivity to a specific concept (e.g., 'striped') for a given class (zebras). It uses learned Concept Activation Vectors (CAVs) obtained by training a linear classifier to distinguish concept examples from examples in any layer. The CAV represents the orthogonal vector to the classification boundary. TCAV quantifies conceptual sensitivity for the target class using the directional derivative $S_{C,k,l}(x)$ [7].

TCAV was used to understand the sensitivity of a skin lesion classification model to concepts adopted by dermatologists to identify skin diseases like melanoma [8]. The Derm7pt dataset [9] was used to train the linear classifiers that distinguish between concepts like pigment network, streaks and blue-whitish veils, since in addition to the images being classified in terms of diagnosis they are divided into different classes of concepts, see fig. 2.5.

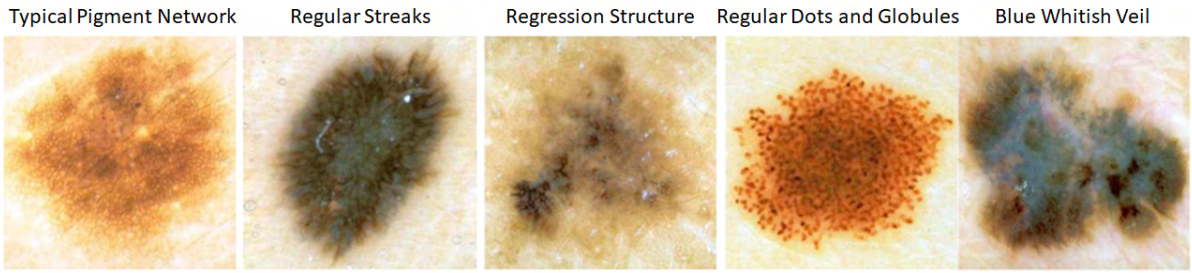


Figure 2.5: Examples of dermatological concepts of skin lesions, recognized in the medical community and present in the Derm7pt dataset [8, 9].

The defined concept classes are as follows: pigment network (PN), pigment network typical (PN_T), pigment network atypical (PN_AT), streaks (ST), streaks regular (ST_R), streaks irregular (ST_IR), re-

gression structures (RS), dots & globules (DG), regular dots & globules (DG_R), irregular dots & globules (DG_IR), blue-whitish veils (BWV), symmetry (Sym), asymmetry 1-axis (Asym_1) and asymmetry 2-axis (Asym_2) and colours (C_3). Figure 2.6 shows the scores obtained on a particular model layer for the derm7pt dataset. Analyzing the figure it is possible to conclude that the Nevus (NV) class is positively sensitive to benign lesion concepts, such as PN_T, ST_R and DG_R but is not sensitive to concepts related to malignant lesion concepts such as PN_AT, ST_IR, RS, DG_IR and BWV while the Melanoma (MEL) class presents the opposite behavior.

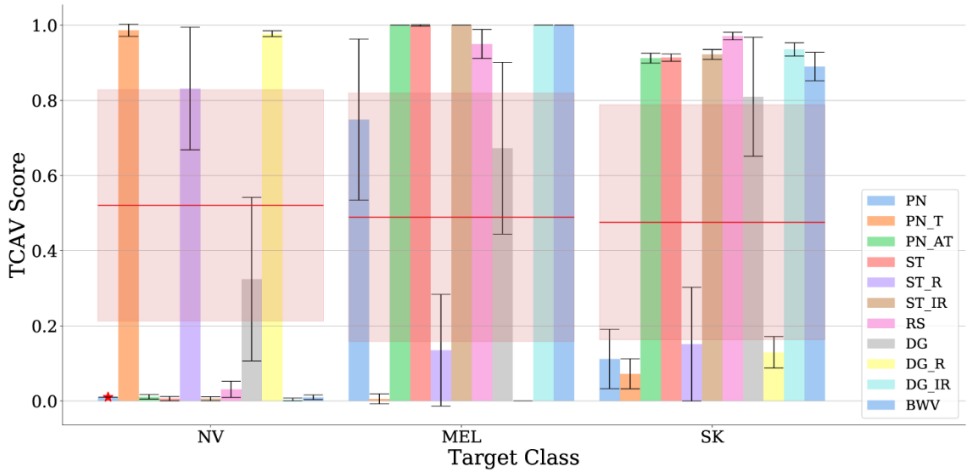


Figure 2.6: Example of the use of TCAV as a textual technique of XAI to understand the sensitivity of diagnostic classes like MEL and NV to concepts adopted by dermatologists for skin cancer detection [8].

These results are very important, since the behavior of the model, seems to be in line with what is expected from the medical point of view. For a more dense and detailed analysis, it is advisable to read the original article [8] in order to understand in detail this experiment and its conclusions, since here it is only shown in a very brief way the utility of TCAV as a technique of XAI applied in a skin lesion classification model.

2.2.3 Example-based XAI

To finish the exposition of the types of XAI used in medical image analysis, example-based explanations are presented [32]. This method explains the decision-making process of a DL model by providing specific examples or cases. The approach relies on similarity between cases, calculated using the model's representation space. By showing examples similar to the analyzed sample, the model's decision can be justified and potential biases can be identified.

A specific example of this approach is ProtoPNet introduced by Chen *et al* [10]. It analyzes parts of an image and compares them with prototypes, which are regions associated with a class. Unlike

example-based methods that compare the entire representation of an input image, ProtoPNet focuses on comparing parts with prototypical parts. The similarity score between the image parts and prototypes influences the model's final decision. ProtoPNet provides transparent, model-specific, and model-based local explanations, resembling how humans think about object detection by identifying fundamental object parts. It offers an intuitive and human-friendly explanation and is of interest for potential use in skin cancer detection. More details and examples of its applications are explored in section 2.3.

2.2.4 Conclusions

After examining the various examples of XAI presented, it is essential to draw conclusions and compare them with ProtoPNet, which serves as the foundation of this thesis.

First, by comparing specific image parts to prototypes ProtoPNet provides a more effective and detailed explanation of deep neural network decision-making. Focusing on these parts allows for the identification of crucial features influencing the model's output. Conversely, comparing the whole image to full-sized prototypes may lack granularity and make it harder to pinpoint essential features for the model's prediction.

Second, concerning the other approaches, Grad-CAM and CAM are post hoc techniques that provide model-specific, local explanations for CNN-based models. They show which parts of an image the model is using to make a decision, but do not reveal the internal path of the model. The manner in which the information is utilized within the model is not clearly specified. TCAV has the advantage of providing an intuitive, human-friendly explanation at the class level, using concepts that may be familiar to physicians. However, like Grad-CAM and CAM, it does not provide transparency into the internal process of the model. Perturbation-based approaches allow researchers to see how changes to the input affect the model's decision, but do not reveal the decision path or provide transparency into the internal workings of the model. They do have the advantage of not requiring modifications to the model itself, like TCAV, which allows the model's accuracy to be maintained. However, they still lack interpretability.

Finally, ProtoPNet distinguishes itself from other approaches by providing an explicit decision path that includes a inherent relationship between the explanation and the decision made. Additionally, although ProtoPNet provides an explanation that is inherent to the model, it also has a certain relationship with post hoc explainability, namely with visualization techniques such as CAM [35]. ProtoPNet provides not only example-based explanations, but also visual information through the identification of image regions that are most similar to prototypical parts. These prototypical parts can themselves represent medical concepts, such as specific structures of skin lesions.

ProtoPNet enables observing the model's attention to image parts and similarity to prototypes for decision-making. It is an interpretable model, providing natural, intuitive, and human-friendly explanations that mimic human reasoning.

2.3 Prototypical Part Network (ProtoPNet)

ProtoPNet is a DL architecture introduced by Chen *et al.* [10]. This architecture introduces a way for humans to understand the process behind image classification by comparing the parts of an image with prototypical parts that are characteristic of certain classes. The comparison between image parts and prototypical parts results in similarity scores, and the weighted average of these scores defines the final model decision. As previously stated, this pathway to the decision results in a transparent, natural, intuitive, and human-like explanation inherent to the model, which mirrors its internal computation process. Throughout this section, the architecture of ProtoPNet as well as the training process are described. Moreover, published work on the application of ProtoPNet as an XAI for medical image analysis is presented.

2.3.1 ProtoPNet architecture

The architecture of the ProtoPNet [10] can be seen in fig. 2.7. It comprises a Convolutional Neural Network (CNN), f, with the parameters w_{conv}, a prototype layer g_p and a fully connected layer h without bias and with parameters, w_h.

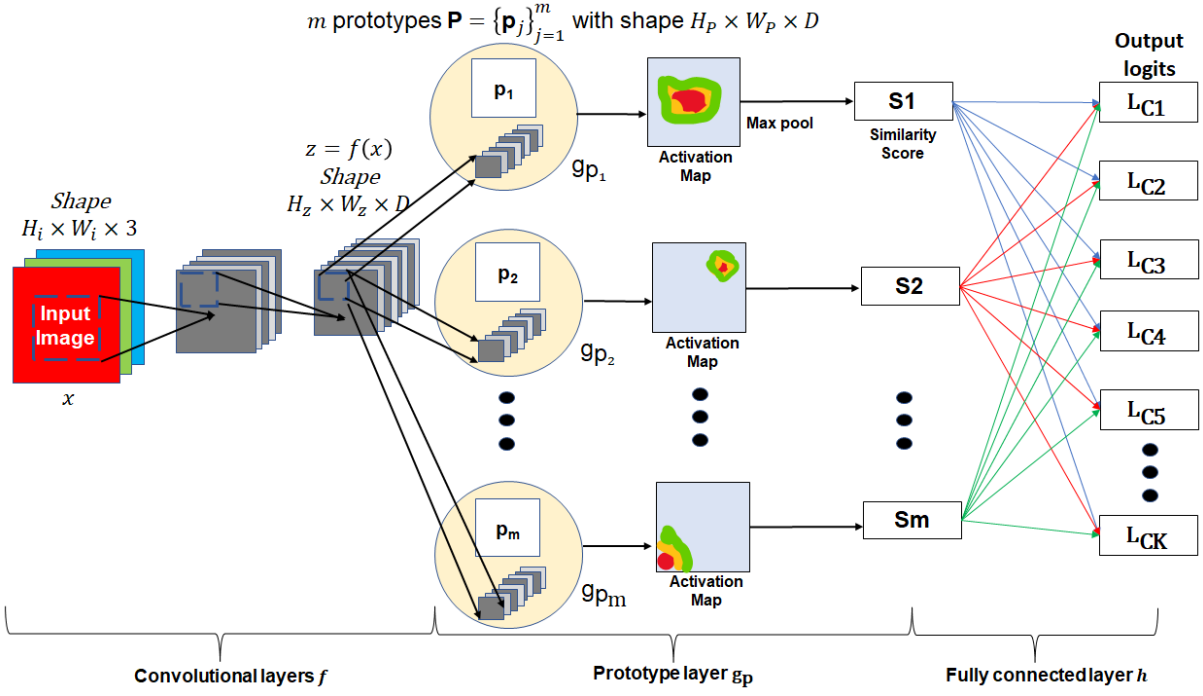


Figure 2.7: ProtoPNet architecture [10].

Given an input image x, of dimension $H_i \times W_i \times 3$, the CNN gives as output $z = f(x)$ with dimension $H_z \times W_z \times D$. The prototype layer g_p contains m prototype units $\{g_{p_j}\}_{j=1}^{m}$, where each unit computes

the squared L_2 distances between the prototype p_j and all the patches of z and converts the distances to similarity scores. Notice that ProtoPNet learns m prototypes $P = \{p_j\}_{j=1}^{m}$ and each prototype has the shape $H_P \times W_P \times D$ with $H_P \leq H_z$ and $W_P \leq W_z$.

Each prototype unit g_{p_j} computes the similarity between prototype p_j and the various regions of the input image x, creating an activation map. Then the activation map of similarity scores between the prototypical part and the image parts in each unit is reduced, using global max pooling, to a single value that defines the affinity between the prototypical part p_j and the image x, i.e. how present the prototypical part p_j is in the image. The output of each unit g_{p_j} is calculated using

$$g_{p_j}(z) = \underset{\tilde{z} \in \text{patches}(z)}{max} \log \frac{||\tilde{z} - p_j||_2^2 + 1}{||\tilde{z} - p_j||_2^2 + \epsilon} \quad , \tag{2.1}$$

where ϵ represents a small number to prevent division by zero. The higher the similarity score at the output of the unit g_{p_j} the smaller the distance L_2 between a given image patch and p_j, in the latent space where z is represented. Notice how the function is decreasing with respect to $||\tilde{z} - p_j||_2^2$.

The number of classes K must be predefined by the user, as well as the number of prototypes to represent each class k. The number of prototypes of each class $k \in \{1...K\}$ is denoted by m_k, with $P_k \subseteq P$ being the subset of prototypes assigned to class k, and these m_k prototypes should represent the most important parts that define class k in the classification task.

In the last part of ProtoPNet, in the fully connected layer h, the m similarity scores obtained are multiplied by the weight matrix w_h, with dimension $K \times m$, to obtain the score of the input image belonging to a given class, normalized using the softmax function. It is important to notice that the probability of the input image x to belong to a given class k is defined by the similarity scores obtained in the prototype layer. Thus, the higher the similarity with the prototypes of class k the higher the probability of belonging to that class.

2.3.2 Training of ProtoPNet

The training of ProtoPNet is divided into three phases: Stochastic Gradient Descent (SGD) of layers before the final layer, projection of prototypical parts and the convex optimization of the final layer [10].

In the first phase, the goal is to obtain a suitable representation of the latent feature space. Namely, it is intended that convolution features representing a certain class are clustered around prototypical parts of the respective class and are far away from prototypical parts of other classes. In this state, a joint optimization is performed using SGD, of the parameters w_{conv} and the prototypical parts $P = \{p_j\}_{j=1}^{m}$ of the prototype layer g_p. Given the training image set $D = [X, Y] = \{(x_i, y_i)\}_{i=1}^{n}$, the optimization function

of the problem to be solved is

$$\min_{P, w_{conv}} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(h \circ g_p \circ f(x_i), y_i) + \lambda_1 L_{\text{Clst}} + \lambda_2 L_{\text{Sep}} \quad , \tag{2.2}$$

where the operator $\circ$, represents the composition of the layers. The cross entropy loss (CrsEnt) penalizes the misclassification of the different classes in the training set. The minimization of the cluster loss

$$L_{\text{Clst}} = \frac{1}{n} \sum_{i=1}^{n} \min_{j: p_j \in P_{y_i}} \min_{z \in \text{patches}(f(x_i))} ||z - p_j||_2^2 \quad , \tag{2.3}$$

promotes that each training image has a part, represented in latent space, that is clustered near a prototypical part of the respective class to which the image belongs. On the other hand, the separation loss

$$L_{\text{Sep}} = -\frac{1}{n} \sum_{i=1}^{n} \min_{j: p_j \notin P_{y_i}} \min_{z \in \text{patches}(f(x_i))} ||z - p_j||_2^2 \quad , \tag{2.4}$$

encourages all parts of the training images to be distant from prototypical parts belonging to classes other than that of the respective training image. Furthermore, it is important to note that L_{Sep} leads to prototypical parts being representative of properties of a given class and distinct from the properties that define the other classes.

During this part, the parameters w_h are held fixed. The parameters are initialized with $w_h^{(k,j)} = 1$ for all j where $p_j \in P_k$ and $w_h^{(k,j)} = -0.5$ for all j where $p_j \notin P_k$. The positive values establish the connection between the output of the prototype units of a given class k and the respective score of the input image belonging to that class. Negative values establish the connection between the output of prototype units not belonging to a given class k and the respective score relative to the input image belonging to that class. Fixing the last layer in this way ensures that the model learns a representation in which parts of the training image, that are clustered near prototypical parts of a given class, increases the probability of the image belonging to that class and decreases the probability of belonging to another class.

In the second part of the training, the projection of prototypical parts is performed, i.e. each p_j is projected onto the closest latent part of a training image that belongs to the same class as p_j. This allows the user to visualize each prototype, or prototypical part, learned by the model as a patch or part of a training image. The projection of the prototypes corresponds to the following: for each prototype $p_j \in P_k$ the projection is given by

$$p_j \leftarrow \arg\min_{z \in Z_j} ||z - p_j||_2 \text{ with } Z_j = \{\tilde{z} : \tilde{z} \in \text{patches}(f(x_i)) \; \forall i \text{ s.t } y_i = k\} \, . \tag{2.5}$$

In the last part of the training, a convex optimization of the last layer is performed, i.e. of the

parameters w_h. It is given by

$$\min_{w_h} \frac{1}{n}\sum_{i=1}^{n} \text{CrsEnt}(h \circ g_p \circ f(x_i), y_i) + \lambda \sum_{k=1}^{K} \sum_{j:p_j \notin P_k} |w_h^{(k,j)}| \quad , \tag{2.6}$$

where the parameters of the prototype layer and w_{conv} are fixed. This allows the entries of the matrix w_h, which had been initialized with $w_h^{(k,j)} = -0.5$ for all j where $p_j \notin P_k$, now take a value $w_h^{(k,j)} \approx 0$. These values that are imposed to zero allow the model to not make a decision in a negative way. For example, the image is classified as belonging to class A because it is not from class B.

2.3.3 Explanation provided by the ProtoPNet

In fig. 2.8 [10], we can see how the ProtoPNet, can be used in the classification task of bird species images, how the internal path of the model works, and consequently the explanation. ProtoPNet compares each part of the input image, represented in latent space, with the prototypical parts of all classes to make a decision. This comparison, allows the creation of activation maps, one for each prototype, of similarity scores between the image parts and each prototypical part. This activation map is adjusted to the original dimension of the input image and placed over it to understand which part of the input image activates a given prototypical part the most, i.e. with which prototypical part it is most similar.

Figure 2.8: Internal path of ProtoPNet and explanation provided by it for bird species classification [10].

In fig. 2.8 [10], we can see how given the original input image the model finds a greater similarity between the bird's head present in the image and the prototypical part of a red-bellied woodpecker's head than the prototypical part of the red-cockaded woodpecker's head, the same happens analogously for the wing. Additionally, the model's final score for classifying an input image as belonging to a particular

class is determined by taking the weighted average of the similarity scores between the image and the prototypical parts associated with that class. Finally, it is important to note that the input image was correctly classified by the ProtoPNet.

2.3.4 ProtoPNet in Medical Image Analysis

This section highlights how ProtoPNet can be used for medical image analysis and what new developments have been introduced to this model to improve the explanation provided by it, as well as its performance.

2.3.4.A Interpretable AI algorithm for Breast Lesions (IAIA-BL)

One work that improves upon ProtoPNet is that of Barnett *et al.* [11]. In this paper, an interpretable model for the classification of mass lesions in digital mammography is presented, and is called Interpretable AI Algorithm for Breast Lesions (IAIA-BL). This model provides a local explanation that describes the internal reasoning for the decision process and can be easily understood by a physician. The decision is made by comparing the test images with prototype images, and showing which parts of the test image are most similar with the prototypes. In other words, it finds in the test mammogram the relevant region associated with a particular medical feature and uses this to decide whether the mass under analysis is malignant or benign, as illustrated in the fig. 2.9.

Additionally, it introduces a new aspect to the training process by adding detailed region annotations along with the images as well as a loss, referred to by the authors as a fine-annotation loss [11]. This loss allows IAIA-BL to excel at ensuring that the prototypes it learns are medically adequate by preventing them from having confounding factors, a particularly important problem in medical image analysis [38].

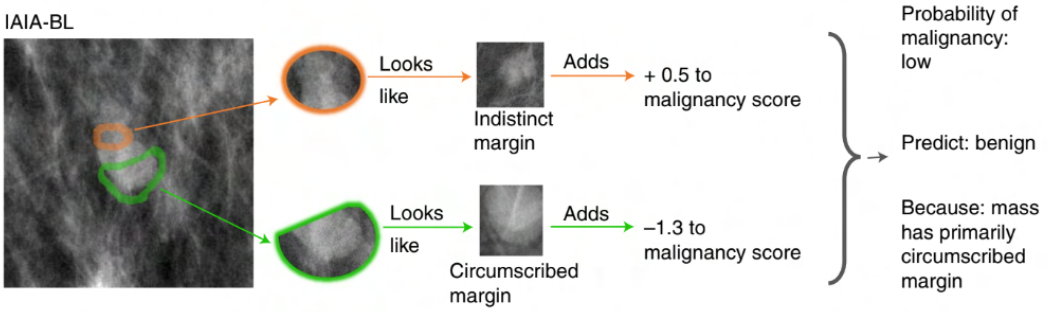


Figure 2.9: Exemplification of the explanation provided by IAIA-BL. Relevant areas are localized and using only the explained evidence, the prediction is based on a specific medical feature [11].

To train the IAIA-BL model, both data with and without detailed region annotations is used, and there is a penalty if a prototype is activated by an area of the image that is not suitable given the annotations provided in the training process. Following the authors' notation [11], there is a training subset D'

that contains the images with the true labels as well as the detailed annotations corresponding to the important medical information. For each training image $x_i \in D'$ a mask m_i is defined that takes the value 0 for the important pixels associated with the region marked as relevant through the annotations and 1 on the non-relevant pixels.

In the first training phase, the $\lambda_f L_{\text{Fine}}$ term is added to the optimization function shown in (2.2), where L_{Fine} represents the fine annotation loss [11], which is defined as

$$L_{\text{Fine}} = \sum_{i \in D'} \left(\sum_{j:\text{class}(p_j)=y_i} ||m_i \odot \text{PAM}_{i,j}||_2 + \sum_{j:\text{class}(p_j)\neq y_i} ||\text{PAM}_{i,j}||_2 \right) \quad . \tag{2.7}$$

The fine-annotation loss, allows the learned prototypes to take into account the relevant medical information present in the medical notes and to be representative of each class and distinct from the others. $\text{PAM}_{i,j}$ corresponds to the prototype activation map p_j referring to the training image x_i and over-sampled to the original dimension of the training image and the detailed annotation mask m_i.

The CrsEnt still refers to the function that penalizes miss classification in the training data, but in this case it is the types of mass lesions in digital mammography. However the functions L_{Clst}

$$L_{\text{Clst}} = \frac{1}{n} \sum_{i=1}^{n} \min_{j:\text{class}(p_j)=y_i} \frac{1}{\kappa} \sum \min_{z \in \text{patches}(f(x_i))} ||z - p_j||_2 \quad , \tag{2.8}$$

and L_{Sep}

$$L_{\text{Sep}} = -\frac{1}{n} \sum_{i=1}^{n} \min_{j:\text{class}(p_j)\neq y_i} \frac{1}{\kappa} \sum \min_{z \in \text{patches}(f(x_i))} ||z - p_j||_2 \quad , \tag{2.9}$$

are slightly different. In the sense that they introduce a relaxation where mink is used instead of min, i.e. the κ smallest distances L_2 instead of just the smallest, which contributes together with top-k average pooling instead of global max pooling to improve model performance. Additionally, a small detail is that the distances are not squared. This is because, contrary to what is presented in the article [11], in the publicly available code[1] the distances are not squared. Consequently, contrary to what is depicted in (2.1), in this case, the distances are also not squared in the expression.

2.3.4.B BRAIxProtoPNet++

Another work to use ProtoPNet is that of Wang *et al.* [12]. As it was mentioned, interpretable models have the great advantage of justifying their decision, but they usually have lower accuracy when compared with black-box models in the same classification tasks.

In the article by Wang *et al.* [12], this problem is addressed by proposing a model called BRAIxProtoPNet++, which consists of adding an ProtoPNet to a black-box model, in order to add interpretation to

[1] https://github.com/alinajadebarnett/iaiabl

an accurate mammography image classification model. This is achieved by using knowledge distillation from the global model to the ProtoPNet. Another novelty is introduced by ensuring that the prototypes learned by ProtoPNet, present diversity, i.e. each one is associated with a different training image. The results obtained with BRAIxProtoPNet++ [12], show that it allows to obtain higher classification accuracy than the black-box and prototype-based models.

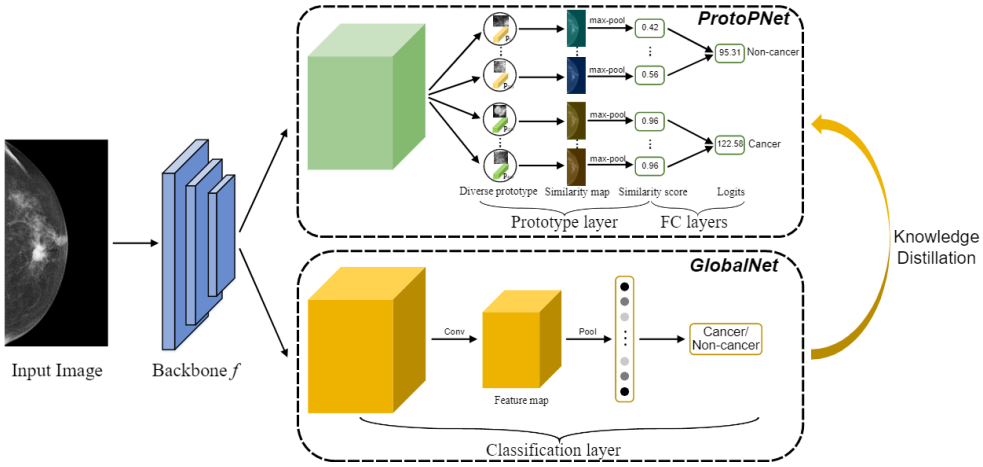


Figure 2.10: Architecture of the model BRAIxProtoPNet++ for interpretable mammogram classification [12].

Going into more detail on the model BRAIxProtoPNet++, presented in the fig. 2.10, there is a shared CNN backbone $z = f_{w_f}(x)$, w_f being its parameters and x being the input sample. The uninterpretable GlobalNet classification is represented by $\tilde{y}^U = u_{w_u}(z)$ and its decision can take only two values $\tilde{y}^U \in [0, 1]$. The ProtoPNet is defined as $\tilde{y}^P = h_{w_h}(g_p(z))$ and the decision can also take only two values $\tilde{y}^P \in [0, 1]$. The prototypes in the prototype layer are represented by $P = \{p_j\}_{j=1}^{m}$, with each class, having equally $\frac{m}{2}$ prototypes. The optimization function is given by

$$\min_{P, w_{conv}, w_u} \frac{1}{n}\sum_{i=1}^{n} \text{CrsEnt}(\tilde{y}_i^P, y_i) + \lambda_1 L_{\text{Clst}} + \lambda_2 \text{max}(0, \gamma - L_{\text{Sep}}) + \lambda_3 \frac{1}{n}\sum_{i=1}^{n} \text{CrsEnt}(\tilde{y}_i^U, y_i) + \lambda_4 L_{\text{KD}} \quad . \quad (2.10)$$

Notice, in (2.10) [12], the term associated with the cross entropy loss multiplied by the hyper-parameter λ_3 to train the uninterpretable model, with y being the ground truth and $\tilde{y}^U$ the respective decision. Plus, L_{KD} represents the Knowledge Distillation Loss (KDL), used to transfer the knowledge from GlobalNet to ProtoPNet in order to increase decision accuracy and is given by

$$L_{\text{KD}} = \frac{1}{n}\sum_{i=1}^{n} \text{max}(0, (y_i)^T(\tilde{y}_i^U) - (y_i)^T(\tilde{y}_i^P) + \omega) \quad , \quad (2.11)$$

where $(y_i)^T(\tilde{y}_i^U)$ and $(y_i)^T(\tilde{y}_i^P)$ correspond to the probability of belonging to the true class obtained by the uninterpretable model and ProtoPNet respectively. In addition, w is a positive number that represents

a predefined margin that controls ProtoPNet's confidence gain. Finally, notice how the loss associated with ProtoPNet is identical to the original presented in (2.2), but with the introduction of the hinge loss, to decrease the risk of overfitting. Also, it is recalled that the L_{Clst} and L_{Sep} were described previously in (2.3) and in (2.4), respectively.

To conclude the presentation of the model BRAIxProtoPNet++ [12], it is important to note that the authors ensure the diversity of prototypes during the projection phase. They do this by recording the indices of the training images that have already been assigned as a prototype in order to choose the closest image that has not yet been assigned.

2.3.4.C Concept-level Debugging of ProtoPNet

This section presents the work of Bontempelli *et al.* [13], in particular ProtoPDebug an efficient concept-level debugger for ProtoPNet that allows a human supervisor to provide input on which prototypes should be maintained or forgotten. The model is then adjusted to work in harmony with this supervision. It is important to mention that ProtoPNet has an interesting feature: confounding factors only impact the decisions if they are represented in the prototypes. If present, they can be corrected with supervision at the conceptual level.

There are a few advantages of concept-level supervision when compared to pixel-level annotations in particular: (1) using human supervision, with feedback through selection, is inexpensive when using an interface where it shows the prototypes as well as their activations on the images; (2) it is quite informative, enables a fine distinction between content and context, and generalizes across images and (3) by generalizing across images, it allows for faster convergence [13].

Considering a ProtoPNet P_N and the training set $D = [X, Y] = \{(x_i, y_i)\}_{i=1}^{n}$, several rounds of debugging are performed. In each round, for each prototypical part $p_j \in P$, the η training examples that activate the prototype the most are presented, then the user is asked to select the prototypes that appear to be representatives of confounding factors by analyzing the activation map.

Next ProtoPDebug extracts the "cut-out" x_r from the image x, defined with the box containing 95% of the activation of the prototypical part. Then x_r is encoded into latent space via f, representing here the convolution layers, and is added to the set of forbidden concepts F, which is organized into class-specific subsets F_k, i.e. the user may want to add it as a forbidden concept for all classes or for one in particular. Similarly, after the user selects the prototypes he considers valid, the latent representation of x_r is added to the set of valid concepts V.

After all prototypes have been inspected, P_N is updated to avoid the forbidden concepts F and not to forget the valid concepts V, then a new round begins and this continues until all prototypes are valid from the user's point of view. It is crucial to emphasize that V consists of n_v elements, denoted as $V = [v_i]_{i=1}^{n_v}$, and F comprises n_f elements, represented as $F = [f_i]_{i=1}^{n_f}$. The process of updating P_N is

obtained by combining the loss in (2.2), with two new losses[2], the forgetting loss

$$L_{\mathsf{F}} = \frac{1}{n_f} \sum_{i=1}^{n_f} \sum_{j:\text{class}(p_j)=\text{class}(f_i)} \log \frac{||p_j - f_i||_2^2 + 1}{||p_j - f_i||_2^2 + \epsilon} \quad . \tag{2.12}$$

and the remembering loss

$$L_{\mathsf{R}} = -\frac{1}{n_v} \sum_{i=1}^{n_v} \sum_{j:\text{class}(p_j)=\text{class}(v_i)} \log \frac{||p_j - v_i||_2^2 + 1}{||p_j - v_i||_2^2 + \epsilon} \quad . \tag{2.13}$$

The forgetting loss L_{F} minimizes the activation of the prototypes of each class k with respect to the forbidden concepts for that class belonging to F_k, i.e. that have to be forgotten. On the other hand, the remembering loss L_{R} maximizes the activation of the prototypes of a given class k with respect to the concepts valid for that class belonging to V_k, i.e. that cannot be forgotten.

In fig. 2.11, ProtoPDebug is used for medical image analysis to classify COVID-19 presence in chest radiographs. Two prototypes per class are considered. The user examines prototype activations and decides validity. Valid prototypes are added to the class's valid concepts, while invalid ones become forbidden concepts. The process continues until all prototypes are deemed valid by the user.

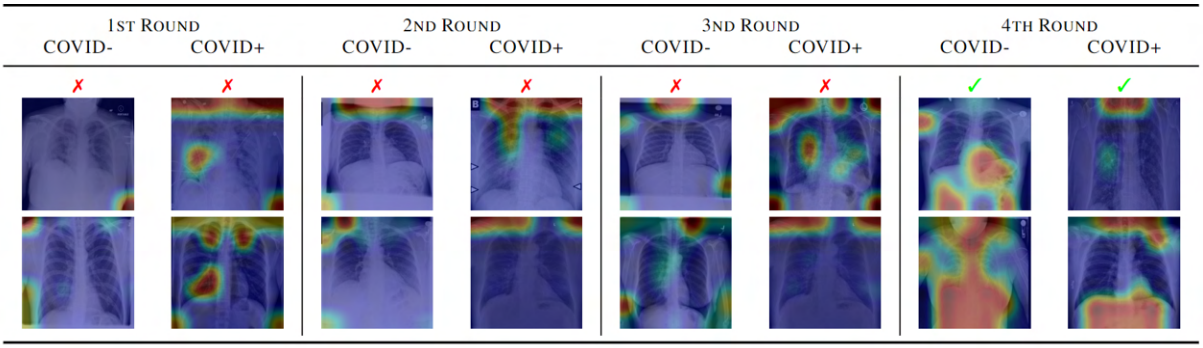


Figure 2.11: Example of ProtoPDebug usage in medical image analysis, specifically in the detection of COVID-19 from chest radiographies. Four rounds are presented with two prototypes for the classes "COVID-" (COVID absent) and "COVID+" (COVID present). In this case, the user should observe the activations of each prototype and determine whether it is valid (marked as a green check mark) or invalid (marked as a red cross). It is noteworthy that in the fourth round, the prototypes are deemed valid by the user, thereby concluding the debugging process [13].

ProtoPDebug offers an alternative to using masks for better prototype supervision, preventing them from representing confounding factors. What's especially intriguing is its potential to incorporate human feedback from medical experts, greatly aligning prototypes with medical concepts.

[2]According to the code provided on the official GitHub repository at https://github.com/abonte/protopdebug.

2.3.4.D ProtoPNet and Skin Lesions

An example where ProtoPNet was applied for melanoma detection is the work of Hussaindeen *et al.* [14]. In that work, they resorted to the use of ProtoPNet for multi-label classification, and used top-k average pooling instead of max pooling to calculate the similarity scores as was done in the IAIA-BL model [11]. Based on the Seven Point Checklist, one of the most commonly used criteria for detecting melanoma, the authors propose a prototype-based interpretable melanoma detector.

Furthermore, in fig. 2.12 it is possible to understand that a class with the highest score, whether it is the absence of a class or a sub type of a dermoscopic property, is considered to be a precise correspondence for each criteria. Additionally notice how in this case each prototypical part represents a dermatology property associated with the classification of skin lesions. For example, shown in the fig. 2.12 are the prototypes associated with Blue Whitish Veil (BWV), Dots and Globules (DaG) and Regression Structures (RS), all of which are properties that when present in the skin allow the dermatologist to make a decision regarding the diagnosis of melanoma.

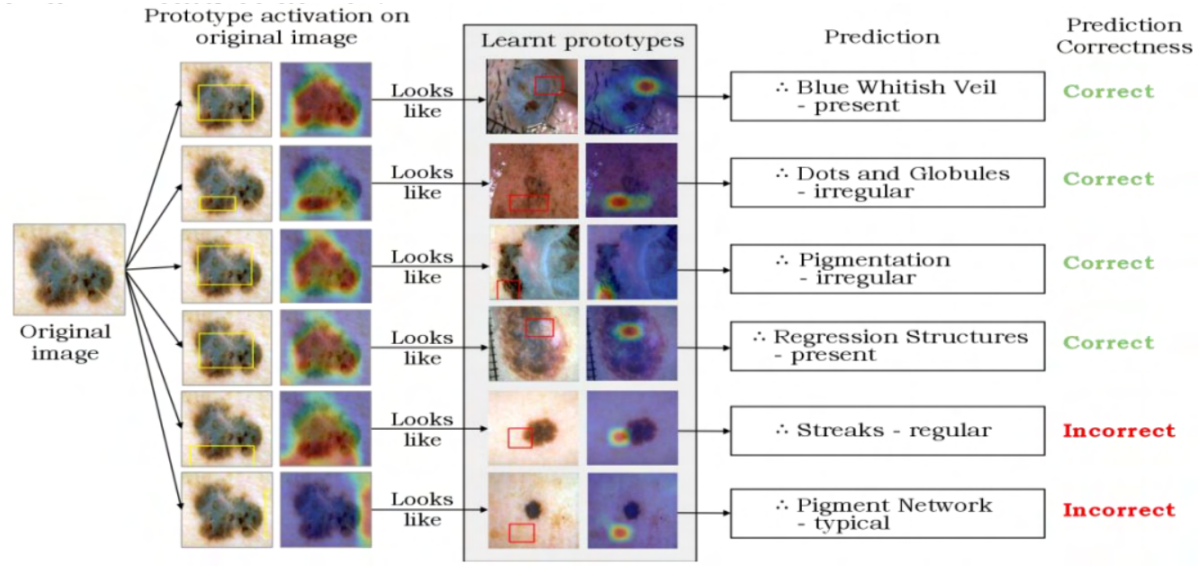


Figure 2.12: Path and explanation provided by the interpretable model proposed for assistance in melanoma diagnosis, based on the ProtoPNet [10] and the Seven Point Checklist [14].

Moreover, note how in the input image corresponding to a melanoma case, considering the first line associated with BWV, the region marked in red is quite activated and similar to the region of the respective prototype outlined with a red line. Consequently, this allows the model to conclude that BWV is indeed present in the input image.

Concluding the presentation on the application of ProtoPNet in medical image analysis, it is essential to highlight its successful implementation for Alzheimer's detection through MRI images [39], as well as its effective utilization for COVID-19 detection in CT-Scan images [40].

2.3.4.E Conclusions

The IAIA-BL model enhances ProtoPNet accuracy by employing top-k average pooling instead of global average pooling. It introduces fine annotation loss to ensure prototypes align with relevant image regions, enhancing the model's explanation capability. However, it does not tackle prototype diversity, potentially leading to repeated or similar prototypes. Increased prototype numbers may not improve image part comparison. Utilizing more diverse and prototypical parts could boost both the model's explanation and performance.

BRAIxProtoPNet++ enhances ProtoPNet accuracy with KDL, while maintaining global average pooling instead of top-k average pooling. Prototypes come from different training images to ensure diversity, but there's no mechanism to prevent biases or capture the most diagnostically relevant image regions. Lack of a validation component for prototype quality is observed. Moreover, the model in section 2.3.4.D doesn't provide an automatic diagnosis for melanoma. Its purpose is to identify structures within the lesion aligned with the Seven Point Checklist criteria for doctors to use in diagnosis. The model lacks diverse prototypes, doesn't address confounding factors, and lacks validation, potentially leading to ineffective representations of intended structures.

In the context of the research conducted in this thesis and considering the identified issues in the literature, we aim to apply the ProtoPNet framework to effectively achieve automated diagnosis of skin lesions. We explore two specific problems: a binary classification problem of Melanoma Vs Nevus and a multi-class problem involving eight categories of skin lesions. Initially, we employ prototypes corresponding to each lesion class. Furthermore, to ensure that the learned prototypes align with regions of potential clinical relevance, we investigate the use of two non-expert supervision techniques at the prototype level. These techniques involve the utilization of masks based on lesion segmentation. The first technique incorporates a component of the model loss inspired by IAIA-BL, while the second one incorporates mask information directly into the model's forward process instead of using it on the loss. Additionally, we explore a third prototype-level supervision technique similar to ProtoPDebug. Here, we employ the remembering loss as an alternative to mask-based approaches, utilizing a straightforward evaluation criterion by non-expert users. However, we integrate the remembering loss directly at the beginning of the model training process, streamlining comparisons with other supervision techniques and eliminating the need for multiple debug rounds. We also incorporate top-k average pooling and we also ensure prototype distinctiveness by intervening directly in the prototype projection process.

In the binary classification problem, we further explore a second novel approach where we only use prototypes related to a single class, specifically the malignant class. This implies that the decision and explanation depend solely on the similarity with the malignant prototypes, making the explanation simpler. In this approach, we experiment with introducing a loss component that promotes intra-class diversity of the prototypes, going beyond ensuring diversity solely in the prototype projection process.

3

Proposed Approach

Contents

3.1 Interpretable Skin Cancer Detection with Prototypes

In this section, the first adopted approach is presented, specifically focusing on the architecture, training process, and various non-expert prototype supervision techniques explored. This approach is considered more traditional as all classes have prototypes and is applied to two classification problems: the first being the binary classification of Melanoma Vs. Nevus, and the second a multiclass problem with eight classes. In the second adopted approach, exclusive to the binary problem, described in section 3.2, the model only includes prototypes related to the malignant class, simplifying the decision-making process and explanations provided by the model.

3.1.1 Interpretable Model Architecture

Our first approach, similar to ProtoPNet [10], depicted in fig. 3.1, consists of three main components: a CNN called f, a prototype layer denoted as g_p, and a fully connected layer denoted as h [10]. The input to the network is an image x_i with dimensions $H_i \times W_i \times 3$. This image is processed by f, which has parameters w_{conv} and outputs a feature map z with dimensions[1] $H_z \times W_z \times D$, where each pixel in z corresponds to a patch in x_i. The prototype layer g_p learns m prototypes, denoted as $P = \{p_j\}_{j=1}^{m}$, each one used in the corresponding prototype unit g_{p_j} to compute the L_2 distances between p_j, with dimensions $1 \times 1 \times D$, and all the $H_z \times W_z$ pixels in the feature map z. These distances are then transformed into similarity scores, resulting in an activation map. This activation map represents the similarity scores between the prototypical part and various image regions within the unit.

To obtain a single value that quantifies the affinity between the prototypical part p_j and the image x_i, top-k average pooling is applied [11]. This pooling operation reduces the activation map to a single value, indicating how prominent the prototypical part p_j is within the image. The output of each unit g_{p_j} is calculated using

$$g_{p_j}(z) = \underset{\tilde{z} \in \text{patches}(z)}{\text{avg top-k}} \log \frac{||\tilde{z} - p_j||_2 + 1}{||\tilde{z} - p_j||_2 + \epsilon}, \tag{3.1}$$

where ϵ is a small number introduced to prevent division by zero. The higher the similarity score at the output of the unit g_{p_j}, the smaller the L_2 distance between a given image patch and the prototype p_j in the latent space representation of the feature map z. Thus, a higher output value indicates a stronger presence of the prototypical part p_j within the image.

The user is required to define the number of classes, denoted as K, and the number of prototypes assigned to each class, denoted as m_k. The prototypes created aim to capture the essential features that distinguish each class within the classification task. In the final stage of the model, the fully connected layer h employs a weight matrix w_h with dimensions $K \times m$ to multiply the m similarity scores obtained. This multiplication results in the score indicating the likelihood of the input image belonging to a specific

[1] It is important to mention that $H_z = W_z$.

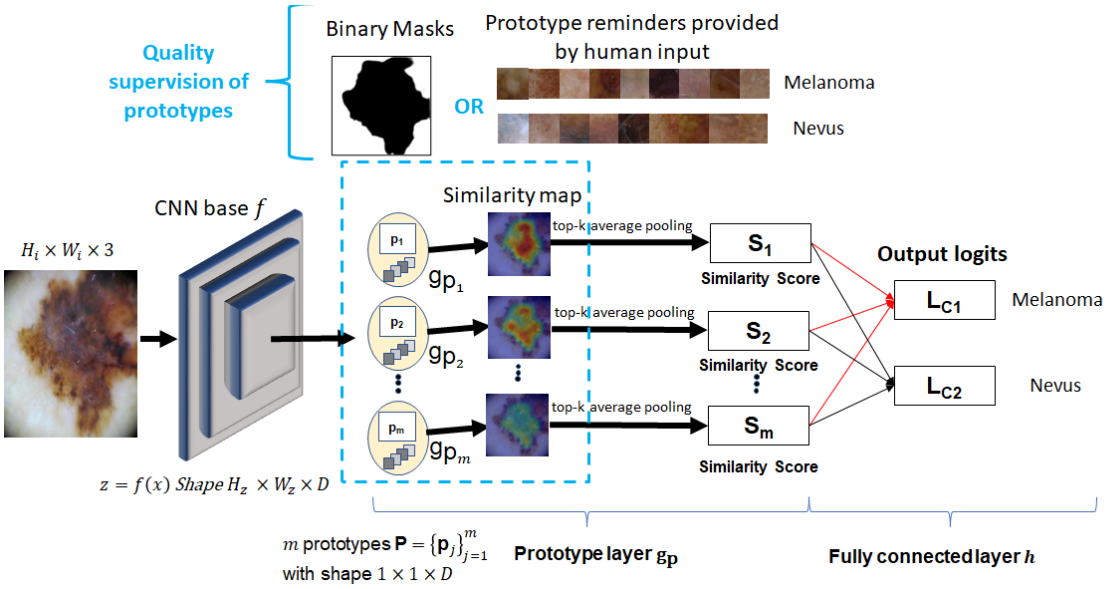


Figure 3.1: Interpretable model for skin cancer classification of Melanoma vs Nevus. The model is based on ProtoPNet [10] and enables non-expert supervision of prototypes through the use of a binary mask or human feedback. This model can be extended to a multiclass problem, specifically in the classification of 8 types of skin lesions.

class. The higher the similarity between the input image and the prototypes of class k, the greater the probability of it belonging to that class. Examining fig. 3.1, it can be observed how the interpretable model is utilized in a binary classification process between melanoma and nevus. Furthermore, the application of this model is also explored in a multiclass context, specifically with eight classes. For more details, refer to section 4.1.1.

3.1.2 Prototype Learning

The training of our model, similar to ProtoPNet [10], involves three main steps: training with a fixed final layer h, projection of prototypical parts, and training of the final layer [10]. For the sake of simplicity, we will start by describing the standard training process, without non-expert supervision on prototype quality, given the training image set $D = [X, Y] = \{(x_i, y_i)\}_{i=1}^{n}$.

In the first step, we perform joint optimization on the convolution layers and prototypical parts of the prototype layer. The optimization process is defined by

$$\min_{P, w_{conv}} L_{\mathsf{P}} \Leftrightarrow \min_{P, w_{conv}} \frac{1}{n} \sum_{i=1}^{n} \mathsf{CrsEnt}(h \circ g_p \circ f(x_i), y_i) + \lambda_1 L_{\mathsf{Clst}} + \lambda_2 L_{\mathsf{Sep}} \quad , \tag{3.2}$$

which includes the cross-entropy loss, the clustering loss with weight λ_1

$$L_{\text{Clst}} = \frac{1}{n}\sum_{i=1}^{n} \min_{j:\text{class}(p_j)=y_i} \frac{1}{\kappa}\sum \min_{z\in\text{patches}(f(x_i))}^{\text{k}} ||z-p_j||_2 \quad , \tag{3.3}$$

and the separation loss with weight λ_2

$$L_{\text{Sep}} = -\frac{1}{n}\sum_{i=1}^{n} \min_{j:\text{class}(p_j)\neq y_i} \frac{1}{\kappa}\sum \min_{z\in\text{patches}(f(x_i))}^{\text{k}} ||z-p_j||_2 \quad . \tag{3.4}$$

The second step projects prototypes onto the closest latent parts of training images from the same class, enabling the direct association of learned prototypes as parts of images. We ensure that prototypes are projected onto different training images to promote diversity [12]. In other words, each prototype is always projected onto the closest available training image that has not been assigned yet. In the final step, we optimize the parameters of the last layer while keeping the prototype layer fixed

$$\min_{w_h} \frac{1}{n}\sum_{i=1}^{n} \text{CrsEnt}(h\circ g_p \circ f(x_i), y_i) + \lambda_5 \sum_{k=1}^{K}\sum_{j:p_j\notin P_k} |w_h^{(k,j)}|. \tag{3.5}$$

This optimization aims to minimize cross-entropy loss and promote sparsity in the last layer's parameter matrix. Furthermore, it is noteworthy to mention that in our case, the parameters of the final layer are initialized as follows with $w_h^{(k,j)} = 1$ for all j where $p_j \in P_k$ and $w_h^{(k,j)} = -1$ for all j where $p_j \notin P_k$. Also, the prototypes are initialized with a uniform distribution between 0 and 1. It is important to note that with m prototypes, the prototypes can be represented by a matrix w_P of dimensions $m \times D \times 1 \times 1$.

3.1.3 Non-expert Supervision of Prototypes

Additional terms can be added to the cost function L_{P} in (3.2), to regulate and supervise the quality of the prototypes. This ensures that the prototypes represent the most relevant aspects of skin lesions and avoid confounding factors such as black borders, image corners, rulers, or hair. We explored two alternative approaches: the mask loss L_{M}, which is associated with lesion segmentation masks, inspired by the fine-annotation loss [11], or the remembering loss L_{R} derived from ProtoPDebug [13] that explores direct human-feedback on the prototypes. Both losses can be added to (3.2), with appropriate weights λ_3 and λ_4.

In addition, we also explore the use of a third supervision technique to enhance the quality of the learned prototypes, involving the utilization of binary masks. However, in this technique, instead of introducing an additional component to the loss that incorporates information from these masks, the masks are directly incorporated into the internal structure of the model during its forward process. Specifically, the masks are employed to guide the model in disregarding patches of the test image that correspond to areas outside the skin lesion. As this technique does not introduce a new term to the loss, we will refer to it as $L_{\text{P-Masked}}$.

3.1.3.A Mask Loss (L_{M})

As mentioned previously, the primary objective of L_{M} is to incorporate a supervision technique to ensure that the prototypes possibly represent more significant characteristics of skin lesions for diagnostic purposes, while minimizing the impact of confounding factors. To achieve this, a binary mask is employed to distinguish between relevant pixels (lesion, marked as 0) and non-relevant pixels (skin and other elements, marked as 1).

Given the difficulty to obtain segmentation masks provided by experts at a large scale, in this work we explore the use of an automatic segmentation model [41] to collect most of the binary masks. This approach has its limitations, as lesion segmentation networks still fail for more challenging cases [42]. Thus, it can be perceived as a proxy for binary masks provided by a non-expert.

A crucial step of this approach involves calculating the element-wise product between the binary mask M_i and the prototype activation map $\mathsf{PAM}_{i,j} = \mathsf{ScaleUp}(g_{p_j}(f(x_i)))$, where the ScaleUp operator resizes the activation map from dimensions $H_z \times W_z$ to the dimensions of the input image $H_i \times W_i$ using a 2D adaptive average pooling technique. For a particular training image x_i belonging to class y_i and with corresponding mask M_i, and given a skin lesion prototype p_j belonging to the same class y_i, the binary mask loss is defined by

$$L_{\mathsf{M}} = \sum_{i \in D} \sum_{j:\mathsf{class}(p_j)=y_i} ||\mathsf{M}_i \odot \mathsf{PAM}_{i,j}||_2. \tag{3.6}$$

By performing an element-wise multiplication between M_i and the prototype activation map, we obtain a map that indicates the regions in the image where the prototype shows activity in areas of low clinical significance. Consequently, the summation within the binary mask loss aims to minimize the number of prototype activations in these clinically insignificant areas, when these prototypes belong to the same class k as the training image x_i. This approach encourages the training algorithm to learn prototypes that effectively capture medically relevant characteristics specific to their assigned categories in skin lesions.

3.1.3.B Remembering Loss (L_{R})

The remembering loss within the ProtoPDebug framework offers a compelling alternative to L_{M} [13]. Unlike L_{M}, which relies on binary masks to identify clinically relevant regions, the remembering loss allows users to provide examples of relevant prototypes that they consider characteristic or representative of each skin lesion class. These user-selected prototypes, presented as images to the model and chosen through human input, serve as valid examples that should activate the prototypes generated by the model during the training process.

In this work, such feedback was given by a non-expert, who followed simple heuristics (*e.g.*, remem-

ber prototypes inside and in the border of skin lesions, and discard prototypes that activated in artifacts, such as dark corners, or skin). However, due to its simplicity, in the future a medical expert could be included in the process.

Let V denote the set of n_{rp} valid prototypes given by the user. For each $v_i \in V$, similarity scores between v_i and the prototypes belonging to the same class as v_i are calculated and summed. This process is repeated for all elements in V, and the average is taken, leading to

$$L_{\mathsf{R}} = -\frac{1}{n_{rp}} \sum_{i=1}^{n_{rp}} \sum_{j:\text{class}(p_j)=\text{class}(v_i)} \log \frac{||p_j - v_i||_2 + 1}{||p_j - v_i||_2 + \epsilon} \quad . \tag{3.7}$$

The remembering loss aims to maximize the activation of prototypes belonging to a specific class k relative to the input example prototypes considered valid for that class k. Please refer to section 4.3 for a more detailed understanding of the process of gathering information from human users in the form of valid prototype examples.

3.1.3.C Mask Integration in Model Forward Process ($L_{\text{P-Masked}}$)

As previously mentioned, in this supervision technique, instead of using masks in a loss component as described in section 3.1.3.A, we utilize masks to force the model to consider only the regions of the image marked as relevant by the mask (i.e., the pixels marked with 0) during the process of computing similarities between the image x_i and a given prototype p_j.

Essentially, within each prototype unit g_{p_j}, the L_2 Euclidean distance is computed between each pixel of the feature map z and the corresponding learned prototype p_j by the model. This process results in the creation of a distance matrix M_D with dimensions $H_z \times W_z$ within each unit. Thus, for a given input image, its binary mask M_i is resized from $H_i \times W_i$ using a 2D adaptive average pooling technique to $H_z \times W_z$, and the resized mask is referred to as M_{ir}. By resizing the mask, we can identify the relevant pixels in z, which have the same dimensions as a prototype, and consequently, the corresponding relevant patches in the input image. Since the similarity measure between an image and a prototype, as described in (3.1), is higher when the distance between a pixel in z and a prototype p_j is smaller, an intervention is performed on M_D to ensure that the model predominantly considers the patches marked with 0 and disregards those marked with 1 during both the similarity calculation process in the prototype unit and the prototype projection process.

Notice how the top-k operation ensures that only the k most similar pixels in z, and therefore the ones with the lowest distance value to the respective prototype, are considered in the similarity measure calculation. Lastly, it is important to remember that the learned prototype is always projected onto the most similar patch in the training image, i.e., the one associated with the smallest distance. Consequently, the

following intervention is performed in the distance matrix

$$M_D = M_D \odot (M_1 - M_{ir}) + M_{ir} \odot M_{MAX-D} \quad , \tag{3.8}$$

where M_1 represents a matrix of dimensions $H_z \times W_z$ with a value of 1, and M_{MAX-D} represents a matrix of the same dimensions but with a high distance value labeled as MAX-D. This causes the model to disregard patches associated with non-relevant areas, located outside the lesion boundary and marked with 1, while learning prototypes corresponding to regions within the lesion marked as 0 in the mask M_i.

In conclusion, this enables the model to associate non-relevant patches of a given image, marked as 1 in the mask, with high distance values and consequently low similarity to the prototypes the model should learn. On the other hand, patches marked as 0 are associated with low distance values and therefore higher similarity to the prototypes the model should learn.

3.2 One-Class Prototypes: Simplified Binary Problem Explanation

This second approach is exclusive to the binary problem, where in this case, instead of the model's explanation and decision being based on prototypes from both classes, only prototypes from one class are considered. In the case of the Melanoma vs Nevus problem, where melanoma represents the class of malignant skin lesions, the prototypes learned by the model will solely belong to this class.

This approach becomes intriguing with the idea that by selecting prototypes corresponding to the malignant class in the binary problem, the decision is solely based on their similarity. In other words, the explanation and decision pathway become even easier to comprehend. The decision is made purely through the following reasoning: if the image to be diagnosed has patches similar to the prototypes of the malignant class, then the probability of classifying it as malignant is higher. Conversely, if the image patches are dissimilar, the probability of it being malignant is lower. Thus, the decision solely relies on the similarity to the malignant class and is not influenced by the similarity to prototypes of the other benign class.

Why do we choose to have prototypes only from the malignant class in this case, and not from the benign class? From a medical standpoint, a malignant lesion may exhibit dermatological characteristics similar to benign ones, but with additional malignant features. On the other hand, a benign lesion should not present malignant characteristics. Therefore, it makes sense to compare the image with malignant prototypes.

3.2.1 Architecture

The majority of the model's structure is identical to the one presented in section section 3.1.1; however, there are, obviously, some differences. In this case, we still have m prototypes, but all belonging to the same class, specifically at the prototype layer. As for the last layer h, the input consists solely of m similarity values with melanoma prototypes, and the output consists solely of one logit associated with the probability of classifying the image as melanoma. In this scenario, in addition to the weight matrix w_h with dimensions $1 \times m$, there is also a bias term b_h. In fig. 3.2, we can observe how the mentioned differences are reflected in the structural changes of the model when compared to the illustration depicted in fig. 3.1.

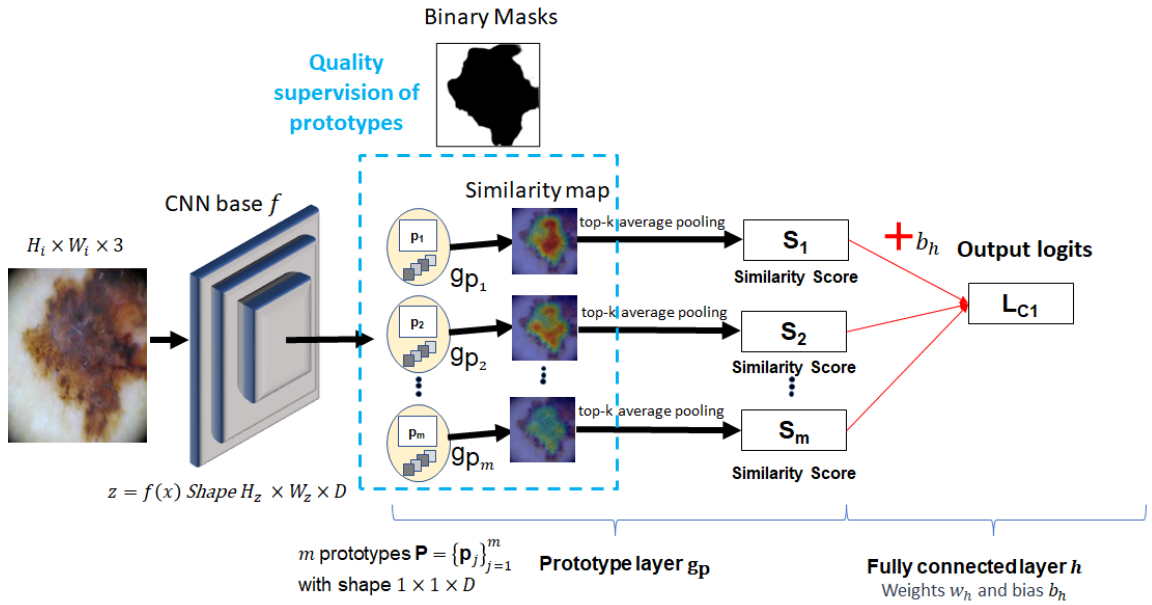


Figure 3.2: Interpretable model for skin cancer detection, specifically designed for a binary problem, as the explanation and decision only take into account prototypes from one class. For instance, in the binary problem of Melanoma Vs Nevus, the chosen prototypes are exclusively from the melanoma class, given that it is the malignant class. In this second approach, we solely explore the use of non-expert supervision to regularize the quality of prototypes using binary masks. This alternative approach involved more concise experiments.

It is important to note that in this case, we have only one output logit. Instead of using the softmax function, we utilize a sigmoid function, which is defined as $\sigma(x) = 1/(1 + e^{-x})$. It is imperative to recall that the sigmoid function possesses the characteristic of compressing any input value within the range of 0 to 1. As the input tends towards positive values, the sigmoid function converges towards 1, signifying a higher probability. Conversely, for negative inputs, the sigmoid function approaches 0, indicating a lower probability.

Furthermore, in our specific scenario, our prototypes belong to the malignant class, but the assigned label is 0 instead of 1. Hence, it is crucial to mention that we initialize the weights of w_h with -1 and the

bias b_h with a value of 20. This choice is based on the fact that label 1 represents nevus. Consequently, the less similarity there is with the melanoma prototypes, the more positive the logit value becomes, resulting in a probability higher than 0.5 and classifying the image as nevus. Conversely, the more similarity there is with the melanoma prototypes, the more negative the output of the logit becomes, associated with a probability lower than 0.5, and assigning label 0.

3.2.2 Training and Non-expert Supervision

In this case, we still have three essential steps during training, but with slight differences as we only have prototypes related to one class. The three steps are as follows: 1) training with a fixed final layer h, 2) projection of prototypical parts, and 3) training of the final layer.

Given the training image set $D = [X, Y] = \{(x_i, y_i)\}_{i=1}^{n}$, in the first step, we still perform joint optimization on the convolution layers and prototypical parts of the prototype layer. The optimization process is defined by

$$\min_{P, w_{conv}} L_{\text{P-1C}} \Leftrightarrow \min_{P, w_{conv}} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(h \circ g_p \circ f(x_i), y_i), \tag{3.9}$$

which includes the cross-entropy loss. Notice how in this approach, we only have the presence of cross-entropy loss. This is because, with the model having prototypes from only one class, there is no justification for using clustering and separation loss. These losses aim to ensure that prototypes are represented in the latent space close to the training images of the same class and far from the prototypes and training images of the other class. However, in the model where the decision relies solely on prototypes from one class, the use of clustering and separation loss is not necessary, and the model can converge without them.

In the case of the second step, which is the prototype projection process, it is exactly identical to the one described in section 3.1.2, but for prototypes of a single class. In the final step, we update only the parameters of the last layer while keeping the remaining model parameters fixed. However, since the weights are only relevant to the connection between prototypes of one class and a logit, the second term, present in (3.5), becomes unnecessary as there are no similarity computations involving prototypes from the other class. This implies that the third step can be summarized in terms of loss as cross-entropy, as shown by

$$\min_{w_h, b_h} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(h \circ g_p \circ f(x_i), y_i) \tag{3.10}$$

Finally, regarding the supervision method explored in this approach to ensure that the prototypes correspond to the interior of the skin lesion rather than confounding artifacts outside its boundary, we decided to use only one technique. As the first approach in section 3.1 already extensively studied the possibility of the three techniques, namely L_{M}, L_{R}, and $L_{\text{P-Masked}}$, and their results are presented

in chapter 5, we chose to employ only a single technique in this approach. Specifically, we utilized the $L_{\text{P-Masked}}$ technique presented in section 3.1.3.C, but as it is applied in this approach with prototypes from only one class, we refer to it as $L_{\text{P-1C-Masked}}$ to differentiate it.

3.2.3 Intrinsic Intra-Class Diversity for Prototype Training

Since in this model, the loss function is solely based on cross-entropy loss and other loss functions such as clustering and separation loss are not present, it becomes interesting to study the introduction of a loss that can promote intra-class diversity among the prototypes belonging exclusively to the malignant class. It is easier to balance the coefficient of this loss when introduced in $L_{\text{P-1C}}$ due to the presence of fewer components in the optimization function, when compared with (3.2). Also because we only have prototypes from a single class.

A particular characteristic of skin cancer, as mentioned earlier, is that there is often a high intra-class variability and low inter-class diversity. Therefore, it is plausible to hypothesize that promoting intra-class diversity among the learned prototypes could aid in model generalization and improve its performance.

In this case, the term of the loss function that promotes intra-class diversity, which we aim to maximize, is represented by L_{ICD}. Essentially, the cost is calculated based on the average Euclidean distance among the prototypes, which, as a reminder, all belong to the same class. Let's denote the matrix of intra-class distances between the prototypes as M_{DP}, which has dimensions $m \times m$. This matrix is symmetric, with zeros on the diagonal, indicating that the number of unique distances is $u = (m \times m - m)/2$. Consequently, we only need to consider the distances present in the upper triangular part of the matrix, excluding the diagonal. By normalizing the distances between 0 and 1 using the maximum distance, summing them up, and dividing by u, we obtain the average of the normalized intra-class distances, which represents the cost of the L_{ICD} loss. This can be expressed as follows:

$$L_{\text{ICD}} = -\frac{1}{u} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \left(\frac{||p_i - p_j||_2}{\max(M_{DP})} \right) \quad , \tag{3.11}$$

Here, m represents the number of prototypes, p_i and p_j are the individual prototypes, and $||\cdot||_2$ denotes the Euclidean distance. The maximum distance in the matrix M_{DP} is obtained using $\max(M_{DP})$. The coefficient associated with L_{ICD} is represented by λ_6. When added to (3.9), it contributes to the overall loss function. The introduction of L_{ICD} allows for contributing to the promotion of diversity in the prototypes learned by the model in a more intrinsic manner. This is because, until now, the assurance of prototype diversity, in the sense that they are all distinct and non-repetitive, was derived from the process of prototype projection. However, now we have a term in the loss function that promotes diversity based on the Euclidean distance between the prototypes' representations in the latent space. This interest in promoting prototype diversity was inspired by the work of Wang *et al.* [43].

4

Experimental Set-Up

Contents

4.1 Datasets

The datasets used were ISIC 2019 [1–3], PH^2 [15], and Derm7pt [9]. The ISIC 2019 dataset was used for the training and validation process, while the PH^2 and Derm7pt datasets were used for testing. These test datasets play a crucial role in understanding the generalization ability of trained models when exposed to images from a different hospital domain with different distribution patterns.

4.1.1 Training and Validation Datasets

The publicly available training set of ISIC 2019 [1–3] consists of 25,331 dermoscopic images of skin lesions, classified into 8 different diagnostic categories. The 8 diagnostic classes of skin lesions are as follows: Actinic keratosis (AK), Basal cell carcinoma (BCC), Benign keratosis (BKL), Dermatofibroma (DF), Melanoma (MEL), Melanocytic nevus (NV), Squamous cell carcinoma (SCC), Vascular lesion (VASC). However, the accompanying test set, with a total of 8,238 images, does not have publicly disclosed labels, and as a result, it was not utilized in this thesis work. This information is summarized in table 4.1.

Table 4.1: Number of samples in the original training and test dataset of ISIC 2019 [1–3].

Original ISIC 2019 Dataset	Total	AK	BCC	BKL	DF	MEL	NV	SCC	VASC
Train	25331	867	3323	2624	239	4522	12875	628	253
Test	8238								

From the images in the original training set of the ISIC 2019 dataset, duplicate images were removed, and this set was divided into training and validation sets. Consequently, the training dataset and validation dataset used in this thesis work, obtained from the original ISIC 2019 training dataset [1–3] are summarized in table 4.2. Additionally, it should be noted that out of the original 25,331 images, the dataset was reduced to 25,294 after removing 37 duplicate images. The remaining 25,294 images were divided into 20,228 for training and 5,066 for validation purposes. Furthermore, this work addresses two problems: a binary classification problem and a multi-class classification problem. In the binary problem, only samples from two classes, MEL and NV, are considered. However, in the multi-class problem, all eight classes are taken into account.

Table 4.2: Number of samples in the training and validation datasets used in this thesis work, obtained from the original ISIC 2019 training dataset [1–3].

Dataset	AK	BCC	BKL	DF	MEL	NV	SCC	VASC	Total
Train	687	2653	2089	191	3611	10293	502	202	20228
Validation	173	664	525	48	904	2575	126	51	5066

Additionally, in the multiclass problem, we also evaluate how accurately the assigned labels were placed within the correct group from a malignant versus benign point of view. In this case, the malignant classes consist of AK (pre-malignant), BCC, MEL, and SCC, while the remaining classes are benign. This means that apart from assessing the recall of each class when the model is trained with all 8 classes, we can also observe how accurately the labels were assigned within the malignant or benign group. The same analysis is conducted with the test datasets.

4.1.2 Test Datasets

One of the datasets used for testing was PH^2 [15]. Despite being a small dataset with only 200 dermoscopy images, out of which 40 represent cases of melanoma (MEL) and 160 cases of nevus (NV), it is an interesting dataset to analyze the model's generalization when exposed to images from a different hospital domain.

Additionally, another dataset considered and used in this work as a test set was Derm7pt [9]. It is important to note that in this work, the classes lentigo, melanosis, and miscellaneous from the Derm7pt [9] dataset were not used. Therefore, for testing purposes, all images in Derm7pt [9] belonging to the following classes were considered: Basal cell carcinoma (BCC), Benign keratosis or seborrheic keratosis (BKL), Dermatofibroma (DF), Melanoma (MEL), Nevus (NV), and Vascular lesion (VASC). Thus, this test set has 6 classes in common with the 8 classes present in the training and validation process. In table 4.3, a summarized overview of this information is provided, including the number of samples for each class.

Lastly, in the binary classification problem, only the MEL and NV classes are used for testing. In the case of the multiclass problem, when using the PH^2 dataset, only the MEL and NV classes can be considered. However, when using the Derm7pt dataset, all six classes can be taken into account.

Table 4.3: Number of samples in the test datasets used in this thesis work, obtained from the PH^2 [15] and Derm7pt [9] datasets.

Test Datasets	AK	BCC	BKL	DF	MEL	NV	SCC	VASC	Total
PH2					40	160			200
Derm7pt		42	45	20	252	575		29	963

4.2 Image and Mask Preprocessing

As previously mentioned, datasets of skin dermoscopy images will be required. However, these images possess a notably high resolution in their original format, which calls for the resizing of the images to achieve an optimal size. In particular, all input images for the various experimental approaches were resized to a standardized dimension of $H_i \times W_i \times 3 = 224 \times 224 \times 3$. During the image resizing process,

to preserve the original image scale, the image was first made square by adding black pixels based on the maximum width or height value. Only then was it resized to the desired smaller size.

In two of the prototype supervision processes, as mentioned earlier, binary masks will be utilized to discriminate the most relevant areas. This ensures that the prototypes represent the significant regions of the skin lesion, while minimizing the impact of confounding factors such as image corners, black borders, air bubbles, hair, or rulers.The masks contain only values of 0 or 1. Pixels marked as 0 represent the relevant pixels, while those marked as 1 represent the non-relevant pixels. Please refer to fig. 4.1 to visualize the image and mask preprocessing process mentioned.

Figure 4.1: Example of a training image from the ISIC 2019 dataset [1–3] belonging to the melanoma class, shown in its original size (left), the resized input size of the model (top right), and the corresponding binary mask (bottom right). The images are displayed to scale, preserving their original proportions.

In the case of images from the ISIC 2019 dataset [1–3], some of these images were accompanied by corresponding segmentation masks from the task 1 of the ISIC 2018 dataset challenge [3, 44]. The segmentation masks have a value of 0 outside the lesion segmentation region, which can be considered less relevant for diagnosing the type of skin lesion, and a value of 1 within the lesion segmentation region. To address this, the segmentation masks were inverted, meaning that 0 was swapped with 1 and vice versa. Additionally, for the remaining images, an automatic segmentation model was employed to obtain segmentation masks [41][1], which were subsequently inverted as previously described. Consequently, we have binary masks for all the training and validation images. For the binary problem, approximately 14% of the masks were derived from the ISIC 2018 dataset, while the remaining 86% were generated through the segmentation algorithm. As for the multiclass problem, approximately 10% of the masks originated from the ISIC 2018 dataset, with the remaining 90% obtained using the segmentation algorithm.

[1] https://github.com/qubvel/segmentation_models.pytorch

In the case of the PH^2 test dataset [15], manually obtained segmentation masks were available for all the images. Therefore, only the inversion of masks was required. Regarding the Derm7pt test dataset [9], all binary masks had to be obtained by inverting segmentation masks generated through a segmentation algorithm [41]. Lastly, the binary masks have the same dimensions as the images, specifically $H_{BM} \times W_{BM} \times D_{BM} = 224 \times 224 \times 1$.

4.3 Human-in-the-Loop Information Gathering

As previously mentioned, one of the explored supervision techniques, to ensure that prototypes do not represent confounding factors beyond the lesion boundary involves the incorporation of human-provided information, which will be provided as input to the model, see section 3.1.3.B. Similar to ProtoPDebug [13], the idea is for the user to provide the model with examples of valid prototypes that the model should learn and that hold relevance from the user's perspective. In ProtoPDebug [13], there are two types of prototypes that can be provided: prototypes that the model should remember, representing valid concepts in the latent space, and prototypes that are deemed irrelevant for clinical diagnosis, which the model should forget.

In our case, we decided to solely utilize the remembering loss, denoted as L_{R}, as we found that using the forgetting loss, L_{F}, either alone or in conjunction with the remembering loss did not perform as effectively in ensuring a high number of valid prototypes as when only using the remembering loss, L_{R}. The criterion we established for a prototype to be considered valid is that it must correspond to an area within the skin lesion or at least its boundary, see fig. 4.2, allowing for a fairer comparison between the L_{P} approach using the remembering loss (L_{P} + L_{R}) and the approach employing mask loss (L_{P} + L_{M}).

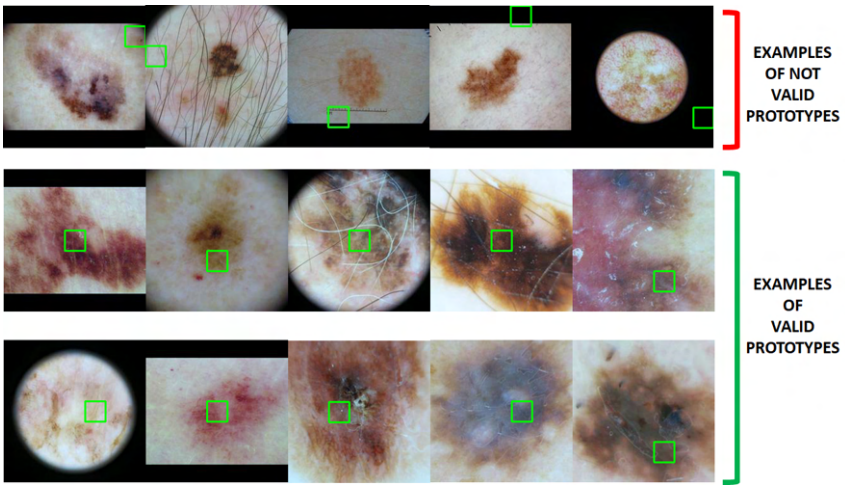


Figure 4.2: Examples of prototypes that are not considered valid (first row) and those that are considered valid (second and third rows) from the user's perspective, according to the criterion that a prototype must be contained within or on the boundary of the skin lesion to be deemed valid.

In fig. 4.2, examples of prototypes that are not considered valid from the user's perspective can be observed. In the first row, we can see prototypes (green-bordered squares) that correspond to a corner of the image far from the lesion (1st prototype), hair (2nd prototype), a ruler (3rd prototype), and black borders (4th and 5th prototypes). Additionally, in the second and third rows, we have examples of prototypes that are considered valid, as they are all contained within or on the boundary of the skin lesion. The user provided examples of prototypes by observing previously trained models using the L_{P} approach without any regularization of prototype representation quality, as well as models trained with L_{M}. For each prototype in a trained model, the 10 most activated prototypes were observed, and each of those prototypes was classified as valid or invalid, see fig. 4.3. The corresponding image containing the prototype and the index of the image patch identifying the prototype were saved.

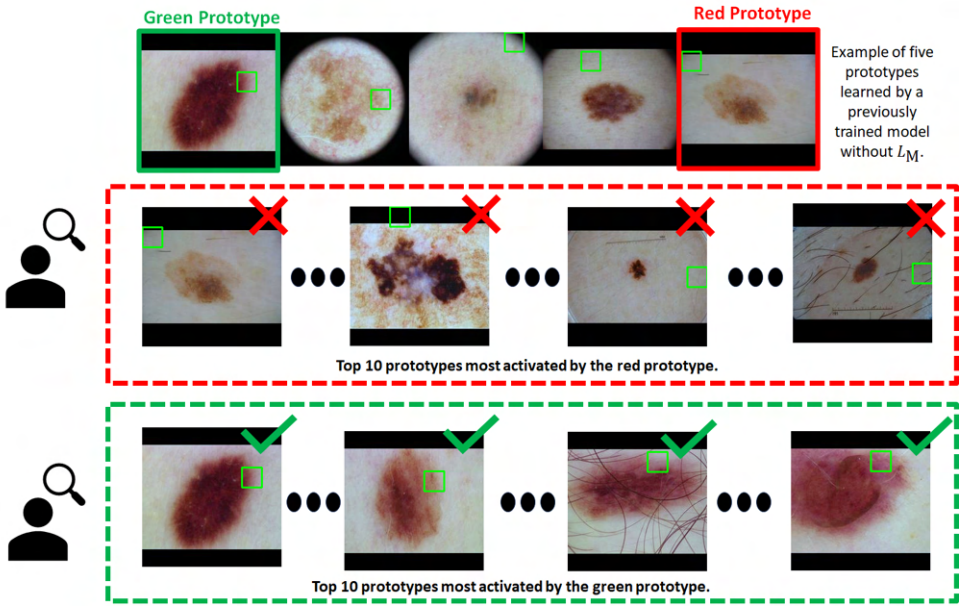


Figure 4.3: An illustration exemplifying the evaluation performed by the human user for the top 10 prototypes most activated by each of the prototypes learned by a previously trained model without L_{M}. The red cross represents a prototype deemed invalid by the user, while the green check-mark represents a prototype considered valid by the user.

In the binary problem, each model had 18 prototypes, requiring the user to observe and classify 180 prototype examples based on the given criteria. In the multi-class problem, each model had 24 prototypes, resulting in the classification of 240 prototypes. After performing this process for a minimum of 3 models for each problem, 25 prototypes were randomly selected per class from the user-defined set of valid prototypes to represent the remembering prototypes in the L_{P} + L_{M} approach for the binary problem. For the multi-class problem, 6 remembering prototypes per class were used. The difference in the number of remembering prototypes per class between the binary and multi-class problems was due to limitations in the hardware.

With ProtoPDebug [13], the user begins with a trained model and conducts multiple rounds of debugging. The user observes a specified number, n_{ap}, of the most activated prototypes for each prototype and classifies them as valid or invalid. The process continues until all the n_{ap} associated prototypes for each learned prototype are considered valid. However, in our approach with L_{R}, the model is trained from scratch using the information provided by the human user. In our case, we have 25 example prototypes per class in the binary problem, and 6 per class in the multi-class problem, which are provided as input to the model. Only a single round is conducted. We do this because it enables us to train multiple models using the same valid prototype examples as input, thus facilitating a fairer comparison between different models trained with L_{R}. Furthermore, the absence of multiple rounds, but rather just one, allows for a fairer comparison with L_{M}.

Moreover, there is another distinction between the ProtoPDebug [13] approach and our method. In the ProtoPDebug approach, a white image with a patch in the center, which could vary in size, was used as input to the model. In contrast, in our case, we provide the input image itself, which includes the prototype along with additional information, specifically the index number that identifies the fixed-size patch corresponding to the prototype within the input image, see fig. 4.4.

Recalling and justifying the case of using only L_{R} alone and not using only L_{F} or both simultaneously $L_{\mathsf{R}}+L_{\mathsf{F}}$ is due to the fact that an initial experiment was conducted for the binary case. In this experiment, we had 18 prototypes (9 per class) and used 12 models. These models were trained from scratch using the ResNet-18 CNN backbone and different configurations of D and top-k. The experiment was designed to test the three cases, and the results were as follows:

- Experience 1 (L_{R}) - 9 out of 12 models learned at least 15 prototypes considered valid from the user's perspective.

- Experience 2 (L_{F}) - 5 out of 12 models learned at least 15 prototypes considered valid from the user's perspective.

- Experience 3 ($L_{\mathsf{R}} + L_{\mathsf{F}}$) - 6 out of 12 models learned at least 15 prototypes considered valid from the user's perspective.

Given that the scenario where only L_{R} was utilized ensured that 75% of the models in the experiment had at least 80% of valid prototypes from the user's perspective, the decision was made to exclusively employ L_{R} and abstain from using L_{F}.

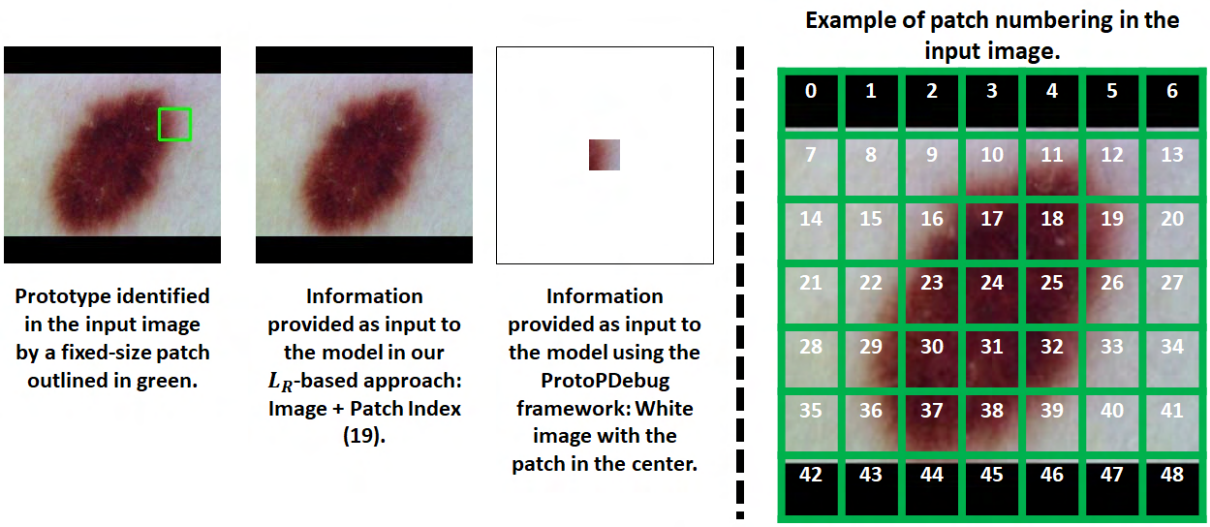


Figure 4.4: Illustration demonstrating the difference in how a prototype deemed valid by the human user is provided as input data to the model in our approach with L_{R} compared to the case when ProtoPDebug [13] is used.

4.4 Model Configurations and CNN Architectures

The CNN backbone architectures used for feature extraction in the various approaches within the ProtoPNet framework were as follows: ResNet-18 [45], ResNet-50 [45], EfficientNet B3 [46], Densenet-169 [47], and VGG-16 [48]. Additionally, corresponding non-interpretable baseline architectures, can also be referred to as black-box architectures, were considered for comparative purposes in both the binary and multi-class problems. In all cases, the CNNs were initialized with the weights of the corresponding architectures pre-trained on ImageNet.

All approaches were trained for 21 epochs, from 0 to 20, with the prototype projection operation performed at epochs 5, 10, 15, and 20. During the projection epochs, 10 iterations of optimization of the last layer are performed. Epochs 0 to 4 serve as a warm-up phase. Similar to IAIA-BL [11], the warm-up epochs train the parameters of the models related to the two convolution layers (add-on layers) before the prototype layer with a learning rate of 2×10^{-3}, while the prototypes are trained with a learning rate of 3×10^{-3}. In the standard epochs, the parameters related to features, add-on layers, and prototypes are updated with learning rates of 2×10^{-4}, 3×10^{-3}, and 3×10^{-3}, respectively, and with a decay of the learning rate of each parameter group by 10^{-1} every 5 epochs. In the iterations of optimization of the last layer after the projection process, the learning rate is set to 1×10^{-3}. The optimization algorithm used was Adam.

Additionally, the models were trained with online data augmentation. The online data augmentation processes employed were vertical and horizontal flips with a probability of 0.5, and ColorJitter with

parameters for brightness, contrast, saturation, and hue set to 0.4 for the first three, and 0.1 for the last one. The training batch size had a dimension of 75 [11]. The number of prototypes remained fixed at $m = 18$ in the binary problem $(K = 2)$, with the number of prototypes per class $m_k = 9$. In the multiclass problem with $K = 8$, the number of prototypes was set to $m = 24$, and $m_k = 3$ for each class.

Recalling that in both the binary and multi-class problems, in the first approach (section 3.1), four scenarios were considered: *i)* without non-expert supervision on the quality of prototypes L_{P}, *ii)* with non-expert supervision using $L_{\text{P}} + L_{\text{M}}$, *iii)* with non-expert supervision using $L_{\text{P}} + L_{\text{R}}$, and *iv)* with non-expert supervision $L_{\text{P-Masked}}$. In these cases, the coefficients of the different loss function terms during training were as follows[2]: $\lambda_1 = 0.8$, $\lambda_2 = 0.08$, $\lambda_3 = 0.001$, $\lambda_4 = 0.02$, and $\lambda_5 = 1 \times 10^{-4}$ [10, 11, 13].

It should be noted that for the different CNN backbone architectures in the scenarios *i* and *ii* from the first approach, models were trained with the following possibilities for the dimension of the latent space, D: 128, 256, and 512. In the cases where ResNet-18, ResNet-50, EfficientNet B3, and Densenet-169 CNNs backbones were used, the possible values for top-k were as follows: 1, 3, 7, 10, 13, 16, 19, 22, 25, 28, 31, 40 and 49. This is because for these architectures, the output feature map z of the CNN backbone has dimensions $H_z = W_z = 7$. On the other hand, when using VGG-16 with $H_z = W_z = 14$, the possible values for top-k were: 1, 12, 28, 40, 52, 64, 76, 88, 100, 112, 124, 160, 196. In summary, the models were trained with 13 possible values for the top-k between 1 and the maximum value $H_z \times W_z$. In the initial approach, each architecture in scenarios *i* and *ii* requires the training of 39 models. This is due to the existence of 39 possible configurations for D and top-k, from which the best configuration is chosen. Specifically, D can take on three values, and top-k can take on 13 values. Since the search for these parameters in the mentioned scenarios was a time-consuming process, in the remaining scenarios, a shorter parameter search was conducted based on the results that had already been obtained.

Regarding the $L_{\text{P}} + L_{\text{R}}$ scenario, both in the binary and multi-class problems, only 10 models were trained for each architecture, representing only 10 configurations of D and top-k, from which the best one was chosen. These configurations were chosen based on the best results in terms of Balanced Accuracy (BA) from the scenario *ii*. We would like to remind you again that in this scenario, in addition to the regular images provided as input to the model, valid prototype examples are also provided. This allows the model to learn prototypes similar to the ones provided and chosen by a human user. Specifically, in the binary case, 25 examples per class are provided, and in the multiclass case, 6 examples per class are provided, as mentioned in section 4.3. Finally, in scenario *iv* of the first approach, we trained each architecture with the same and best configuration of D and top-k as obtained in scenario *ii*.

Referring to the literature, initial experiments supported values close to $\lambda_1 = 0.8$ (approximating 1) and $\lambda_2 = 0.08$ (approximating 0.1) for our cost function coefficients. To validate this, we conducted a broader range experiment with λ_1 and λ_2 spanning 0.001, 0.01, 0.1, 1, and 10. In this experiment, the

[2]Based on empirical observations, the reported coefficient values in the referenced articles were suitable for our problem and validation dataset.

ResNet-18 CNN backbone was used with $D = 128$ and top-k=13 for the binary problem. The $L_P + L_M$ scenario was considered, with $\lambda_3 = 0.001$. Analyzing fig. 4.5, we can observe that combinations where λ_1 and λ_2 are around 1 and 0.1, respectively, yielded higher values of BA. Therefore, the decision was made to retain the values of 0.8 and 0.08, as referenced in the literature. A similar behavior was observed in the multiclass problem.

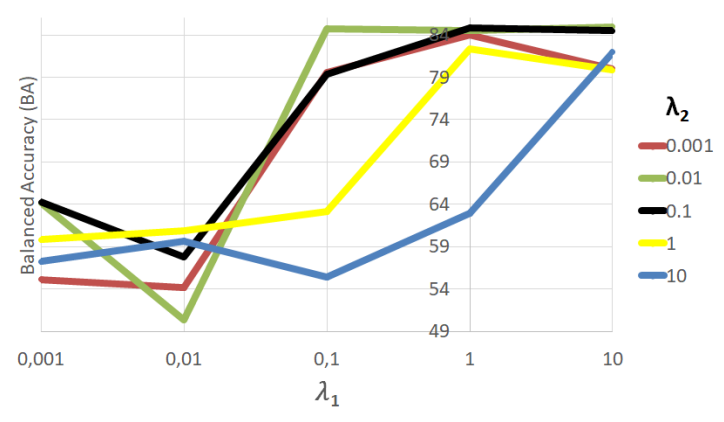


Figure 4.5: Observation of the BA behavior concerning the values of λ_1 and λ_2 when using the ResNet-18 as the CNN backbone, with $D = 128$ and top-k=13, in the $L_P + L_M$ scenario with $\lambda_3 = 0.001$. The horizontal axis represents the value of λ_1 associated with the cluster loss, while the colors represent different values of λ_2 associated with the separation loss. Note that the values of BA were higher for cases where λ_1 was around 1 and λ_2 was around 0.1. Therefore, the decision was made to fix the value of λ_1 at 0.8 and λ_2 at 0.08, as referenced in the literature.

A similar experiment was conducted to observe the behavior of BA concerning the value of λ_3, the coefficient associated with L_M. The ResNet-18 was considered as the CNN backbone with $D = 128$ and top-k=13, and $\lambda_1 = 0.8$ and $\lambda_2 = 0.08$ were used for the binary case. It was found that the value $\lambda_3 = 0.001$ was suitable for the binary problem, as shown in fig. 4.6, and a similar behavior was also observed in the multiclass case.

Regarding the coefficient value associated with L_R in the scenario $L_P + L_R$, since in ProtoPDebug they exemplified its use with $\lambda_4 = 0.02$, we decided to assess if this value remained suitable for our dataset and problem, see fig. 4.7. The experiment was conducted considering the case where we used ResNet-18 as the CNN backbone with $D = 128$ and top-k=13, setting $\lambda_1 = 0.8$ and $\lambda_2 = 0.08$. Consequently, we considered 5 possible values for λ_4: 0.0002, 0.002, 0.02, 0.2, and 2. It was observed that with λ_4 values below 0.02, the performance in terms of BA was better, but the L_R's purpose was not being fulfilled. The objective of L_R is to improve the prototypes that the model learns, ensuring that they are considered valid from the user's perspective as long as they are within or on the boundary of the lesion, but not outside of it. For values greater than or equal to 0.02, the criterion was satisfied for 17 out of the 18 prototypes. However, if we increased the value beyond 0.02, the BA started to decrease. Therefore, it was concluded that the value $\lambda_4 = 0.02$ was appropriate, and it was also adopted for the

multiclass problem and the other CNN backbones.

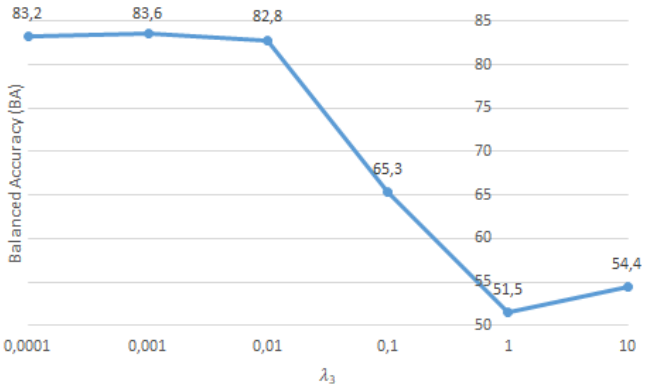


Figure 4.6: Observation of the behavior of BA concerning the value of λ_3 in the $L_{\text{P}} + L_{\text{M}}$ scenario for the binary case. The ResNet-18 was used as the CNN backbone with $D = 128$ and top-k=13, and $\lambda_1 = 0.8$ and $\lambda_2 = 0.08$ were considered. Six possible values for λ_3 were experimented with, specifically 0.0001, 0.001, 0.001, 0.01, 0.1, 1, 10. It was found that the value of $\lambda_3 = 0.001$ was the most suitable, as it resulted in the highest BA value. The same value of $\lambda_3 = 0.001$ was also adopted for the remaining CNN backbones.

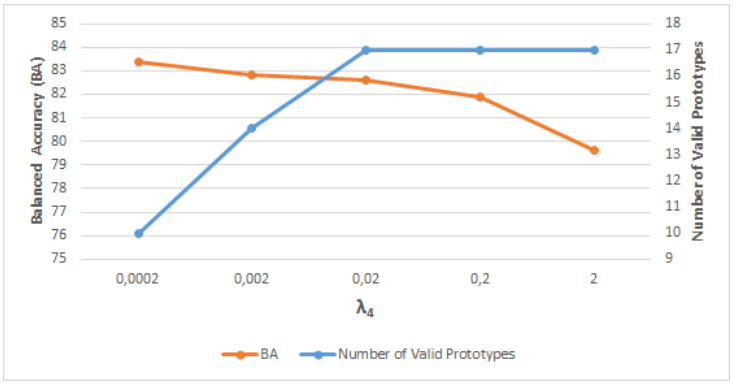


Figure 4.7: Observed behavior in terms of BA and the number of prototypes considered valid by the user as a function of λ_4. We considered the case where the CNN backbone is ResNet-18 with $D = 128$ and top-k=13, while fixing $\lambda_1 = 0.8$ and $\lambda_2 = 0.08$ in the scenario $L_{\text{P}} + L_{\text{R}}$. Five possible values for λ_4 were taken into account: 0.0002, 0.002, 0.02, 0.2, and 2. The analysis concludes that $\lambda_4 = 0.02$ was the most suitable choice since it allowed for a high number of prototypes considered valid by the user without significantly compromising the BA value, as it continues to decrease when λ_4 is further increased.

Regarding the second approach described in section 3.2, besides comparing with black-box models, we have three scenarios: *I)* without non-expert supervision $L_{\text{P-1C}}$, *II)* with non-expert supervision $L_{\text{P-1C-Masked}}$, and *III)* with non-expert supervision and intra-class diversity promotion $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$. Please be reminded that in this case, we only have prototypes of the malignant class, so $m = 9$. Furthermore, in this second approach, being a smaller-scale experiment, we decided to set the same values for all architectures. Specifically, we fixed the CNN backbones to the previously mentioned five choices, with $D = 256$ and a top-k value of 1, which implies that top-k average pooling simplifies to max pooling

in this case. We chose to use only top-k=1 since increasing the number of top-k values resulted in a performance decrease in this second approach, see fig. 4.8, whereas in the first approach, the same issue was not observed. In scenario *III*, the value of λ_6, representing the coefficient that reflects the weight of L_{ICD} in the cost function, is set to $\lambda_6 = 0.005$. To select the value of λ_6, the performance effect in terms of BA was observed for seven possible values. Specifically, the considered values were 0.1, 0.05, 0.01, 0.0075, 0.006, 0.005, and 0.001, see fig. 4.9.

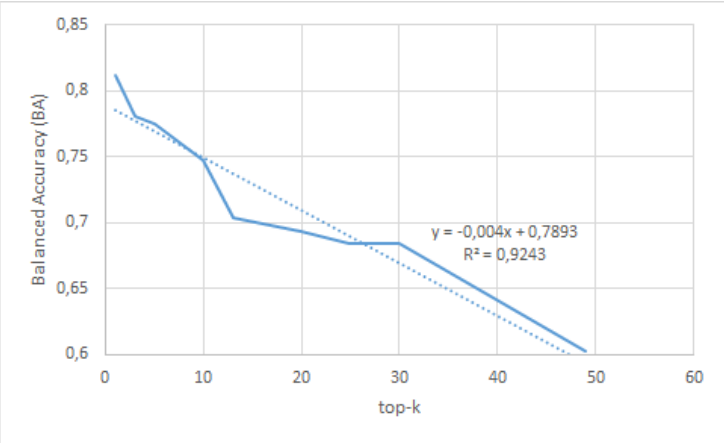


Figure 4.8: Example of increasing the top-k value for the $L_{\text{P-1C-Masked}}$ scenario in the second approach. The second approach is exclusive to the binary problem, where we only have prototypes related to the malignant class. In this specific example, we use the ResNet-18 CNN backbone with a fixed D of 256. It is evident that there is a decreasing trend in performance, measured by BA, as the top-k value increases, hence the decision to fix top-k at 1.

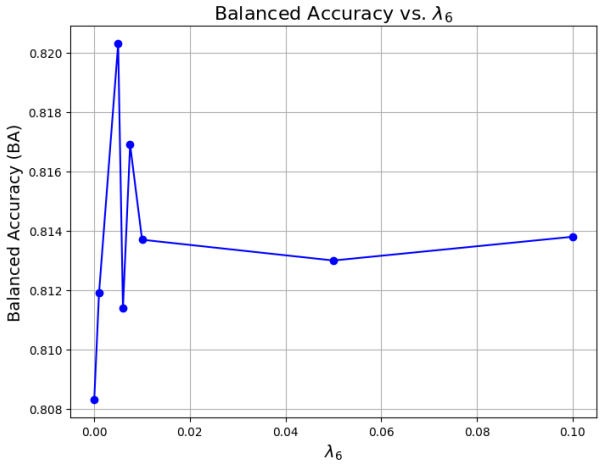


Figure 4.9: Selection of the L_{ICD} coefficient value, i.e., λ_6. The value was chosen by conducting an experiment with seven possible values for the $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ scenario, using the ResNet-18 CNN backbone with $D = 256$ and top-k set to 1. It is noteworthy that for $\lambda_6 = 0.005$, we achieved the highest BA value among the seven considered values: 0.1, 0.05, 0.01, 0.0075, 0.006, 0.005, and 0.001. As a result of this finding, $\lambda_6 = 0.005$ was also retained for the remaining CNN backbones.

Lastly, the black-boxes architectures, which serve as non-interpretable counterpart models for com-

parative analysis of results, were trained with the same batch sizes as mentioned earlier. They were trained for 100 epochs, with a learning rate of 1×10^{-3} for EfficientNet B3, while the others were trained with a learning rate of 1×10^{-4}. In all cases, stochastic gradient descent with a momentum of 0.9 was used.

4.5 Evaluation Metrics

The performance metrics considered to evaluate the different approaches with the different architectures were BA and recall, obtained through the confusion matrix. Additionally, in scenario $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$, as described in section 3.2.3, after training the model, in addition to evaluating the L_{ICD} value itself, we consider an additional metric to assess the intra-class diversity of the prototypes. The additional metric we consider is the normalized intra-class variance, denoted as V_{ICN}, see section 4.5.3.

4.5.1 Confusion Matrix

A confusion matrix, as the name suggests, is a matrix that allows us to observe and better understand the performance of a model. Given that we have K classes, the matrix will have dimensions $K \times K$, where each entry at position (i, j) indicates the number of samples classified as belonging to class j, knowing that their true class is class i. In other words, each row represents the true class, and each column represents the assigned class, see fig. 4.10.

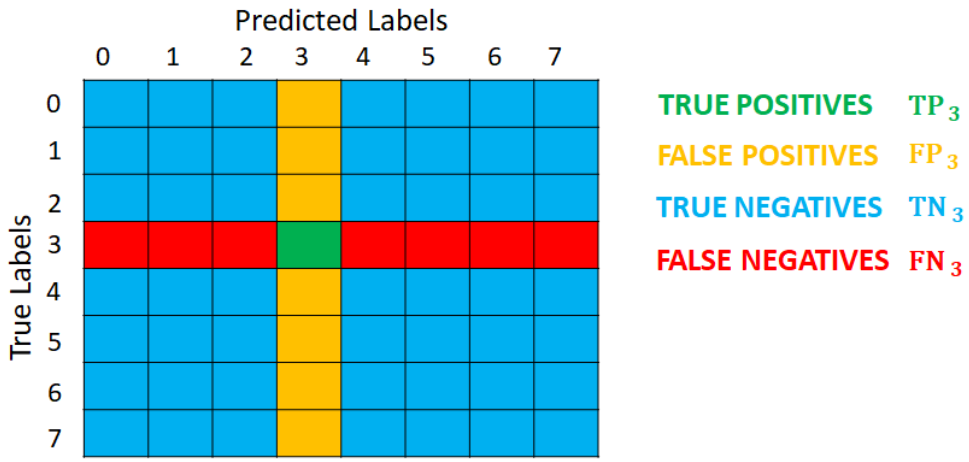


Figure 4.10: Example of a confusion matrix when considering 8 classes, $K = 8$. Additionally, it illustrates true positives TP_i, false positives FP_i, true negatives TN_i, and false negatives FN_i when analyzing the matrix using the one-versus-all strategy, considering class $i = 3$ as the positive class and the remaining classes as the negative ones.

By analyzing the confusion matrix for each class i, we can assess the true positives (TP_i), false positives (FP_i), true negatives (TN_i), and false negatives (FN_i), considering i as the positive class in a

strategy against the remaining classes, which are perceived as negative.

Recalling the meanings of TP_i, FP_i, TN_i, and FN_i, we have the following definitions. TP_i represents the instances that actually belong to class i, and the model correctly predicts them as the class i. FP_i represents the instances that do not actually belong to class i, and the model incorrectly predicts them as the class i. TN_i represents the instances that do not belong to class i, and the model accurately predicts them as a negative class. FN_i represents the instances that actually belong to class i, but the model incorrectly predicts them as a negative class.

4.5.2 Balanced Accuracy and Recall

Considering the datasets used, see section 4.1, both in the context of the binary problem and the multiclass problem, which are unbalanced, it becomes crucial to analyze the recall of each class i, where i can take values from 0 to $K-1$. It is important to note that $K=2$ in the binary problem and $K=8$ in the multiclass problem. For each class i, the recall R_i is calculated as follows

$$R_i = \frac{\mathsf{TP}_i}{\mathsf{TP}_i + \mathsf{FN}_i}. \tag{4.1}$$

Therefore, since BA is an average of the recall across the K classes, it is given by the following expression:

$$BA = \frac{1}{K}\sum_{i=0}^{K-1} R_i. \tag{4.2}$$

4.5.3 Normalized intra-class variance (V_{ICN})

In this metric, we view the prototypes of the same class (section 3.2) as a cluster and calculate the centroid represented by

$$c_p = \frac{\sum_{j=1}^{m} p_j}{m} \quad . \tag{4.3}$$

For each prototype p_j where j ranges from 1 to m, we calculate the squared Euclidean distance to the centroid, normalize it by the maximum value, and sum up the normalized squared distances. Finally, we divide the sum by the number of prototypes. Formally, considering the m squared distances relative to the centroid of the cluster represented by the set $\mathcal{D}^2_{P-c_p}$, the expression for the normalized intra-class variance is given by

$$V_{\mathsf{ICN}} = \frac{1}{m}\sum_{j=1}^{m} \frac{||p_j - c_p||^2}{\max(\mathcal{D}^2_{P-c_p})} \tag{4.4}$$

It is crucial to acknowledge that there might exist exceptional cases or circumstances in which all prototypes are equidistant from each other and from the centroid of the cluster they form. In such instances, a V_{ICN} value of 1 may not accurately capture the concept of diversity. Therefore, when assessing the

diversity of prototypes within the $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ scenario, it is essential to consider not only the value of V_{ICN} but also that of L_{ICD}.

Additionally, although it is not included in the main body of this thesis, it is interesting to observe that besides being able to assess measures that reflect the diversity among prototypes, we can also determine if there are diverse prototypes that hold a similar level of importance for decision-making. Attached, see table A.3, is an example that complements the results of section 5.2.2, illustrating the effect of removing a prototype from the model's decision process and the impact of this removal on BA performance in the validation set. If, when each prototype is individually removed, it has a similar impact on the model's performance compared to the others, we can conclude that prototypes generally carry a similar weight or importance in the decision-making process and similar data representativeness.

4.6 Computational Environment

The majority of experiments in this thesis were performed using a computer equipped with the high-performance NVIDIA GeForce RTX 3090 graphics card. However, for smaller-scale experiments, an Asus ROG Strix G512LW_G512LW laptop was utilized. This laptop features an Intel Core i7-10750H processor with a base clock speed of 2.60GHz and a maximum turbo frequency of 2.59GHz, accompanied by 16 GB of RAM. Additionally, the laptop is equipped with an NVIDIA GeForce RTX 2070 graphics card, which includes 8GB of GDDR6 RAM memory for enhanced performance in graphics-intensive tasks. Furthermore, it is crucial to note that the implementation was carried out using the Python programming language. The project heavily relied on popular libraries for DL, including PyTorch, Scikit-learn, and NumPy.

5

Experimental Results and Discussion

Contents

In section 5.1 and section 5.2, we present results related to section 3.1 and section 3.2. Section 5.3 examines prototype alignment with dermatological concepts, and in section 5.4, we assess explanations for scenarios in the binary problem using five guidelines.

5.1 Interpretable Skin Cancer Detection with Prototypes

5.1.1 Binary Problem: Melanoma vs. Nevus.

5.1.1.A Performance and Generalization Results

In table 5.1, the results for the approach described in section 3.1 can be found, according to the settings mentioned in section 4.4. Results are presented for 5 different CNN architectures used as backbones in 4 distinct scenarios, along with the corresponding case of the black-box model.

Table 5.1: Results for Melanoma vs. Nevus using 5 CNN architectures as backbone in 4 scenarios. Black-box counterparts included for comparison. Interpretable scenarios: L_{P}, $L_{\text{P}} + L_{\text{M}}$, $L_{\text{P-Masked}}$, $L_{\text{P}} + L_{\text{R}}$. Best performance on ISIC 2019 validation set [1–3], with corresponding results on PH^2 [15] and Derm7pt [9] test sets for generalization evaluation. Metrics: BA and recall. Bold indicates the highest values for each architecture and metric.

Model	Scenario	Best Results ISIC 2019			Results PH^2			Results Derm7pt		
		BA	R-MEL	R-NV	BA	R-MEL	R-NV	BA	R-MEL	R-NV
RN-18	Black-Box	83.7	81.7	85.6	78.7	60.0	97.5	72.4	53.2	**91.6**
	L_{P}	**84.6**	79.9	89.3	81.6	67.5	95.6	75.8	68.2	83.3
	$L_{\text{P}}+L_{\text{M}}$	83.8	75.0	**92.6**	**83.1**	70.0	96.2	**78.2**	74.6	81.7
	$L_{\text{P}}+L_{\text{R}}$	82.7	80.0	85.5	74.7	50.0	**99.4**	69.8	51.2	88.3
	$L_{\text{P-Masked}}$	77.4	**82.2**	72.7	79.4	**87.5**	71.2	70.8	**85.7**	55.8
RN-50	Black-Box	**85.1**	**83.0**	87.2	76.2	55.0	97.5	73.2	59.1	87.3
	L_{P}	83.9	80.3	87.5	**85.0**	**70.0**	**100.0**	70.8	57.9	83.6
	$L_{\text{P}}+L_{\text{M}}$	83.9	77.3	**90.4**	80.6	62.5	98.7	72.5	61.5	83.5
	$L_{\text{P}}+L_{\text{R}}$	83.1	82.8	83.3	76.6	62.5	90.6	**73.6**	**66.7**	**80.5**
	$L_{\text{P-Masked}}$	81.2	74.3	88.0	70.3	60.0	80.6	66.4	53.2	79.6
EN-B3	Black-Box	**87.0**	82.0	92.1	80.6	65.0	96.2	76.3	60.7	91.8
	L_{P}	86.5	**84.6**	88.3	78.1	57.5	98.7	76.2	62.7	89.7
	$L_{\text{P}}+L_{\text{M}}$	86.0	83.4	88.4	**89.4**	**80.0**	**98.7**	76.8	61.5	**92.2**
	$L_{\text{P}}+L_{\text{R}}$	84.8	82.1	87.5	85.0	72.5	97.5	**78.2**	**74.2**	82.1
	$L_{\text{P-Masked}}$	84.3	76.0	**92.6**	82.5	67.5	97.5	77.9	73.8	82.1
DN-169	Black-Box	**86.1**	83.6	88.5	76.6	55.0	98.1	76.5	66.7	86.6
	L_{P}	85.8	**84.1**	87.6	81.2	65.0	97.5	76.8	65.1	**88.5**
	$L_{\text{P}}+L_{\text{M}}$	84.7	78.0	91.4	**84.1**	**70.0**	**98.1**	74.5	61.9	87.1
	$L_{\text{P}}+L_{\text{R}}$	85.2	83.6	86.9	77.8	57.5	98.1	73.2	63.1	83.3
	$L_{\text{P-Masked}}$	84.2	77.0	**91.5**	80.3	67.5	93.1	**77.1**	**75.4**	78.8
VGG-16	Black-Box	**84.0**	**82.8**	**85.2**	76.9	57.5	96.2	70.0	54.0	86.1
	L_{P}	80.9	77.2	84.7	65.6	32.5	**98.7**	71.3	54.0	**88.7**
	$L_{\text{P}}+L_{\text{M}}$	80.5	78.5	82.6	**81.6**	**65.0**	98.1	**77.4**	**75.8**	79.0
	$L_{\text{P}}+L_{\text{R}}$	81.7	82.4	81.0	78.7	60.0	97.5	72.6	64.7	80.5
	$L_{\text{P-Masked}}$	82.8	81.2	84.4	78.8	62.5	95.0	70.8	72.6	69.0

Results on the validation set: It can be easily observed that the black-box model achieved the highest BA value in 4 out of 5 architectures on the validation set, indicating that the interpretable model struggled to compete with the black-box approach. However, considering the 4 interpretable scenarios, it is evident that L_{P} exhibited the best performance in 3 out of 5 architectures. Moreover, $L_{\text{P}} + L_{\text{M}}$ outperformed $L_{\text{P}} + L_{\text{R}}$ in 3 out of 5 architectures. On the other hand, $L_{\text{P}} + L_{\text{R}}$ showed better performance than $L_{\text{P-Masked}}$ in 4 out of 5 architectures.

In terms of R-MEL, in 3 out of 5 architectures, the interpretable scenarios allowed for higher values. It is crucial to remember that melanoma is a malignant class. However, out of these 3 instances, only 1 was related to a scenario with prototype-level supervision. This may imply that restricting the model to focus on the interior of the lesion and disregard confounding factors outside the boundary hampers its ability to detect artifacts such as hair or rulers. In the unsupervised model, which is less constrained, detecting and addressing these artifacts leads to an increase in both BA and R-MEL. Nevertheless, the primary goal is not for the model to provide a diagnosis with an emphasis on these artifacts, as that would result in reduced confidence and trust in the model's predictions. Therefore, there is a trade-off between performance on the validation set and ensuring a more reliable diagnosis that relies on factors of potentially greater clinical relevance present within the lesion. If we only consider the interpretable scenarios with prototype-level supervision ($L_{\text{P}} + L_{\text{M}}$, $L_{\text{P}} + L_{\text{R}}$, $L_{\text{P-Masked}}$), in 3 out of the 5 architectures, $L_{\text{P}} + L_{\text{R}}$ was the technique that yielded the highest performance in terms of R-MEL. On the other hand, it was $L_{\text{P}} + L_{\text{M}}$ that achieved the highest BA value in 3 out of the 5 architectures.

Results on the PH^2 test set: In contrast to what was observed in the validation set, it is evident that the interpretable scenarios led to higher BA values when compared to their respective black-box counterparts. Specifically, in 4 out of the 5 architectures, the $L_{\text{P}} + L_{\text{M}}$ scenario achieved the highest BA value. Additionally, both $L_{\text{P}} + L_{\text{R}}$ and $L_{\text{P-Masked}}$ outperformed the black-box in terms of BA in 4 out of the 5 architectures. Furthermore, in the interpretable but without prototype-level supervision, L_{P}, it also outperformed the black-box in 3 out of the 5 architectures. Consequently, we can deduce that the interpretable scenarios contributed to superior generalization on the PH^2 test set compared to the black-boxes. Focusing on R-MEL, the L_{P} scenario outperformed the black-box in 3 out of the 5 architectures. The $L_{\text{P}} + L_{\text{M}}$ scenario consistently surpassed the black-box in all architectures, as did the $L_{\text{P-Masked}}$ scenario. The $L_{\text{P}} + L_{\text{R}}$ scenario showed this superiority in 4 out of the 5 architectures. Furthermore, the highest R-MEL value belongs to $L_{\text{P}} + L_{\text{M}}$ in 3 out of the 5 architectures. Also, $L_{\text{P-Masked}}$ outperformed $L_{\text{P}} + L_{\text{R}}$ in terms of R-MEL in 3 out of the 5 architectures.

Results on the Derm7pt test set: In all architectures, the scenario that led to the highest BA value was consistently an interpretable scenario with prototype-level supervision. Out of the 5 architectures, the best scenario was $L_{\text{P}} + L_{\text{M}}$ in 2 of them, $L_{\text{P}} + L_{\text{R}}$ in another 2, and only in 1 architecture was $L_{\text{P-Masked}}$ considered the most effective. Focusing on BA results, all interpretable scenarios (L_{P}, $L_{\text{P-Masked}}$, $L_{\text{P}} + L_{\text{M}}$,

and $L_{\text{P}} + L_{\text{R}}$), outperformed the black-box method in 3 out of the 5 architectures.

By considering the R-MEL value, it becomes evident that an interpretable scenario with prototype-level supervision consistently emerged as the best option across all architectures. Out of the 5 architectures, the highest value was observed in the $L_{\text{P-Masked}}$ scenario in 2 of them. In another 2 architectures, the best performing scenario was $L_{\text{P}} + L_{\text{R}}$, and in only 1 architecture was it $L_{\text{P}} + L_{\text{M}}$. In the 5 architectures, L_{P} and $L_{\text{P}} + L_{\text{R}}$ outperform the black-box in 3 out of 5, while $L_{\text{P}} + L_{\text{M}}$ and $L_{\text{P-Masked}}$ outperform the black-box in 4 out of 5.

In summary, interpretable scenarios exhibited superior performance on test sets compared to their respective black-box models, despite struggling to compete on the validation set. The use of prototype-level supervision played a crucial role in this improvement. By focusing on relevant areas within the lesion boundaries and excluding less pertinent factors, the learned prototypes became potentially more clinically meaningful. Particularly, the interpretable scenario utilizing prototype-level supervision, denoted as $L_{\text{P}} + L_{\text{M}}$, demonstrated consistently superior generalization in the context of skin cancer diagnosis.

5.1.1.B Observing Prototypes and Explanation

In fig. 5.1 we can observe the explanation provided by the interpretable model, taking into account the three most similar prototypes to the images intended for diagnosis. The most similar prototype corresponds to a melanoma prototype, while the other two are nevus prototypes. However, the melanoma prototype carries more significance in the decision-making process.

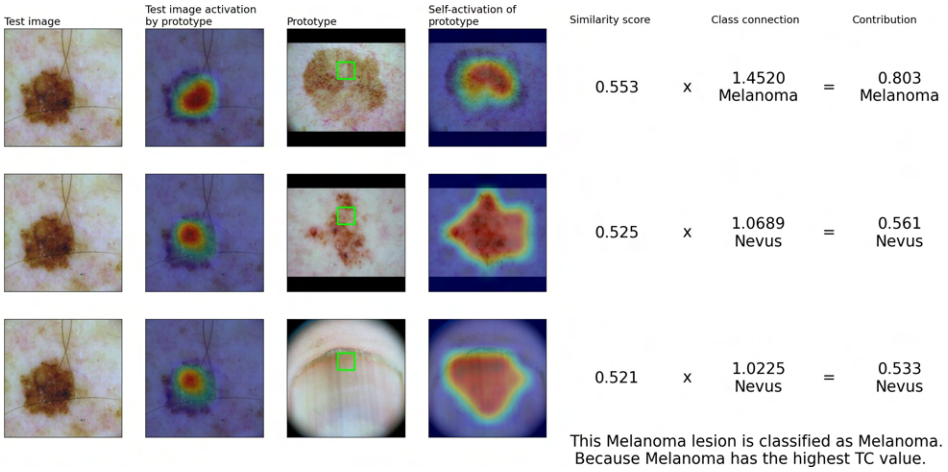


Figure 5.1: Example of an explanation provided by the interpretable model in the scenario $L_{\text{P}} + L_{\text{M}}$ for the EN-B3 architecture. Notice how the lesion, belonging to the ISIC 2019 validation set [1–3], is of the melanoma class and is correctly classified. Upon observing the three most activated prototypes, the prototype with the highest similarity corresponds to a melanoma, whereas the other two prototypes belong to the nevus class. Although only 3 out of the 18 prototypes are displayed, the total contribution (TC) of the melanoma class exceeds that of the nevus class, hence it is correctly classified.

Although all 18 prototypes are not displayed for the sake of simplicity, ultimately, the resemblance to

the melanoma prototypes holds a higher contribution, leading to its classification within that class. On the other hand, in the appendix, see fig. A.7, we can observe a case of melanoma that was misclassified, exhibiting a greater resemblance to nevus prototypes rather than melanoma prototypes.

In fig. 5.2, we can observe the prototypes obtained in 4 interpretable scenarios for EN-B3 architecture. Notice how in scenario L_{P} without prototype-level supervision, many of the prototypes did not remain within the lesion and ended up representing artifacts and regions with potentially less clinical relevance outside the lesion's boundaries. In contrast, in the interpretable scenarios with prototype-level supervision, they are representative of the interior of the skin lesion and likely more significant for diagnosis.

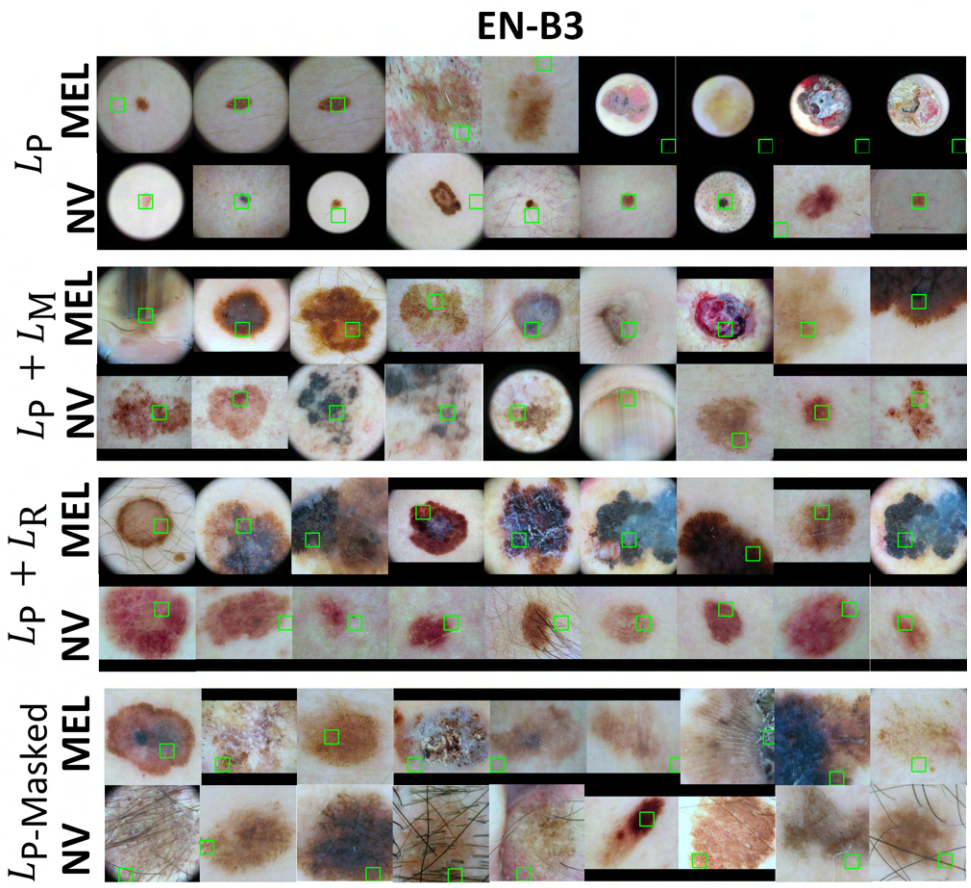


Figure 5.2: 18 prototypes obtained for each interpretable scenario using the EN-B3 architecture. For each scenario, we have 2 sets of prototypes: the first set (top row) corresponds to melanoma prototypes, and the second set (bottom row) corresponds to nevus class prototypes. Without prototype-level supervision L_{P}, many of the obtained prototypes are related to areas with black borders or areas of skin outside the lesion boundary, which potentially may not hold significant clinical relevance for diagnosis. On the other hand, in the interpretable scenarios with non-expert prototype-level supervision ($L_{\text{P}} + L_{\text{M}}$, $L_{\text{P}} + L_{\text{R}}$, $L_{\text{P-Masked}}$), it allows the prototypes to be restricted within the lesion boundary.

Additionally, for the different architectures, in scenario L_{P}, not all prototypes were considered valid,

meaning they were not representative of the interior or the boundary of the lesion, as described in section 4.3. In scenario L_{P}, for all architectures, the number of valid prototypes out of 18 did not exceed 11. However, in the scenario $L_{\text{P}} + L_{\text{R}}$, there were always a minimum of 16 out of 18 valid prototypes, and in the remaining scenarios $L_{\text{P}} + L_{\text{M}}$ and $L_{\text{P-Masked}}$, all 18 out of 18 prototypes were valid.

5.1.2 Multiclass Challenge: Classifying 8 Distinct Classes

5.1.2.A Performance and Generalization Results

In table 5.2, we can observe the results obtained using the approach described in section 3.1, according to the definitions provided in section 4.4. The results are presented for 5 different architectures, with 4 interpretable scenarios, and the presence of the black-box model for comparative purposes.

Table 5.2: Results for 5 CNN architectures used as backbone in 4 scenarios. Black-box counterparts included for comparison. Interpretable scenarios: L_{P}, $L_{\text{P}} + L_{\text{M}}$, $L_{\text{P-Masked}}$, $L_{\text{P}} + L_{\text{R}}$. Best performance on ISIC 2019 validation set [1–3]. Metrics: BA and recall. Bold indicates the highest values for each architecture and metric. Considering the multiclass problem, employing the first approach outlined in section 3.1.

Model	Scenario	Best Results ISIC 2019								
		BA	**R-AK**	**R-BCC**	**R-BKL**	**R-DF**	**R-MEL**	**R-NV**	**R-SCC**	**R-VASC**
RN-18	Black-Box	**70.9**	**68.2**	**78.2**	**57.9**	**78.6**	**62.7**	78.6	**62.7**	80.4
	L_{P}	61.8	51.4	61.7	38.5	83.3	57.5	71.6	47.6	82.3
	$L_{\text{P}}+L_{\text{M}}$	59.2	52.0	71.5	35.6	66.7	41.8	**79.8**	39.7	**86.3**
	$L_{\text{P}}+L_{\text{R}}$	53.0	64.2	34.0	47.2	75.0	48.0	74.2	7.1	74.5
	$L_{\text{P-Masked}}$	53.7	43.9	56.7	35.4	68.7	48.4	52.8	37.3	86.3
RN-50	Black-Box	**74.0**	67.6	**83.1**	**67.8**	79.2	**69.2**	**82.8**	**59.5**	82.3
	L_{P}	61.9	48.5	54.7	44.9	**81.2**	55.4	74.2	54.0	82.3
	$L_{\text{P}}+L_{\text{M}}$	56.8	46.2	62.0	37.3	62.5	45.8	79.8	42.1	78.4
	$L_{\text{P}}+L_{\text{R}}$	62.4	**67.6**	53.5	47.0	71.0	60.6	67.5	47.6	**84.3**
	$L_{\text{P-Masked}}$	52.1	53.2	45.5	28.9	72.9	48.7	55.8	27.8	84.3
EN-B3	Black-Box	**76.6**	**68.8**	**85.7**	**71.8**	**77.1**	**75.4**	81.7	64.3	88.2
	L_{P}	72.6	63.0	77.1	60.2	77.1	69.4	79.4	**66.7**	88.2
	$L_{\text{P}}+L_{\text{M}}$	64.8	52.6	70.8	48.2	62.5	54.6	**81.7**	61.9	86.3
	$L_{\text{P}}+L_{\text{R}}$	56.2	23.1	45.9	49.3	70.8	55.7	81.0	37.3	86.3
	$L_{\text{P-Masked}}$	60.0	52.0	58.6	44.6	64.6	59.7	57.7	50.8	**92.2**
DN-169	Black-Box	**74.8**	**61.4**	**85.5**	**71.6**	**72.9**	**70.7**	**84.1**	**65.9**	**86.3**
	L_{P}	63.3	54.3	59.5	51.8	72.9	59.7	74.2	50.0	84.3
	$L_{\text{P}}+L_{\text{M}}$	61.2	53.7	64.3	47.4	68.7	46.9	82.0	48.4	78.4
	$L_{\text{P}}+L_{\text{R}}$	62.9	53.7	63.1	50.7	62.5	58.2	75.8	55.6	84.3
	$L_{\text{P-Masked}}$	50.9	43.9	53.8	29.5	64.6	49.1	51.7	35.7	78.4
VGG-16	Black-Box	**72.6**	**71.1**	**77.1**	**59.2**	**72.9**	**63.0**	**75.5**	**73.8**	88.2
	L_{P}	55.9	50.3	58.6	40.0	66.7	49.3	71.9	30.2	80.4
	$L_{\text{P}}+L_{\text{M}}$	56.0	47.4	64.0	38.5	66.7	45.7	73.6	31.7	80.4
	$L_{\text{P}}+L_{\text{R}}$	59.9	62.4	64.0	38.1	70.8	55.7	71.6	28.6	**88.2**
	$L_{\text{P-Masked}}$	52.7	50.3	50.7	32.8	60.4	43.9	69.4	35.7	78.4

Results on the validation set: It is evident that the interpretable models face significant challenges

in competing with the non-interpretable models in terms of performance on the validation set. Across all architectures, the best performance consistently belonged to the black-box model, both in terms of overall BA and individual recall for each class. Additionally, focusing on the 4 interpretable scenarios, the best outcome was achieved in 3 out of 5 architectures using the scenario without prototype-level supervision, L_{P}. In the remaining 2 architectures, the optimal performance was observed in the scenario with prototype-level supervision, incorporating human feedback, $L_{\text{P}} + L_{\text{R}}$. Considering only scenarios with prototype-level supervision, in 3 out of the 5 architectures, the best-performing scenario was $L_{\text{P}} + L_{\text{R}}$. However, in the remaining 2, the best scenario was $L_{\text{P}} + L_{\text{M}}$.

In table 5.3, we can observe the results of the models trained to classify the 8 classes and how they perform in a benign vs. malignant analysis. The results pertaining to ISIC 2019 complement and align with those presented in table 5.2. In other words, besides being able to observe an individual analysis concerning the classification of the 8 classes, we can also assess the model's ability to, even when making errors, assign a label belonging to the same malignancy group as the ground truth label. The malignant classes consist of AK (pre-malignant), BCC, MEL, and SCC, while the remaining classes are benign. From the standpoint of the malignancy vs. benign analysis, the black-box model continues to perform the best on the ISIC 2019 validation set. If we solely focus on interpretable scenarios, the scenario L_{P} is the optimal choice in 4 out of the 5 architectures. However, when considering only interpretable scenarios with non-expert prototype-level supervision, the $L_{\text{P}} + L_{\text{R}}$ scenario is the best option in 3 out of the 5 architectures, while the $L_{\text{P}} + L_{\text{M}}$ scenario outperforms in the remaining two, in terms of the value of BA-BM.

Analyzing only the value of R-M, we observe that, considering only interpretable scenarios, $L_{\text{P}} + L_{\text{R}}$ is the best in 3 out of the 5 architectures for better identification of malignant classes. If we focus solely on interpretable scenarios with supervision on prototypes ($L_{\text{P}} + L_{\text{M}}$, $L_{\text{P}} + L_{\text{R}}$, and $L_{\text{P-Masked}}$), the scenario with human feedback, $L_{\text{P}} + L_{\text{R}}$, remains the best in 3 out of the 5 architectures, while in the other two, $L_{\text{P}} + L_{\text{M}}$ outperforms.

Results on the PH^2 test set: Analyzing the results from table 5.3 in terms of BA, in 3 out of the 5 architectures, the best-performing scenario was always an interpretable one. Moreover, if we only consider scenarios with prototype-level supervision, $L_{\text{P}} + L_{\text{R}}$, proved to be the best in 3 out of the 5 architectures. Focusing on R-MEL, in 4 out of the 5 architectures, the best result was consistently achieved with an interpretable scenario. If we consider only scenarios with supervision on the prototypes, both $L_{\text{P}} + L_{\text{R}}$ and $L_{\text{P-Masked}}$ demonstrated superior performance in 2 out of the 5 architectures, while $L_{\text{P}} + L_{\text{M}}$ was the best in only 1 of them.

Considering the malignancy vs. benign analysis, in 3 out of the 5 architectures, the best-performing scenario was always an interpretable one, based on the value of BA-BM. Furthermore, if we focus on scenarios with supervision on the prototypes, both $L_{\text{P}} + L_{\text{R}}$ and $L_{\text{P-Masked}}$ demonstrated superior

performance in 2 out of the 5 architectures, while $L_{\text{P}} + L_{\text{M}}$ was the best in only 1 of them. Regarding the R-M metric for malignant classes, the results and conclusions are identical to those of R-MEL.

Table 5.3: Results for 5 CNN architectures used as backbone in 4 scenarios. Black-box counterparts included for comparison. Interpretable scenarios: L_{P}, $L_{\text{P}} + L_{\text{M}}$, $L_{\text{P-Masked}}$, $L_{\text{P}} + L_{\text{R}}$. Metrics: BA and recall. Bold indicates the highest values for each architecture and metric. Considering the multiclass problem, employing the first approach outlined in section 3.1. BA-BM represents the BA considering the individual recalls for the set of benign classes (R-B) and the set of malignant classes (R-M). Performance on the PH^2 [15] test set in terms of melanoma vs. nevus analysis. Since the model is trained on 8 classes but tested on a dataset with only two classes, some of the assigned labels may refer to classes that do not exist in the test set. Therefore, the analysis of benign vs. malignant is also presented. Performance on ISIC 2019 validation set [1–3], in terms of benign vs. malignant based on the results from table 5.2.

Model	Scenario	Results PH^2						Results ISIC 2019		
		BA	R-MEL	R-NV	BA-BM	R-B	R-M	BA-BM	R-B	R-M
RN-18	Black-Box	68.7	40.0	**97.5**	69.4	**98.7**	40.0	**82.9**	**83.5**	**82.3**
	L_{P}	65.9	40.0	91.9	68.7	97.5	40.0	77.2	75.2	79.3
	L_{P}+L_{M}	72.2	47.5	96.9	72.5	97.5	47.5	76.4	83.3	69.4
	L_{P}+L_{R}	**78.1**	**67.5**	88.7	**79.1**	90.6	**67.5**	74.7	80.4	69.1
	$L_{\text{P-Masked}}$	64.1	60.0	68.1	72.2	84.4	60.0	74.4	78.2	70.5
RN-50	Black-Box	72.2	45.0	**99.4**	72.2	**99.4**	45.0	**84.7**	**86.6**	**82.8**
	L_{P}	**75.9**	**55.0**	96.9	**75.9**	96.9	**55.0**	78.7	81.6	75.8
	L_{P}+L_{M}	71.6	45.0	98.1	71.6	98.1	45.0	77.5	82.4	72.7
	L_{P}+L_{R}	64.1	30.0	98.1	64.1	98.1	30.0	77.9	75.5	80.2
	$L_{\text{P-Masked}}$	70.3	42.5	98.1	70.3	98.1	42.5	73.0	77.6	68.3
EN-B3	Black-Box	**81.9**	70.0	93.7	**82.2**	94.4	70.0	**86.8**	**86.5**	**87.0**
	L_{P}	75.6	55.0	**96.2**	76.2	97.5	55.0	84.2	83.6	84.9
	L_{P}+L_{M}	77.2	60.0	94.4	79.4	**98.7**	60.0	80.7	84.6	76.9
	L_{P}+L_{R}	79.1	65.0	93.1	81.2	97.5	65.0	78.2	84.9	71.4
	$L_{\text{P-Masked}}$	71.9	**75.0**	68.7	81.9	88.7	**75.0**	77.6	81.7	73.5
DN-169	Black-Box	72.2	45.0	**99.4**	72.2	**99.4**	45.0	**86.0**	**88.0**	**84.0**
	L_{P}	75.3	60.0	90.6	76.2	92.5	60.0	80.3	81.2	79.4
	L_{P}+L_{M}	**78.1**	57.5	98.7	78.1	98.7	57.5	78.0	85.4	70.6
	L_{P}+L_{R}	77.2	67.5	86.9	77.2	86.9	67.5	80.7	80.5	80.9
	$L_{\text{P-Masked}}$	74.7	**80.0**	69.4	**81.9**	83.7	**80.0**	72.6	76.1	69.1
VGG-16	Black-Box	**82.2**	**70.0**	94.4	**83.4**	96.9	**70.0**	**82.7**	**81.8**	**83.6**
	L_{P}	78.7	60.0	97.5	79.1	98.1	60.0	76.4	80.0	72.8
	L_{P}+L_{M}	76.2	57.5	95.0	76.6	95.6	57.5	75.9	81.2	70.6
	L_{P}+L_{R}	76.9	60.0	93.7	77.2	94.4	60.0	76.3	75.8	76.8
	$L_{\text{P-Masked}}$	66.9	35.0	**98.7**	66.9	**98.7**	35.0	74.4	79.2	69.5

Results on the Derm7pt test set: Before conducting a detailed analysis, it is important to first note the fact that regardless of whether the model is interpretable or not, none of the cases from table 5.4 yield a BA value above 50%. This outcome highlights that classifying individual skin lesion classes in a multiclass problem remains a challenging and current issue, particularly when confronted with a test dataset that significantly differs from the training and validation data in terms of distribution. It is worth emphasizing that the Derm7pt dataset only shares 6 common classes with ISIC 2019.

Table 5.4: Results for 5 CNN architectures used as backbone in 4 scenarios. Black-box counterparts included for comparison. Interpretable scenarios: L_{P}, $L_{\text{P}}+L_{\text{M}}$, $L_{\text{P-Masked}}$, $L_{\text{P}}+L_{\text{R}}$. Results on the test set Derm7pt [9] for the first approach (section 3.1) and the multiclass problem. Metrics: BA and recall. Bold indicates the highest values for each architecture and metric. BA-BM represents the BA considering the individual recalls for the set of benign classes (R-B) and the set of malignant classes (R-M).

M	Scenario	Results Derm7pt									
		BA	**R-BCC**	**R-BKL**	**R-DF**	**R-MEL**	**R-NV**	**R-VASC**	**BA-BM**	**R-B**	**R-M**
RN-18	Black-Box	**39.7**	**4.7**	**35.6**	**40.0**	55.2	**88.7**	13.8	71.5	**89.2**	53.7
	L_{P}	29.4	2.4	4.4	10.0	61.1	74.4	24.1	66.2	74.3	58.2
	$L_{\text{P}}+L_{\text{M}}$	36.1	0.0	17.8	30.0	69.0	79.3	20.7	**71.5**	79.1	63.9
	$L_{\text{P}}+L_{\text{R}}$	33.2	0.0	15.6	20.0	**70.6**	75.8	17.2	70.8	75.6	**66.0**
	$L_{\text{P-Masked}}$	34.9	2.4	24.4	30.0	50.8	43.0	**58.6**	62.0	77.0	46.9
RN-50	Black-Box	40.3	2.4	**42.2**	50.0	51.2	**89.2**	6.9	68.9	**89.5**	48.3
	L_{P}	34.7	0.0	20.0	30.0	55.9	84.9	17.2	69.6	85.0	54.1
	$L_{\text{P}}+L_{\text{M}}$	**44.4**	**7.1**	24.4	**70.0**	**62.3**	81.7	20.7	**71.0**	83.1	**58.8**
	$L_{\text{P}}+L_{\text{R}}$	30.5	0.0	33.3	0.0	57.9	77.9	13.8	69.2	82.4	56.1
	$L_{\text{P-Masked}}$	34.2	2.4	8.9	30.0	45.6	56.2	**62.1**	61.6	80.7	42.5
EN-B3	Black-Box	45.1	**9.52**	42.2	45.0	**70.6**	79.3	24.1	**74.2**	81.5	**67.0**
	L_{P}	46.1	4.8	40.0	**65.0**	53.6	88.9	24.1	71.0	90.7	51.4
	$L_{\text{P}}+L_{\text{M}}$	**46.5**	7.1	**44.4**	60.0	46.8	**93.2**	27.6	70.5	**94.2**	46.9
	$L_{\text{P}}+L_{\text{R}}$	37.3	2.4	33.3	35.0	39.3	89.9	24.1	66.9	93.3	40.5
	$L_{\text{P-Masked}}$	39.4	2.4	31.1	40.0	45.6	58.5	**58.6**	65.7	87.3	44.2
DN-169	Black-Box	34.4	**14.3**	33.3	25.0	38.9	**91.6**	3.4	65.3	**92.2**	38.4
	L_{P}	37.8	0.0	**53.3**	35.0	54.8	73.6	10.3	66.2	81.5	51.0
	$L_{\text{P}}+L_{\text{M}}$	**46.4**	7.1	42.2	**65.0**	**65.5**	81.4	17.2	**72.6**	83.7	**61.6**
	$L_{\text{P}}+L_{\text{R}}$	28.2	0.0	22.2	0.0	57.9	85.7	3.4	71.4	84.3	58.5
	$L_{\text{P-Masked}}$	30.2	9.5	4.4	20.0	57.9	44.8	**44.8**	64.0	72.5	55.4
VGG-16	Black-Box	**44.5**	**11.9**	**55.6**	40.0	59.9	78.8	20.7	**71.5**	84.4	58.5
	L_{P}	26.0	0.0	20.0	0.0	51.2	**81.6**	3.5	66.3	83.3	49.3
	$L_{\text{P}}+L_{\text{M}}$	42.2	2.3	24.4	**55.0**	**67.9**	72.5	31.0	69.0	74.7	**63.3**
	$L_{\text{P}}+L_{\text{R}}$	26.0	9.5	11.1	0.0	53.2	75.3	6.9	65.6	77.4	53.7
	$L_{\text{P-Masked}}$	31.1	2.4	13.3	10.0	42.9	80.2	**37.9**	62.9	**85.3**	40.5

Starting with the analysis of BA, in 3 out of the 5 architectures, the best-performing scenario was the interpretable one with supervision on the prototypes, $L_{\text{P}}+L_{\text{M}}$. If we limit the analysis to only interpretable scenarios, this result holds true for all architectures. Furthermore, in the R-MEL metric, the best scenario in 4 architectures was consistently an interpretable one with supervision on the prototypes. Specifically, in 3 out of the 4 architectures, the optimal scenario was $L_{\text{P}}+L_{\text{M}}$, and only in 1 architecture was it $L_{\text{P}}+L_{\text{R}}$, but with the highest value of 70.6%. Additionally, the supervised scenario $L_{\text{P-Masked}}$ was consistently the best in all architectures for the R-VASC metric.

In the malignancy vs. benign analysis, for the BA-RM metric, the best scenario in 3 out of the 5 architectures was $L_{\text{P}} + L_{\text{M}}$. This result held true for all architectures when considering only scenarios with prototype-level supervision. Regarding the recall for malignant classes, R-M, superior values were consistently achieved in 4 out of the 5 architectures for interpretable scenarios with prototype-level supervision, with $L_{\text{P}} + L_{\text{R}}$ being the best in 1 architecture, and $L_{\text{P}} + L_{\text{M}}$ in the remaining 3.

In conclusion, we can assert that the multiclass problem is considerably more challenging for the explored interpretable scenarios when compared to black-box models, as evaluated on the validation set. However, when assessing generalization on test sets with significantly different distributions from the training and validation data, interpretable scenarios generally exhibit better performance, particularly favoring those with supervision on the prototypes, especially when identifying malignant classes.

In particular, the $L_{\mathsf{P}} + L_{\mathsf{M}}$ scenario stands out as the preferred option based on the results from the Derm7pt test set. This dataset is considerably older, with different data quality and distribution compared to the other sets.

5.1.2.B Observing Prototypes and Explanation

In Figure 5.3, we can observe examples of prototypes obtained for the 8 classes, with 3 prototypes per class. A comparison between two scenarios can be made: one with unsupervised prototypes using L_{P}, and the other with supervised prototypes using masks with $L_{\mathsf{P}} + L_{\mathsf{M}}$.

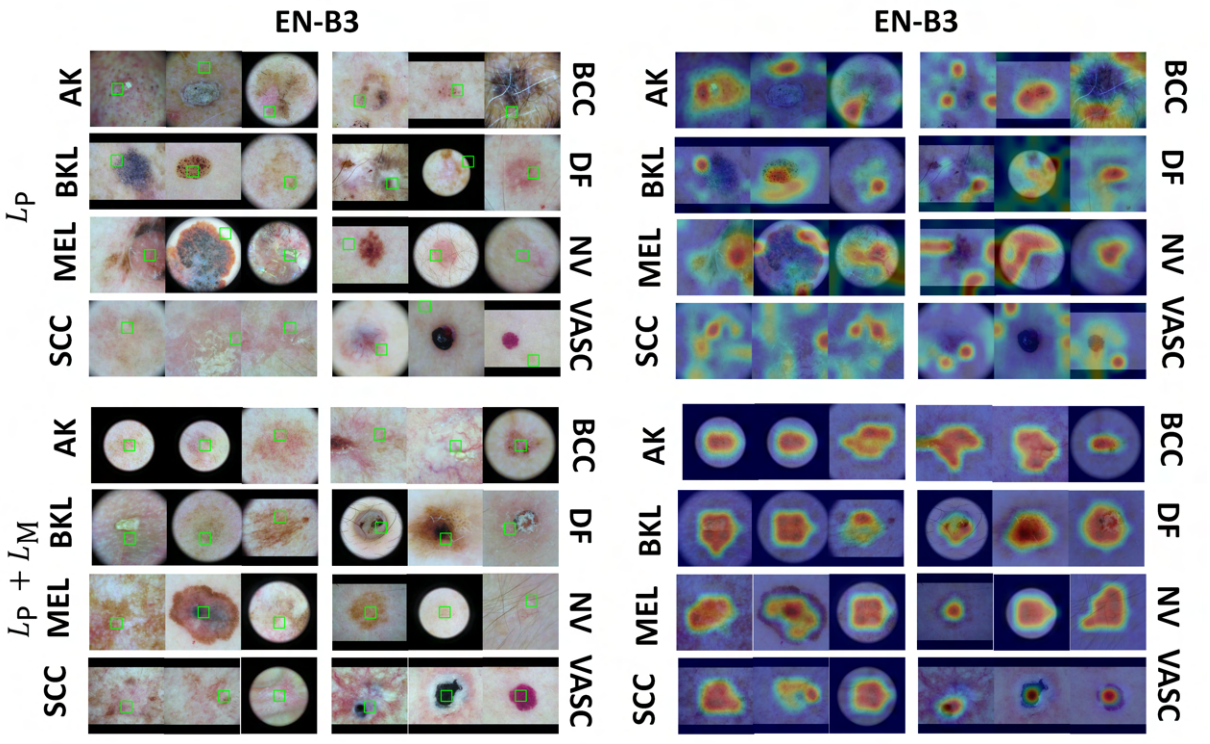


Figure 5.3: 24 prototypes for two interpretable scenarios, 3 per class, highlighted by a green square in the image (left), along with their respective activation maps on the image itself (right). In the upper section, we can observe the prototypes for the interpretable scenario without supervision, denoted as L_{P}, and in the lower section, for the interpretable scenario with prototype-level supervision, denoted as $L_{\mathsf{P}} + L_{\mathsf{M}}$. In the $L_{\mathsf{P}} + L_{\mathsf{M}}$ scenario, all the prototypes are consistently associated with the interior or boundary of the lesion, and their activations align accordingly. However, in the L_{P} scenario, this is not the case for all prototypes. In this case, the CNN backbone used was the EN-B3.

In the first scenario, similar to the binary problem, some prototypes are distant from the actual lesion, encompassing transitional zones with the background, thereby reducing their diagnostic relevance. Conversely, in the $L_{\mathsf{P}} + L_{\mathsf{M}}$ scenario, prototypes remain confined within the lesion boundaries, providing a more meaningful explanation and decision pathway. Furthermore, it is noteworthy to observe the difference in prototype activation within the image itself in both scenarios. In the case of L_{P}, the activation is not specifically focused and restricted to the interior of the lesion boundary, in contrast to the case of $L_{\mathsf{P}} + L_{\mathsf{M}}$. Importantly, we note that in all cases, the L_{P} scenario yielded no more than 16 out of 24 valid prototypes, while the $L_{\mathsf{P}} + L_{\mathsf{R}}$ scenario consistently ensured a minimum of 22 out of 24 valid prototypes. Additionally, in all other interpretable scenarios with prototype supervision, all 24 prototypes were consistently valid.

Moreover, in fig. 5.4, we can observe an example of the explanation provided in the scenario $L_{\mathsf{P}} + L_{\mathsf{M}}$ for a case belonging to the BCC class, which is correctly classified. By examining the five most similar prototypes, we note that the three most similar ones correspond to the same class as the image under diagnosis, while the remaining two prototypes belong to different classes. In other words, we can readily understand that the image is classified as BCC due to its greater similarity to BCC prototypes than to prototypes of other classes. It is also interesting to note that among the top 5 most activated prototypes, 4 belonged to malignant classes, considering that the AK class is pre-malignant and the NV class is benign. Finally, to access further examples of prototypes and diagnostic cases with explanations, please refer to appendix A.

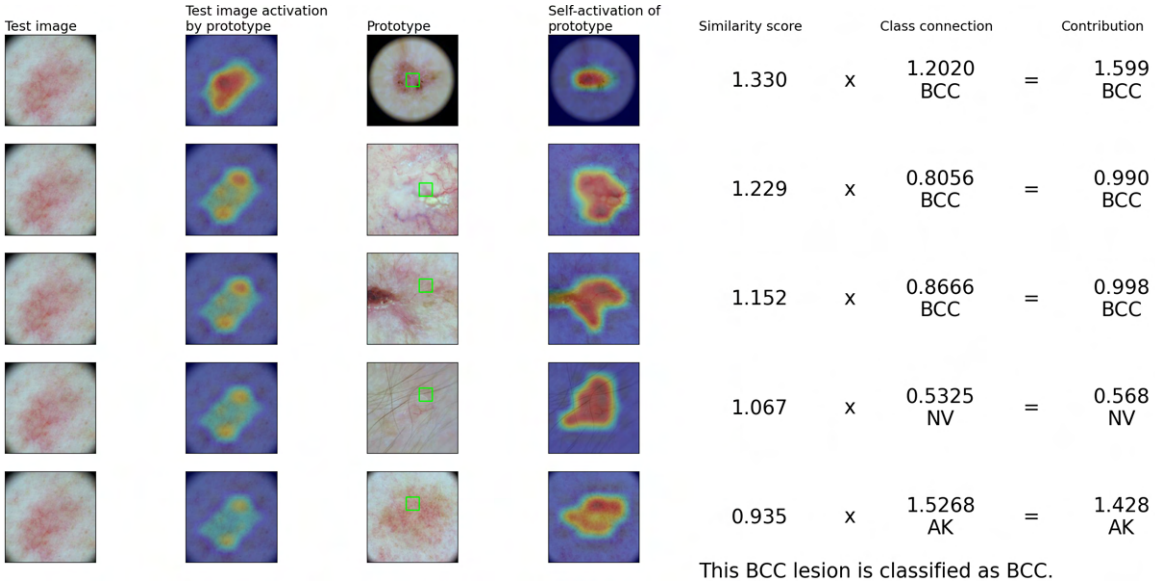


Figure 5.4: Example of the explanation provided by the interpretable model in the $L_{\mathsf{P}} + L_{\mathsf{M}}$ scenario for the EN-B3 architecture and for the multiclass problem. The lesion belonging to the ISIC 2019 validation set [1–3], is of the BCC class and is correctly classified. Upon observing the five most activated prototypes, the three prototypes with the highest similarity belong to the BCC class, whereas the other two prototype belongs to the NV and AK class, respectively.

5.2 One-Class Prototypes: Simplified Binary Problem Explanation

5.2.1 Performance and Generalization Results

5.2.1.A Results Analysis

In table 5.5, we can find the results obtained for the binary problem of Melanoma vs. Nevus using the second approach described in section 3.2. The configurations used correspond to those described section 4.4. In this approach, the model's structure is simplified to only include prototypes related to the malignant class, streamlining the decision-making process and consequently generating more interpretable explanations.

Table 5.5: Results for 5 CNN architectures used as backbone in 3 scenarios. Black-box counterparts included for comparison. Interpretable scenarios: $L_{\text{P-1C}}$, $L_{\text{P-1C-Masked}}$, $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$. Best performance on ISIC 2019 validation set [1–3], with corresponding results on PH^2 [15] and Derm7pt [9] test sets for generalization evaluation. Metrics: BA and recall. Bold indicates the highest values for each architecture and metric. Considering the binary problem of Melanoma vs. Nevus, employing the second approach outlined in section 3.2.

Model	Scenario	Best Results ISIC 2019			Results PH^2			Results Derm7pt		
		BA	R-MEL	R-NV	BA	R-MEL	R-NV	BA	R-MEL	R-NV
RN-18	Black-Box	**83.7**	**81.7**	85.6	78.7	60.0	**97.5**	72.4	53.2	**91.6**
	$L_{\text{P-1C}}$	82.9	79.2	86.6	79.4	67.5	91.2	**77.4**	**74.2**	80.7
	$L_{\text{P-1C-Masked}}$	81.1	74.4	87.8	**88.7**	**85.0**	92.5	67.5	50.4	84.7
	$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$	81.4	73.0	**89.9**	83.7	72.5	95.0	68.4	53.6	83.3
RN-50	Black-Box	**85.1**	**83.0**	87.2	76.2	55.0	97.5	73.2	59.1	**87.3**
	$L_{\text{P-1C}}$	78.0	74.6	81.4	69.1	40.0	98.1	**73.4**	**69.4**	77.4
	$L_{\text{P-1C-Masked}}$	81.6	72.4	**90.8**	79.4	60.0	**98.7**	68.3	63.1	73.6
	$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$	81.1	73.2	89.0	**82.5**	**70.0**	95.0	67.6	50.8	84.3
EN-B3	Black-Box	**87.0**	**82.0**	**92.1**	80.6	65.0	96.2	76.3	60.7	91.8
	$L_{\text{P-1C}}$	81.9	80.3	83.5	82.5	70.0	95.0	75.1	57.1	**93.0**
	$L_{\text{P-1C-Masked}}$	82.2	77.4	87.0	**83.4**	**70.0**	96.9	75.3	**77.8**	72.9
	$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$	82.7	75.2	90.2	81.2	65.0	**97.5**	**77.0**	68.6	85.4
DN-169	Black-Box	**86.1**	**83.6**	88.5	76.6	55.0	98.1	**76.5**	66.7	86.6
	$L_{\text{P-1C}}$	77.0	76.5	77.5	66.6	35.0	**98.1**	65.7	**76.6**	54.8
	$L_{\text{P-1C-Masked}}$	82.4	76.1	88.8	**86.9**	**80.0**	93.7	69.0	52.0	86.1
	$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$	82.4	74.3	**90.4**	83.1	72.5	93.7	69.5	51.6	**87.5**
VGG-16	Black-Box	**84.0**	**82.8**	**85.2**	76.9	57.5	96.2	70.0	54.0	86.1
	$L_{\text{P-1C}}$	78.0	77.5	78.4	69.7	40.0	**99.4**	71.9	**68.2**	75.6
	$L_{\text{P-1C-Masked}}$	81.3	79.9	82.7	71.2	50.0	92.5	66.4	61.5	71.3
	$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$	80.6	78.8	82.4	**79.4**	**60.0**	98.7	**72.7**	59.1	**86.3**

Results on the validation set: It is evident that in all architectures, the black-box model outperformed the interpretable scenarios in terms of BA. However, when we focus solely on the interpretable scenarios, we observe that $L_{\text{P-1C-Masked}}$ achieved the highest BA value in 3 out of the 5 architectures. Additionally, in 3 out of the 5 architectures, the scenario incorporating L_{ICD} either matched or surpassed the performance of $L_{\text{P-1C-Masked}}$. Furthermore, within the interpretable scenarios, in 4 out of the 5 architec-

tures, the superior performance is consistently observed in scenarios with prototype-level supervision. Considering only the R-MEL metric, the black-box model outperformed all architectures. When we focus solely on the interpretable scenarios, $L_{\text{P-1C}}$ was the best in 4 out of the 5 architectures, and $L_{\text{P-1C-Masked}}$ outperformed $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ in 4 out of the 5 architectures.

Results on the PH^2 test set: The best generalization in terms of BA consistently corresponds to an interpretable scenario with prototype-level supervision. Additionally, in 2 out of the 5 architectures, the optimal scenario involved promoting greater intra-class variability through the use of L_{ICD}. Furthermore, when analyzing the BA metric, the interpretable scenario without prototype-level supervision, $L_{\text{P-1C}}$, outperformed the black-box model in only 2 out of the 5 architectures. On the other hand, both the $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ scenarios consistently achieved better performance than the black-box model. Additionally, in 4 out of the 5 architectures, $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ outperformed the $L_{\text{P-1C}}$ scenario as well.

Focusing on the observation of the R-MEL metric, the best-performing scenario consistently turned out to be an interpretable one with prototype-level supervision. In 2 out of the 5 architectures, promoting intra-class diversity through L_{ICD} resulted in the best outcomes. The $L_{\text{P-1C}}$ scenario outperformed the black-box in only 2 out of the 5 architectures. Moreover, $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}}+L_{\text{ICD}}$, surpassed or matched the performance of the black-box models in 4 out of the 5 architectures, and in all architectures, respectively. Furthermore, the scenario $L_{\text{P-1C-Masked}}$, consistently outperforms or matches the scenario, $L_{\text{P-1C}}$, in all architectures. Similarly, in the scenario where L_{ICD} is introduced, the same pattern holds true in 4 out of the 5 architectures.

Results on the Derm7pt test set: Starting by evaluating the results based on the BA metric, in 4 out of the 5 architectures, the best-performing scenario was always an interpretable one, with 2 of them involving prototype-level supervision and the promotion of greater intra-class diversity. Moreover, the $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ scenario outperformed $L_{\text{P-1C-Masked}}$ in 4 out of the 5 architectures.

Analyzing the R-MEL metric, we observe that the best result consistently arises from an interpretable scenario across all 5 architectures. Additionally, the $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ scenarios outperform the black-box models in 3 out of the 5 architectures. Furthermore, in 4 out of the 5 architectures, $L_{\text{P-1C-Masked}}$ exhibits superior R-MEL performance compared to $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$.

In conclusion, similar to the first approach, the interpretable model faced greater challenges in competing with the black-box models on the validation set. However, on the test sets, the interpretable scenarios demonstrated better generalization. Particularly, in the PH^2 test set, the introduction of prototype supervision consistently yielded the best results.

5.2.1.B Comparing with the First Approach using Prototypes in Both Classes

To compare the two approaches in the binary problem, we based our selection on the performance of BA on the ISIC 2019 validation set [1–3]. For this purpose, the best results considering only scenarios with prototype-level supervision were chosen for each approach.

In the first approach (section 3.1), prototypes were created for both classes, while in the second approach (section 3.2), prototypes were only created for the malignant class. For the first approach, the best-performing scenario was $L_{\text{P}} + L_{\text{M}}$, and for the second approach, it was the scenario $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$. In both cases, these corresponded to the EfficientNet B3 architecture. In table 5.6, we can observe the results obtained from table 5.1 and table 5.5, along with the black-box model present for comparative purposes.

Table 5.6: Top results with prototype-level supervision for the binary problem for the first (section 3.1) and second (section 3.2) approaches in the ISIC 2019 validation set [1–3], along with their respective outcomes on the PH^2 [15] and Derm7pt [9] test sets.

Model	Scenario	Best Results ISIC 2019			Results PH^2			Results Derm7pt		
		BA	R-MEL	R-NV	BA	R-MEL	R-NV	BA	R-MEL	R-NV
EN-B3	Black-Box	**87.0**	82.0	**92.1**	80.6	65.0	96.2	76.3	60.7	91.8
	$L_{\text{P}} + L_{\text{M}}$	86.0	**83.4**	88.4	**89.4**	**80.0**	**98.7**	76.8	61.5	**92.2**
	$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$	82.7	75.2	90.2	81.2	65.0	97.5	**77.0**	**68.6**	85.4

It is evident that, from the perspective of the validation set, the best interpretable approach in terms of BA and R-MEL was the first one, corresponding to the scenario $L_{\text{P}} + L_{\text{M}}$. The same trend was observed for the PH^2 test set. However, for the Derm7pt test set, the best result in terms of BA and R-MEL was achieved by the second approach with the scenario $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$. It is worth noting that the Derm7pt dataset is significantly older than the other datasets, resulting in a considerable domain shift in terms of data quality. However, the difference in terms of BA between $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ and $L_{\text{P}} + L_{\text{M}}$ in Derm7pt is only 0.2%. Additionally, when considering the average BA, taking into account the uncertainty based on the standard deviation, for the three datasets for each approach, we obtain values $84.1\% \pm 5.4\%$ and $80.3\% \pm 2.4\%$ for the first and second approaches, respectively.

Consequently, we can conclude that the most favorable approach in terms of performance is the first one, corresponding to scenario $L_{\text{P}} + L_{\text{M}}$. However, if we prioritize a simpler explanation, the best approach would be $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$.

5.2.2 Observing Prototypes and Explanation

In fig. 5.5, we can observe the difference between the prototypes obtained for the DN-169 architecture in two different scenarios. In the $L_{\text{P-1C}}$ scenario, there is no supervision at the prototype level, resulting in prototypes learned from regions outside the skin lesion, including corners and black borders of the

image. However, in the $L_{\text{P-1C-Masked}}$ scenario, where we apply prototype-level supervision, we notice that the prototypes consistently represent the interior of the lesion, and their activation in the rest of the image is limited to similar regions within the lesion. They are not activated by confounding factors beyond the lesion's boundary.

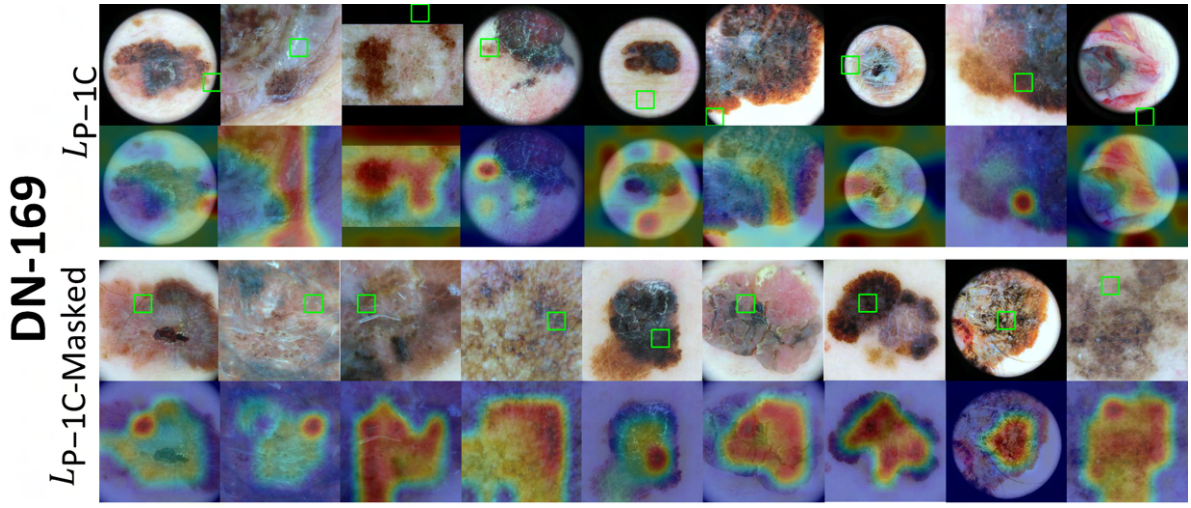


Figure 5.5: Nine prototypes were exclusively learned for the malignant class of melanoma using the Densenet-169 (DN-169) architecture in two distinct scenarios: $L_{\text{P-1C}}$ and $L_{\text{P-1C-Masked}}$. In the second scenario, there is supervision to ensure that the prototypes are located within the lesion's boundary and not outside it, while in the first one there is no supervision. For each scenario, we showcase the 9 melanoma prototypes, identified by a green square, along with their activation throughout the image, referred to as the self-activation map. Without supervision, the learned prototypes may refer to skin areas far from the lesion's border, or even black edges or corners of the image, which could have potentially less clinical relevance.

By analyzing fig. 5.6, we can observe prototypes obtained for 3 architectures in two distinct scenarios. The first row of each architecture displays the prototypes learned by the model, representing parts of the lesion delimited by a green square. This scenario involves prototype-level supervision, ensuring that these prototypes pertain to the interior of the lesion rather than outside its boundary, referred to as $L_{\text{P-1C-Masked}}$. In the second row of each architecture, we can see the prototypes learned in the scenario $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$. This approach encourages more diversified prototype learning while maintaining the aforementioned supervision.

We can also observe the values of L_{ICD} and V_{ICN}, which quantitatively measure the intra-class diversity of the prototypes. These values are normalized between 0 and 1 and they are presented in fig. 5.6. For instance, we notice that in all cases, the prototypes obtained in the scenario $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ exhibit higher values of L_{ICD} and V_{ICN} when compared to the respective scenario without promoting intra-class diversity of prototypes. Specifically, for the three architectures (RN-18, RN-50, ENB-3), the value of L_{ICD} increased by 0.109, 0.11, and 0.011, and V_{ICN} increased by 0.015, 0.025, and 0.194, respectively. Attached in table A.3, we can see that in this scenario, the prototypes not only demonstrate

greater intra-class variability but also have similar levels of importance when assessing their impact on the model upon removal.

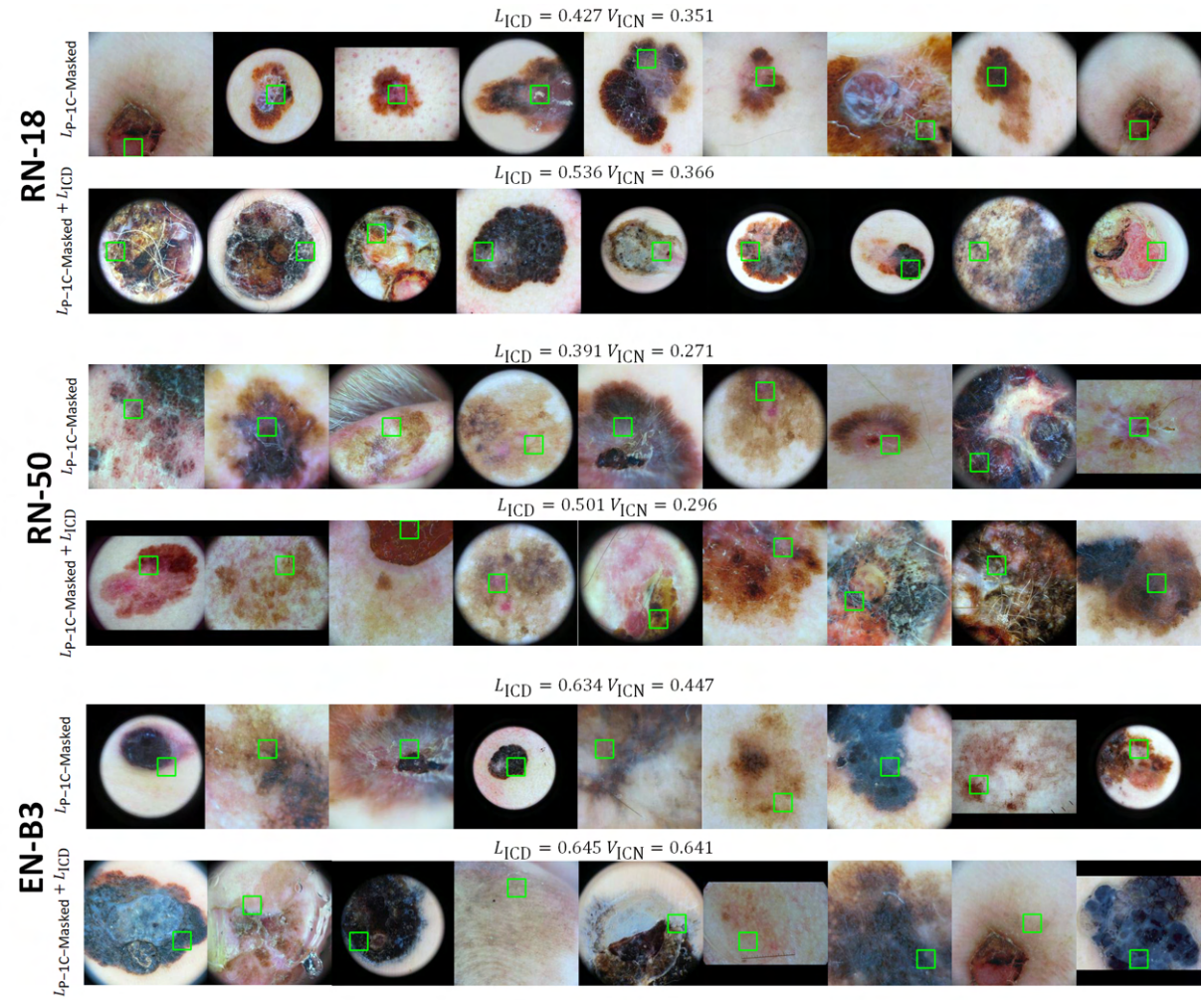


Figure 5.6: Nine prototypes learned exclusively for the malignant class of melanoma for three architectures: RN-18, RN-50, and EN-B3 in two distinct scenarios, namely, $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$. In both scenarios, there is supervision to ensure that the prototypes are located within the lesion's boundary and not outside it. In the latter scenario, intra-class diversity of the prototypes is encouraged through an additional term in the loss, denoted as L_{ICD}. The quantitative metrics for intra-class diversity, L_{ICD} and V_{ICN}, are presented for each scenario.

Additionally, considering the two scenarios and the intra-class diversity metrics, we can assert that, in this approach, as we only have prototypes corresponding to the malignant class, particularly melanoma, the prototypes of EN-B3 show more diversity among themselves than those of RN-18, which, in turn, are more diverse than RN-50. It is also interesting to note that among these three architectures, EN-B3, which demonstrated the highest intra-class diversity in the prototypes in the scenario $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$, also achieved the highest validation BA with a score of 82.7%.

In fig. 5.7, we can observe an example of the explanation generated in this approach. Notice how

the explanation in this case solely relies on the melanoma prototypes. The more similar the image is to the melanoma prototypes, the more negative the total contribution (TC) becomes. If the sum of this contribution and the model's bias is negative, the image is classified as melanoma. On the other hand, when the image bears little resemblance to the melanoma prototypes, TC becomes less negative. If, when added to the bias, it yields a positive value, the image is classified as nevus. In other words, melanoma represents the malignant class, associated with a more concerning diagnosis or, in other words, a "negative" outcome, resulting in a negative score in our model. Conversely, a diagnosis as nevus, representing the benign class, is associated with a less worrisome diagnosis or, in other words, a more "positive" outcome, leading to a positive score. This analogy is presented to facilitate a better understanding of the explanation.

In fig. 5.7, we can see the explanation for a melanoma diagnosis, where we have high similarity values with the melanoma prototypes. On the other hand, in the appendix, in fig. A.15, representing a nevus diagnosis, the similarities with the prototypes are lower. Furthermore, the most activated region in the test image with the prototypes is consistently related to the same area, which, in this instance, corresponds to the interior of the lesion. The fundamental difference lies in the degree of resemblance this region bears to the melanoma prototypes, determining whether the test image belongs to the melanoma or nevus class.

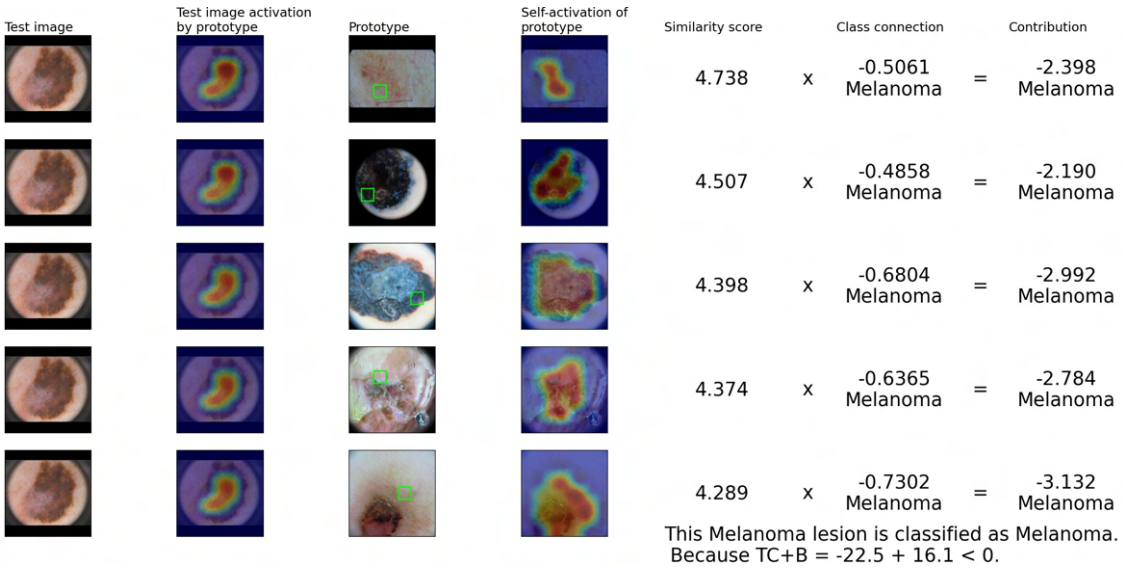


Figure 5.7: Explanation generated by the interpretable model in the scenario with prototype-level supervision and promotion of intra-class diversity, denoted as $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$. The CNN backbone used was EN-B3. The test image belongs to the PH^2 [15] dataset and is correctly classified as melanoma. Observe how the high resemblance to the melanoma prototypes results in a negative sum of total contribution with the bias (TC + B < 0), leading to the classification as melanoma, which is associated with a negative score. We are presenting only the top 5 prototypes that are most similar to the test image, instead of all 9, for the sake of simplicity. However, it should be noted that the value of TC takes into account all 9 prototypes.

5.3 Unraveling the Link: Prototypes & Dermatology Concepts

One question we can pose is, given an interpretable scenario where the prototypes learned by the model lie within the boundary of the lesion, can the prototype represent a dermatological concept?

To address this, we will consider the model from the interpretable scenario with prototype-level supervision, denoted as $L_{\mathsf{P}} + L_{\mathsf{M}}$, for the EN-B3 architecture in the binary problem (sections 3.1 and 5.1.1.B). We will utilize the EASY Dermoscopy Expert Agreement Study dataset, which comprises 129 melanoma images and 113 nevus images[1]. Each image has annotations from 5 expert physicians identifying dermatological concepts present in each image.

To assess whether each prototype, see fig. 5.2, can represent a concept, we obtain the 5 images from the mentioned dataset that belong to the same class as the prototype and are closest in the latent space of the model. For each of these 5 images, we observe the patch that is most activated by the respective prototype. [2] We then identify the concepts present in this patch, which have been commonly identified by at least 3 physicians. The concepts most frequently observed among the 5 images closest to the prototype are recognized as potential concepts that the prototype may be detecting. The results obtained are shown in fig. 5.8.

Upon analyzing the fig. 5.8 and exploring the prototypes of melanoma, it becomes evident that six distinct concepts have been identified. The most prominent concept among them is "Network - Atypical pigment network + Reticulation" (N-APN+R), which stands out with a probability of precisely 60% when detected in most of the prototypes. The probability is tied to the number of patches among the five most similar to the prototype that have the concept. Additionally, we observe that the prototypes can be grouped into four categories, as each group yields the same results. Specifically, group 1 comprises prototypes 1, 2, 5, 6, and 7; group 2 comprises prototype 3; and group 3 consists of prototype 4. Furthermore, group 4 includes prototypes 8 and 9. In appendix A.6 one can observe these groups.

Now, shifting our focus to the outcomes for the prototypes of nevus, we discover three groups: group 5 encompasses prototypes 10 to 13 and 15, 16 and 18; group 6 consists of prototype 14; and group 7 is represented by prototype 17. None of the detected concepts in these groups were identified with a probability higher than 40%. From these observations, the model appears to exhibit a bias towards capturing the concept N-APN+R when detecting melanoma. However, when examining the prototypes of nevus, there is no compelling evidence that the model is confidently capturing any specific concept, as none of them exceeded a 40% probability. Nevertheless, considering that the dataset encompasses 31 possible concepts, the results still offer seven concepts that the prototypes might tend to represent. It is essential to note the ranked concepts associated with melanoma, in descending order by their presence in the prototypes: **1)** N-APN+R; **2)** SWS-SWS, V-D, G+C-I; **3)** V-P and **4)** N-NPN. As for the nevus

[1] https://api.isic-archive.com/collections/166/?page=1

[2] The patch has the same dimensions as a prototype, which is a 32×32-pixel square (equivalent to 1×1 in the 7×7 feature map z (section 3.1)).

concepts, they are: **1)** SWS-SWS; **2)** N-NPN and **3)** N-APN+R.

These findings suggest that the prototypes may have the potential to represent known dermatological concepts used in medical terminology. However, drawing definitive conclusions requires further research. It is plausible that the model would benefit from guidance in learning prototypes that more accurately represent these specific concepts.

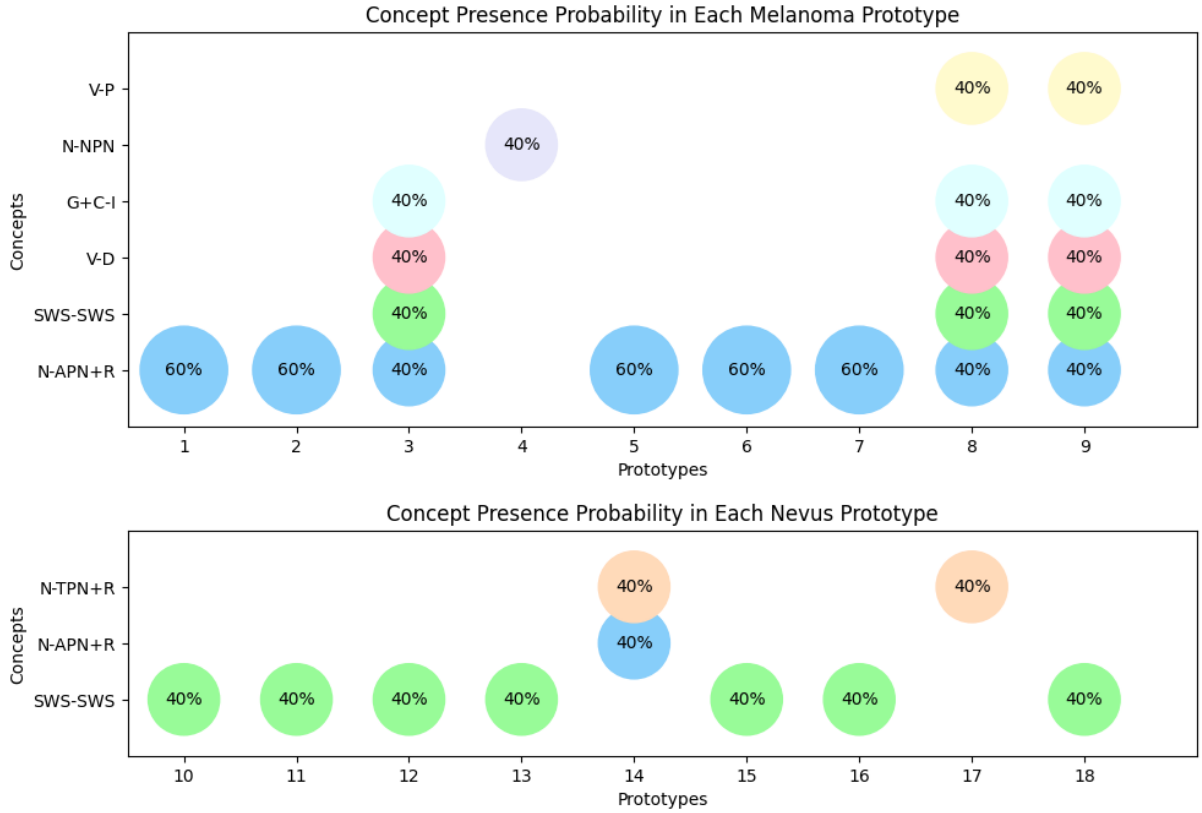


Figure 5.8: Probability of the identified concepts being present in each prototype. The probability is related to the number of patches out of the five most similar to the prototype that possess the concept. Each concept is represented by a different color. The concepts are presented for both melanoma and nevus prototypes. The identified concepts are as follows: "Network - Atypical pigment network + Reticulation" (N-APN+R), "Globules + Clods - Irregular" (G+C-I), "Shiny white structures - Shiny white streaks" (SWS-SWS), "Vessels - Dotted" (V-D), "Vessels - Polymorphous" (V-P), "Network - Negative pigment network" (N-NPN),"Network - Typical pigment network + Reticulation" (N-TPN+R).

5.4 Guideline-Based Evaluation of Medical Image Analysis XAI

In this section, we will conduct an assessment of the explanations provided by our approaches in the binary problem (refer to section 5.1.1 and section 5.2), comparing them with the use of a post-hoc technique such as Grad-CAM [34] in the black-box model. This evaluation is conducted in accordance with five guidelines [16].

The five guidelines are outlined as follows [16]:

- **G1-Understandability:** Explanations should be clear and easily understandable by medical professionals, even without technical expertise.
- **G2-Clinical Relevance:** Explanations should be directly applicable to physicians' clinical decision-making processes and support their reasoning.
- **G3-Truthfulness:** Explanations must faithfully represent the decision-making process of the AI model. This is a fundamental requirement for G4.
- **G4-Informative plausibility:** Users should be able to assess the plausibility of explanations, which can help them evaluate the AI's decision quality, including any potential flaws or biases.
- **G5-Computational efficiency:** Generating explanations should be done promptly, meeting the time constraints acceptable to medical professionals for the specific task at hand.

For this comparative analysis, the interpretable scenarios of the binary problem from the two explored approaches were considered. These scenarios include L_{P}, $L_{\text{P}} + L_{\text{M}}$, $L_{\text{P}} + L_{\text{R}}$, $L_{\text{P-Masked}}$, $L_{\text{P-1C}}$, $L_{\text{P-1C-Masked}}$, $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$, as well as the corresponding non-interpretable black-box model, utilizing the CNN backbone EfficientNet B3.

Starting with the first criterion, G1, we can readily conclude that all the considered scenarios meet this criterion. Regarding G2, given the analysis presented in section 5.3, which indicate that these prototypes may represent dermatological concepts for lesion diagnosis, we consider this criterion satisfied by the interpretable scenarios since the decision and explanation depend entirely on the prototypes. As for the black-box model, we will consider it partially met G2 [16]. Regarding criterion G5, all images in both the interpretable scenarios and the post-hoc method used in the black-box model generate explanations in under 10 seconds. As a result, all of them meet G5. It is worth mentioning that generating explanations in the interpretable scenarios does not involve calculating gradients, unlike the post-hoc method used in the black-box model.

To assess G3 and G4, we employed metrics similar to those proposed in the research article by Jin *et al.* [16] and the PH^2 test dataset was utilized [15]. Nevertheless, the metrics presented in their study were exclusively applied to post-hoc methods, most of which just generate heat maps for each image to elucidate the final decision. Conversely, in our specific case, for the interpretable scenarios, our explanatory approach goes beyond mere heat map displays. Instead, it illustrates the resemblance between the target skin lesion for diagnosis and various prototypes. This similarity can be locally observed through an activation map, and the visualization of the prototypes itself conveys information. To ensure a comprehensive comparison between our interpretable approaches and the black-box model concerning the criteria of G3 and G4, and to adhere as closely as possible to the suggestions outlined

in the article by Jin *et al.* [16], we decided to create a unique map for each image. This map is the result of a weighting process that involves all the activation maps of the image and the different prototypes.

For instance, in the first approach, there are 18 prototypes, 9 representing melanoma and 9 representing nevus. Thus, there are 18 computed similarities between the image and these prototypes. However, these similarities are weighted differently based on the final diagnosis. These weights are present in the model's last layer, denoted as w_h. The first row of weights corresponds to when the image is diagnosed as melanoma ($\hat{y_i} = 0$), and the second row corresponds to when it is diagnosed as nevus ($\hat{y_i} = 1$). Consequently, for an image x_i, the activation maps between image i and each prototype j can be represented as $A_{i,j}$. The unique activation map for that image is then given by

$$A_i = \sum_{j=1}^{m} A_{i,j} \times w_{h:\hat{y}_i,j} \quad , \tag{5.1}$$

where $w_{h:\hat{y}_i,j}$ denotes the weight of the activation map between image i and prototype j when the predicted label was $\hat{y}_i$. Recall that m represents the total number of prototypes. Similarly, in the second approach, although it is quite similar, there are only melanoma prototypes, and the last layer has only one row of 9 weights. Since these weights are negative, their absolute values are used in the weighted summation. In this case we have

$$A_i = \sum_{j=1}^{m} A_{i,j} \times |w_{h:j}| \quad , \tag{5.2}$$

where $w_{h:j}$ represents the weight of the activation map between image i and prototype j, irrespective of the predicted label. For the black-box model case, the heat map H_i used for XAI comparison was generated using Grad-CAM.

To evaluate G3, we created a curve showing the variation of performance in terms of BA as we cumulatively removed percentages of the most important pixels from 0% to 100%, in increments of 5%. This was done based on A_i for interpretable scenarios and H_i for the black-box model. We also repeated the experiment with randomly generated maps, referring to these as baseline maps B_i [16].

For each scenario, given the respective curve C, we calculated the area under the curve C_A. The same process was applied to the baseline curve counterpart C^B with area under the curve C_A^B, obtained from the random maps. Thus, the metric representing G3 is given by

$$M_{G3} = ((C_A^B / C_A) - 1) \quad . \tag{5.3}$$

A higher value of M_{G3} indicates greater truthfulness provided by A_i for interpretable models and H_i for the black-box model. In appendix A, it is possible to observe the curves obtained for the different scenarios, whose areas were used to calculate the value of M_{G3}, particularly in fig. A.24 and in fig. A.25.

The results for the different scenarios according to the value of M_{G3} ranked from best to worst

are as follows: 1-$L_{\text{P-1C-Masked}}$: 0.3538; 2-$L_{\text{P}} + L_{\text{M}}$: 0.3218; 3-$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$: 0.2924; 4-Black-Box: 0.2123; 5-$L_{\text{P}} + L_{\text{R}}$: 0.1473; 6-$L_{\text{P-Masked}}$: 0.1330; 7-$L_{\text{P}}$: 0.1183; 8-$L_{\text{P-1C}}$: -0.0437.

Taking these results into consideration, and in accordance with the guidelines, we can effectively state that the generated explanation satisfies G3 only for values above 0.5. Thus, we must conclude that based on the value of M_{G3}, no scenario meets G3 requirements. However, according to this metric, three interpretable scenarios, with supervision at the level of prototypes, demonstrate superior results compared to the black-box model.

As per the established guidelines, G3 serves as a prerequisite for G4. Nevertheless, we have chosen to conduct an evaluation of G4 despite this fact. To this end, we have utilized the following metric

$$M_{G4i} = \frac{\sum_i M_i \odot A_i}{\sum_i A_i} \times 100 \quad . \tag{5.4}$$

In the case of the black-box model, A_i is replaced by H_i. Essentially, for each image, we calculate the percentage of the map whose activation is situated outside the boundary of the skin lesion. Here, M_i denotes the binary mask, with 0 representing the interior of the lesion and 1 indicating the exterior. By averaging the results from all images in the dataset, we obtain the final value of M_{G4}. A lower value of M_{G4} in this context signifies a superior outcome, indicative of a more plausible explanation.

The results of M_{G4} for the different scenarios, ranked from best to worst, are as follows: 1-$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$: 2.5%; 2-$L_{\text{P-1C-Masked}}$: 2.5%; 3-$L_{\text{P-Masked}}$: 8.2%; 4-$L_{\text{P}} + L_{\text{M}}$: 18.5%; 5-Black-Box: 48.8%; 6-$L_{\text{P}} + L_{\text{R}}$: 73.8%; 7-$L_{\text{P}}$: 78.4%; 8-$L_{\text{P-1C}}$: 81.1%;

Considering these results and deeming a scenario with a value of M_{G4} below 50% as plausible, we can infer that four of the interpretable scenarios, with prototype supervision, meet the G4 criterion and outperform the black-box model when viewed independently of the G3 prerequisite.

The analysis of interpretable scenarios is based on a single activation map, A_i, which summarizes information about the critical parts of the similarity between the skin lesion under diagnosis and the prototypes. However, this map does not fully represent all the explanations available to the physician. While it identifies important areas of the image, additional insights can be gained by visualizing the prototypes, which may represent important dermatological concepts for diagnosis (see section 5.3).

Moreover, the decision is solely determined based on the similarity between the image and the prototypes, and these similarities construct the inherent explanation of the model. There is no need for a post-hoc method as in the black-box model. Therefore, although the approach associated with calculating M_{G3} and M_{G4} has been utilized to approximate the truth and plausibility level of the model, we believe that further research is necessary to develop a fairer metric for comparing explanations between post-hoc methods and those provided by interpretable models.

6

Conclusions and Future Work

Contents

6.1 Conclusions

In this thesis, an intensive study was conducted on the potential use of an interpretable model similar to ProtoPNet as an example of XAI in the context of skin cancer detection for both binary Melanoma Vs Nevus and multiclass problems with eight distinct skin lesion classes. This study explored the use of prototype-level supervision techniques to ensure that the learned prototypes were located inside the lesion rather than outside, avoiding confounding factors such as image corners, hairs, rulers, and black borders, which are artifacts with little clinical relevance. The methods employed successfully achieved this objective.

Regarding the binary problem, it was observed that for the different CNN-backbones used, the black-box models outperformed the interpretable scenarios, both with and without prototype-level supervision, on the validation set. However, this trend was not consistent on the test sets, where the interpretable scenarios, particularly those with prototype-level supervision, generally performed better. This indicates that in the presence of datasets from different hospital domains with varying distributions compared to training and validation data, interpretable scenarios demonstrate superior generalization capabilities.

Moving on to the multiclass problem, the interpretable scenarios, with and without prototype-level supervision, faced greater challenges in competing with their black-box counterparts. However, on the test sets, the interpretable scenarios showed improved performance, especially in correctly identifying lesions as malignant.

Among both the binary and multiclass problems, the interpretable scenario with prototype-level supervision $L_{\text{P}} + L_{\text{M}}$ stands out as it sacrifices less performance compared to the black-box model on the validation set and generalizes better on the test sets. It also outperforms, in the binary problem, the black-box model on the guidelines G3-truthfulness and G4-informative plausibility. Notably, there is evidence suggesting that the learned prototypes may represent dermatological concepts, with the model tending to associate the concept of atypical pigment network with the melanoma class.

Furthermore in the binary problem, an alternative approach to using prototypes for all classes was proposed to simplify the model's generated explanation. By comparing diagnostic images solely with prototypes from the malignant class, the scenarios $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ showed better generalization than the black-box models. It is important to mention that in these scenarios, $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$, the supervision of prototypes belonging solely to the malignant class is achieved using information from masks, similar to $L_{\text{P}} + L_{\text{M}}$. However, in this case, the information is directly used in the forward process of the model without introducing a new component in the loss function. Additionally, in $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$, efforts were made to promote intra-class diversity among the prototypes without compromising performance compared to $L_{\text{P-1C-Masked}}$. This approach allowed for more diverse prototypes while maintaining similar weights for all prototypes in the model's decision-making process.

Moreover, among all the scenarios considered for the binary problem, $L_{\text{P-1C-Masked}}$ yielded the best

results in terms of G3, while $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ performed best in terms of G4, even though they did not outperform $L_{\text{P}} + L_{\text{M}}$ on the validation set in terms of BA.

Finally, it is crucial to mention that the EfficientNet-B3, whether used as a black-box model or as the backbone for interpretable scenarios, led to the highest performance results.

6.2 Future Work

Despite obtaining positive results indicating superior generalization capabilities in interpretable scenarios, especially those with prototype supervision, there is still room for improvement, particularly in the validation set. Further research is needed to surpass the performance of black-box models in skin cancer detection using this interpretable model with prototypes and non-expert supervision on the validation set while maintaining good generalization on the test sets. Notably, the interpretable model faced more challenges competing with black-box models in the 8-class scenario than in the binary problem.

Moreover, considering the challenges faced by the interpretable model in classifying the 8 classes, we propose a strategy for the future: transforming the multi-class problem into multiple binary classification tasks, with each class pitted against all others.

Additionally, a more intensive study is necessary to effectively understand the capacity of prototypes to capture concepts recognized by dermatologists in their lexical field. It might be necessary to provide additional information to the model to ensure that the notion of similarity between skin lesions aligns more with medical expectations and concepts known to dermatologists.

Consequently, future investigation is required to involve doctors more in the model training loop. We believe that the future of XAI in clinical image analysis should involve closer collaboration with medical professionals during the model training process, aiming to gather insights from them to make the model's decision-making process even more comprehensible, thereby instilling more confidence and safety.

A model that successfully achieves this goal would have a significant impact not only in terms of its explanatory component as an XAI tool but also as a potential educational resource for future healthcare practitioners.

Bibliography

[1] M. Combalia and et al., "BCN20000: Dermoscopic Lesions in the Wild," *arXiv preprint arXiv:1908.02288*, 8 2019.

[2] N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, and A. Halpern, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2018-April, pp. 168–172, 5 2018.

[3] P. Tschandl, C. Rosendahl, and H. Kittler, "Data descriptor: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, 8 2018.

[4] W. Li, J. Zhuang, R. Wang, J. Zhang, and W. S. Zheng, "Fusing Metadata and Dermoscopy Images for Skin Disease Diagnosis," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2020-April, pp. 1996–2000, 4 2020.

[5] H. Zunair and A. B. Hamza, "Melanoma detection using adversarial training and deep transfer learning," *Physics in Medicine & Biology*, vol. 65, p. 135005, 7 2020.

[6] X. Li, J. Wu, E. Z. Chen, and H. Jiang, "From Deep Learning Towards Finding Skin Lesion Biomarkers," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 2797–2800, 7 2019.

[7] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres, "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)," *35th International Conference on Machine Learning, ICML 2018*, vol. 6, pp. 4186–4195, 11 2017.

[8] A. Lucieri, M. N. Bajwa, S. A. Braun, M. I. Malik, A. Dengel, and S. Ahmed, "On Interpretability of Deep Learning based Skin Lesion Classifiers using Concept Activation Vectors," *Proceedings of the International Joint Conference on Neural Networks*, 5 2020.

[9] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh, “Seven-Point Checklist and Skin Lesion Classification Using Multitask Multimodal Neural Nets,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, pp. 538–546, 3 2019.

[10] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin, “This Looks Like That: Deep Learning for Interpretable Image Recognition,” *Advances in Neural Information Processing Systems*, vol. 32, 6 2018.

[11] A. J. Barnett, F. R. Schwartz, C. Tao, C. Chen, Y. Ren, J. Y. Lo, and C. Rudin, “IAIA-BL: A Case-based Interpretable Deep Learning Model for Classification of Mass Lesions in Digital Mammography,” *Nature Machine Intelligence*, vol. 3, pp. 1061–1070, 3 2021.

[12] C. Wang, Y. Chen, Y. Liu, Y. Tian, F. Liu, D. J. McCarthy, M. Elliott, H. Frazer, and G. Carneiro, “Knowledge Distillation to Ensemble Global and Interpretable Prototype-Based Mammogram Classification Models,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13433 LNCS, pp. 14–24, 9 2022.

[13] A. Bontempelli and et al., “Concept-level debugging of part-prototype networks,” *International Conference on Learning Representations*, 2023.

[14] A. Hussaindeen, S. Iqbal, and T. D. Ambegoda, “MULTI-LABEL PROTOTYPE BASED INTERPRETABLE MACHINE LEARNING FOR MELANOMA DETECTION,” *INTERNATIONAL JOURNAL OF ADVANCES IN SIGNAL AND IMAGE SCIENCES*, vol. 8, pp. 40–53, 6 2022.

[15] T. Mendonca, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira, “Ph2 - a dermoscopic image database for research and benchmarking,” *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 5437–5440, 2013.

[16] W. Jin, X. Li, M. Fatehi, and G. Hamarneh, “Guidelines and evaluation of clinical explainable ai in medical image analysis,” *Medical Image Analysis*, vol. 84, 2 2023.

[17] Skin cancer statistics — world cancer research fund international. [Online]. Available: https://www.wcrf.org/cancer-trends/skin-cancer-statistics/

[18] How does the sun and uv cause cancer? — cancer research uk. [Online]. Available: https://www.cancerresearchuk.org/about-cancer/causes-of-cancer/sun-uv-and-cancer/how-does-the-sun-and-uv-cause-cancer

[19] J. Kato, K. Horimoto, S. Sato, T. Minowa, and H. Uhara, “Dermoscopy of melanoma and non-melanoma skin cancers,” *Frontiers in Medicine*, vol. 6, p. 180, 8 2019.

[20] S. S. Mohammed, J. M. Al-Tuwaijari, and S. S. Mohammed, “Skin disease classification system based on machine learning technique: A survey,” *IOP Conference Series: Materials Science and Engineering*, vol. 1076, p. 012045, 2 2021.

[21] Y. Cheng, R. Swamisai, S. E. Umbaugh, R. H. Moss, W. V. Stoecker, S. Teegala, and S. K. Srinivasan, “Skin lesion classification using relative color features,” *Skin research and technology : official journal of International Society for Bioengineering and the Skin (ISBS) [and] International Society for Digital Imaging of Skin (ISDIS) [and] International Society for Skin Imaging (ISSI)*, vol. 14, pp. 53–64, 2 2008.

[22] Z. Liu and J. Zerubia, “Skin image illumination modeling and chromophore identification for melanoma diagnosis,” *Physics in medicine and biology*, vol. 60, pp. 3415–3431, 5 2015.

[23] M. Dildar, S. Akram, M. Irfan, H. U. Khan, M. Ramzan, A. R. Mahmood, S. A. Alsaiari, A. H. M. Saeed, M. O. Alraddadi, and M. H. Mahnashi, “Skin Cancer Detection: A Review Using Deep Learning Techniques,” *International Journal of Environmental Research and Public Health 2021, Vol. 18, Page 5479*, vol. 18, p. 5479, 5 2021.

[24] X. Jia, L. Ren, and J. Cai, “Clinical implementation of ai technologies will require interpretable ai models,” *Medical Physics*, vol. 47, pp. 1–4, 1 2020.

[25] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance) - publications office of the eu. [Online]. Available: https://op.europa.eu/de/publication-detail/-/publication/3e485e15-11bd-11e6-ba9a-01aa75ed71a1/language-en/format-PDFA1A

[26] A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, and C. Mooney, “Current Challenges and Future Opportunities for XAI in Machine Learning-Based Clinical Decision Support Systems: A Systematic Review,” *Applied Sciences 2021, Vol. 11, Page 5088*, vol. 11, p. 5088, 5 2021.

[27] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, pp. 206–215, 5 2019.

[28] M. Correia, A. Bissoto, C. Santiago, and C. Barata, “Xai for skin cancer detection with prototypes and non-expert supervision,” accepted at the Workshop on Interpretability of Machine Intelligence in Medical Image Computing at MICCAI 2023.

[29] C. Santiago, M. Correia, M. R. Verdelho, A. Bissoto, and C. Barata, “Global and local explanations for skin cancer diagnosis using prototypes,” accepted at the Eighth ISIC Skin Image Analysis Workshop at MICCAI 2023.

[30] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, “Explainable Machine Learning for Scientific Insights and Discoveries,” *IEEE Access*, vol. 8, pp. 42 200–42 216, 2020.

[31] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, “Definitions, methods, and applications in interpretable machine learning,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, pp. 22 071–22 080, 10 2019.

[32] B. H. van der Velden, H. J. Kuijf, K. G. Gilhuijs, and M. A. Viergever, “Explainable artificial intelligence (XAI) in deep learning-based medical image analysis,” *Medical Image Analysis*, vol. 79, 7 2022.

[33] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” *IEEE Access*, vol. 6, pp. 52 138–52 160, 9 2018.

[34] R. R. Selvaraju and et al., “Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization,” *International Conference on Computer Vision (ICCV)*, 2017.

[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” pp. 2921–2929, 2016.

[36] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, “Visualizing Deep Neural Network Decisions: Prediction Difference Analysis,” *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2 2017.

[37] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps,” *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, 12 2013.

[38] H. Wang, Z. Wu, and E. P. Xing, “Removing confounding factors associated weights in deep neural networks improves the prediction accuracy for healthcare applications,” *Pacific Symposium on Biocomputing*, vol. 24, pp. 54–65, 2019.

[39] S. Mohammadjafari, M. Cevik, M. Thanabalasingam, and A. Basar, “The 34th Canadian Conference on Artificial Intelligence Using ProtoPNet for Interpretable Alzheimer’s Disease Classification,” 2021.

[40] G. Singh and K. C. Yow, “Object or Background: An Interpretable Deep Learning Model for COVID-19 Detection from CT-Scan Images,” *Diagnostics 2021, Vol. 11, Page 1732*, vol. 11, p. 1732, 9 2021.

[41] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, pp. 833–851, 2 2018.

[42] Z. Mirikharaji, K. Abhishek, A. Bissoto, C. Barata, S. Avila, E. Valle, M. E. Celebi, and G. Hamarneh, "A survey on deep learning for skin lesion segmentation," *Medical Image Analysis*, p. 102863, 2023.

[43] C. Wang, Y. Liu, Y. Chen, F. Liu, Y. Tian, D. J. McCarthy, H. Frazer, and G. Carneiro, "Learning support and trivial prototypes for interpretable image classification," 1 2023.

[44] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, H. Kittler, and A. Halpern, "Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)," 2 2019.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, 12 2015.

[46] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10 691–10 700, 5 2019.

[47] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, 8 2016.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2014.

Appendix A: Extra Figures and Tables

Throughout this appendix, one can observe various additional images that complement the obtained results with more examples of prototypes derived from different approaches, as well as more instances of the explanations provided by interpretable models for different diagnostic cases. The order of the images and tables follows the sequence of the chapter 5.

A.1 Additional information for section 5.1.1.A

Table A.1: Best hyperparameter configuration for the results in Table 5.1.

Model	Approach	Hyperparameters	
		D	top-k
ResNet-18	L_{P}	512	25
	$L_{\text{P}} + L_{\text{M}}$	256	28
	$L_{\text{P}} + L_{\text{R}}$	256	40
	$L_{\text{P-Masked}}$	256	28
ResNet-50	L_{P}	256	40
	$L_{\text{P}} + L_{\text{M}}$	512	13
	$L_{\text{P}} + L_{\text{R}}$	512	19
	$L_{\text{P-Masked}}$	512	13
EfficientNet B3	L_{P}	512	19
	$L_{\text{P}} + L_{\text{M}}$	512	13
	$L_{\text{P}} + L_{\text{R}}$	256	10
	$L_{\text{P-Masked}}$	512	13
Densenet-169	L_{P}	512	49
	$L_{\text{P}} + L_{\text{M}}$	256	31
	$L_{\text{P}} + L_{\text{R}}$	128	49
	$L_{\text{P-Masked}}$	256	31
VGG-16	L_{P}	128	112
	$L_{\text{P}} + L_{\text{M}}$	128	76
	$L_{\text{P}} + L_{\text{R}}$	128	112
	$L_{\text{P-Masked}}$	128	76

A.2 Additional information for section 5.1.1.B

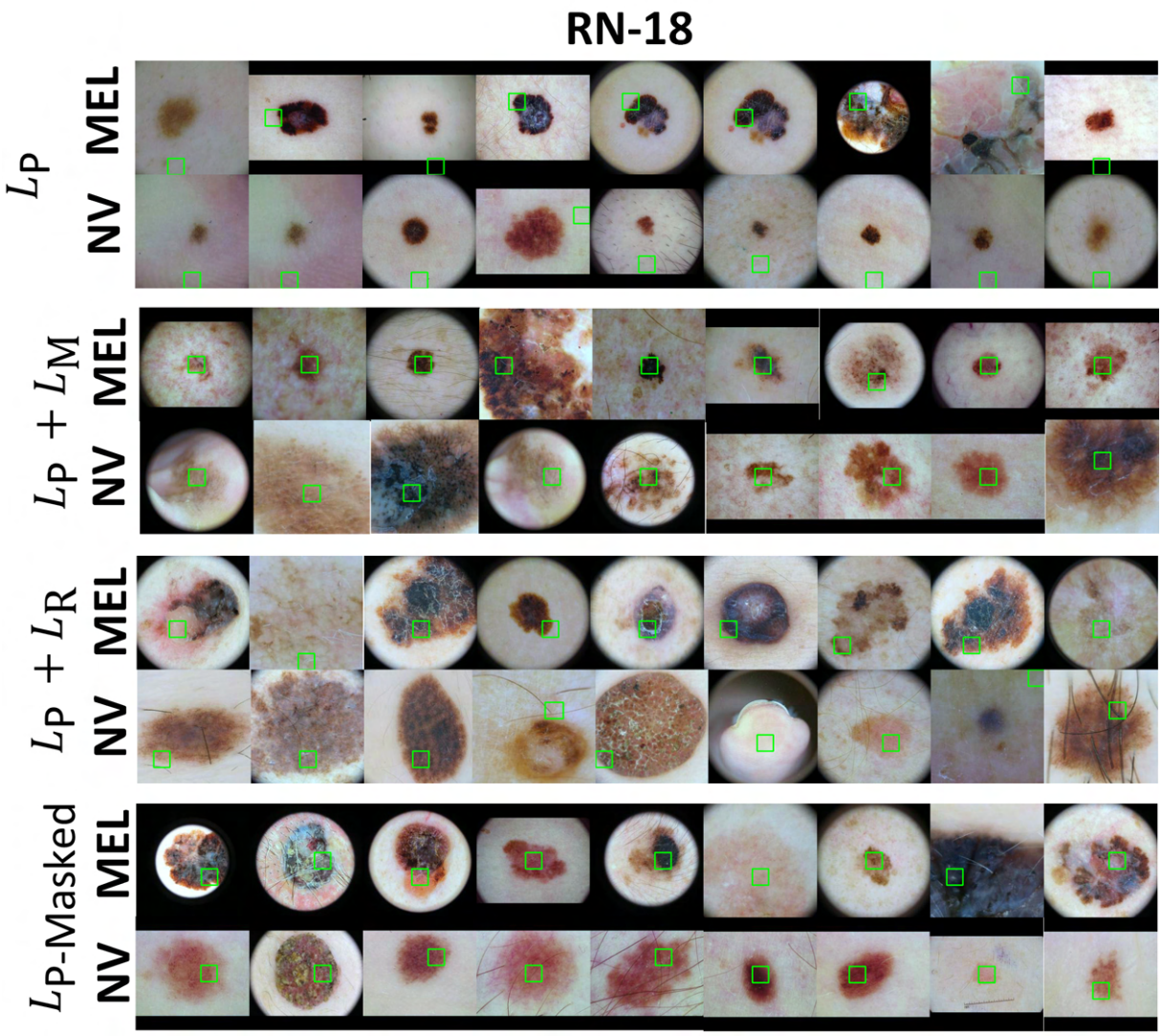


Figure A.1: Eighteen prototypes obtained for each interpretable scenario using the ResNet-18 architecture. For each scenario, we have two sets of prototypes: the first set corresponds to melanoma prototypes, and the second set corresponds to nevus class prototypes. Unsupervised scenario at the prototype level L_P, which does not ensure that the prototypes remain inside the lesion boundary, and supervised scenarios at the prototype level that already address this issue: $L_P + L_M$, $L_P + L_R$, $L_{P\text{-Masked}}$. All these scenarios are interpretable concerning the binary problem of Melanoma vs Nevus, following the approach outlined in section 3.1.

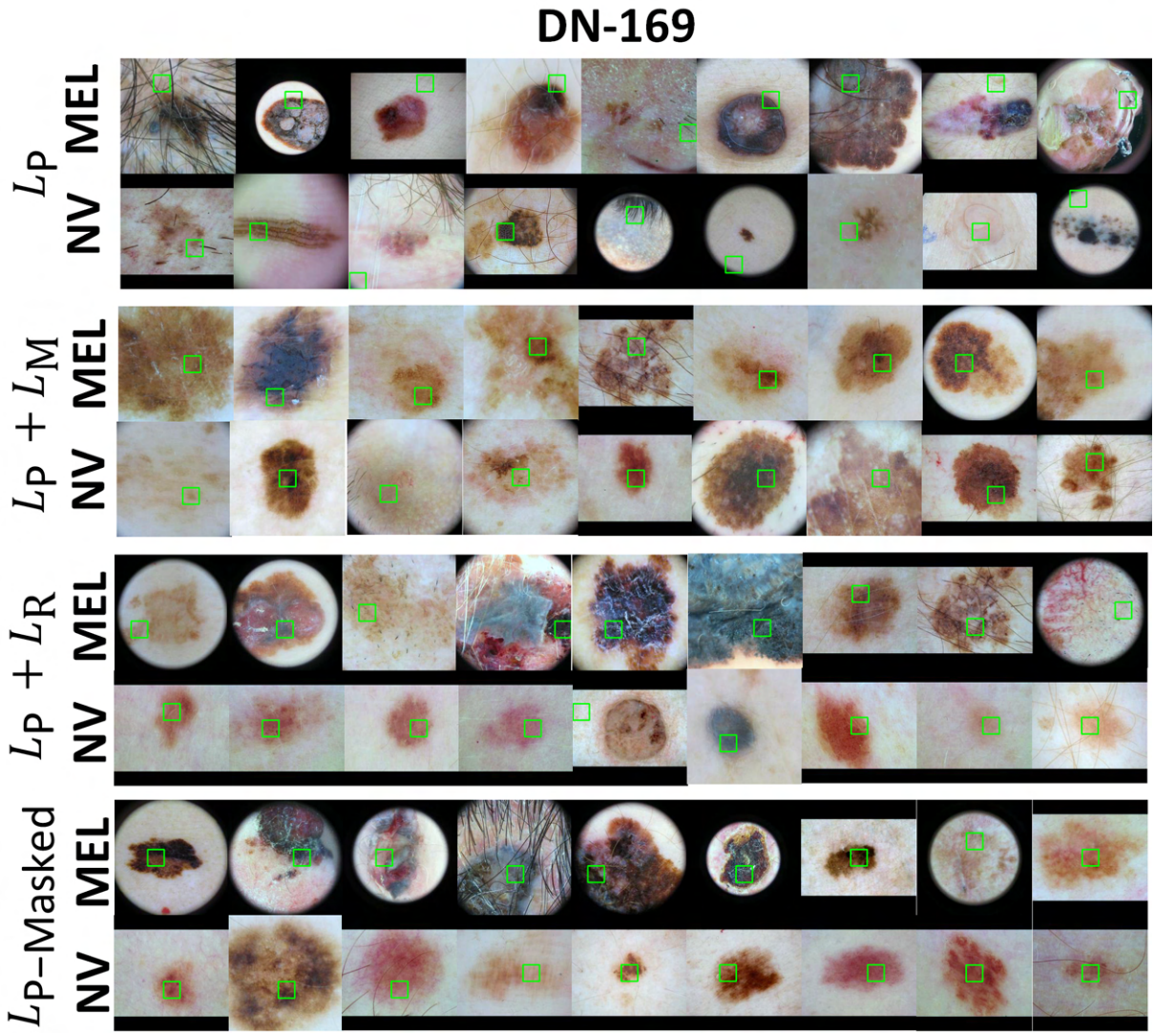


Figure A.2: Eighteen prototypes obtained for each interpretable scenario using the Densenet-169 architecture. For each scenario, we have two sets of prototypes: the first set corresponds to melanoma prototypes, and the second set corresponds to nevus class prototypes. Unsupervised scenario at the prototype level L_P, which does not ensure that the prototypes remain inside the lesion boundary, and supervised scenarios at the prototype level that already address this issue: $L_P + L_M$, $L_P + L_R$, $L_{P\text{-Masked}}$. All these scenarios are interpretable concerning the binary problem of Melanoma vs Nevus, following the approach outlined in section 3.1.

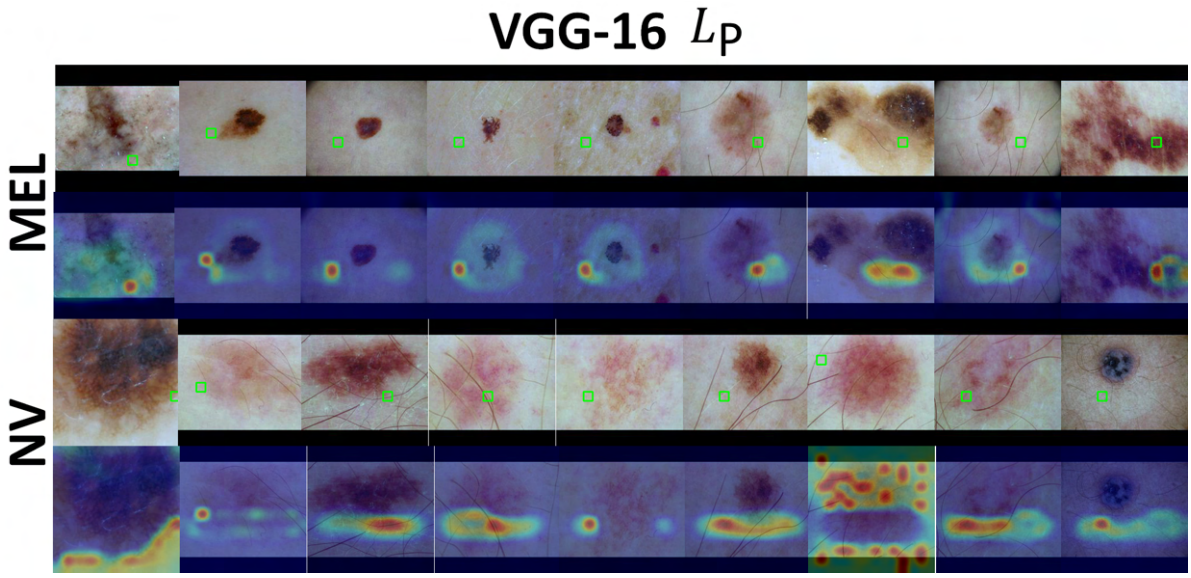


Figure A.3: Eighteen prototypes were obtained, nine of melanoma and nine of nevus, along with their respective activation maps for the L_P scenario using the VGG-16 architecture, for the binary problem employing the approach described in section 3.1.

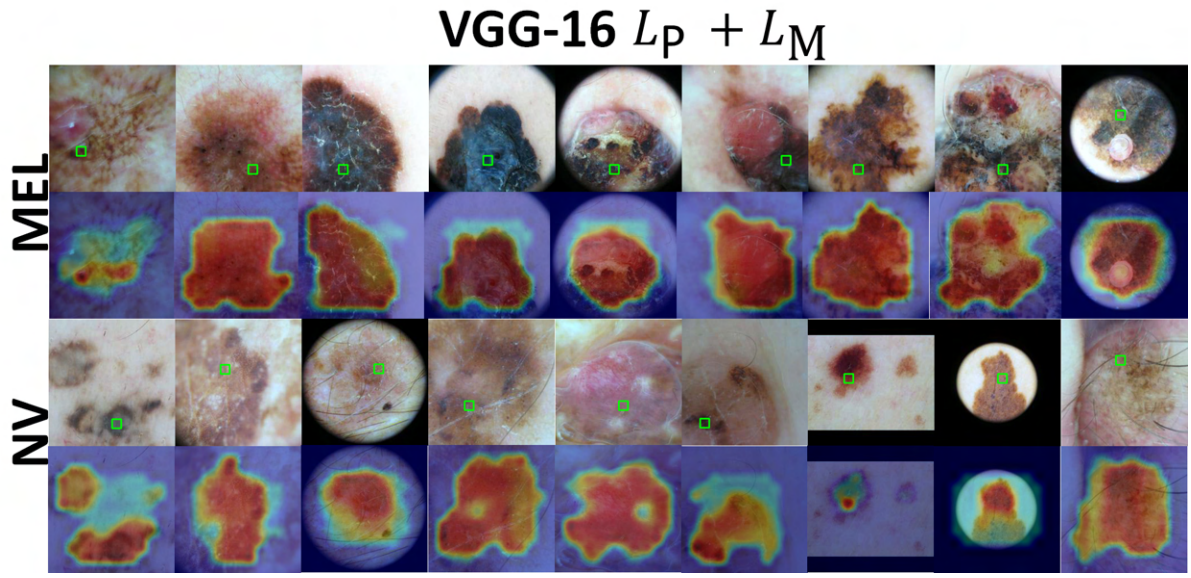


Figure A.4: Eighteen prototypes were obtained, nine of melanoma and nine of nevus, along with their respective activation maps for the $L_P + L_M$ scenario using the VGG-16 architecture, for the binary problem employing the approach described in section 3.1.

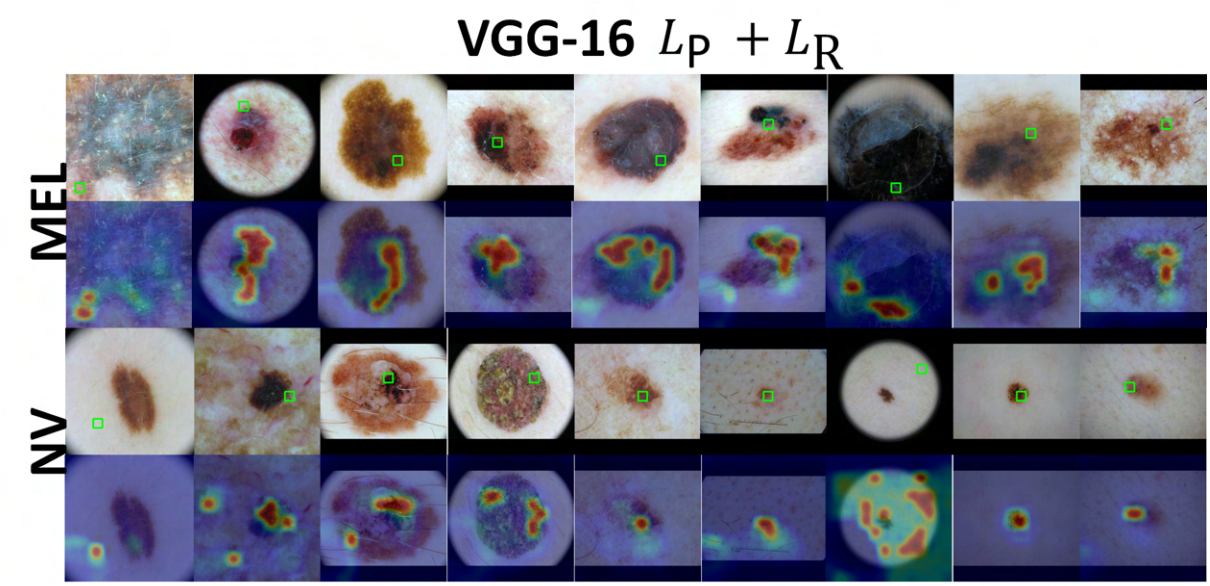


Figure A.5: Eighteen prototypes were obtained, nine of melanoma and nine of nevus, along with their respective activation maps for the $L_P + L_R$ scenario using the VGG-16 architecture, for the binary problem employing the approach described in section 3.1.

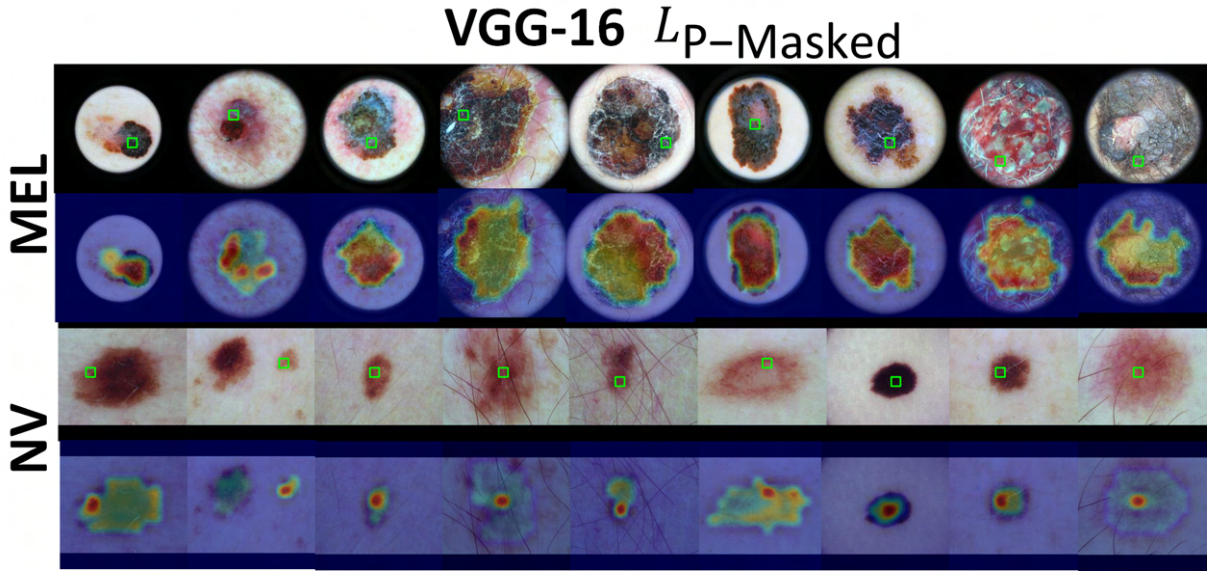


Figure A.6: Eighteen prototypes were obtained, nine of melanoma and nine of nevus, along with their respective activation maps for the $L_{P\text{-}Masked}$ scenario using the VGG-16 architecture, for the binary problem employing the approach described in section 3.1.

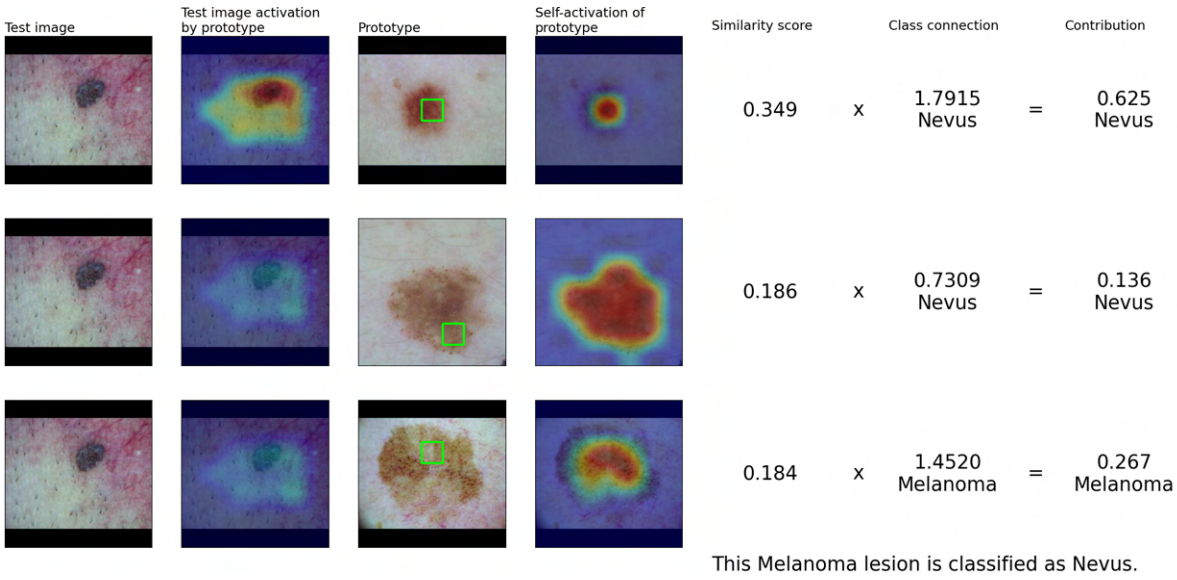


Figure A.7: Example of the explanation provided by the interpretable model in the $L_P + L_M$ scenario for the EN-B3 architecture, considering the binary problem melanoma vs nevus (section 3.1). The lesion, belonging to the ISIC 2019 validation set [1–3], is of the melanoma class and is misclassified. Upon observing the three most activated prototypes, the two prototypes with the highest similarity belong to the nevus class, whereas the other prototype belongs to the melanoma class.

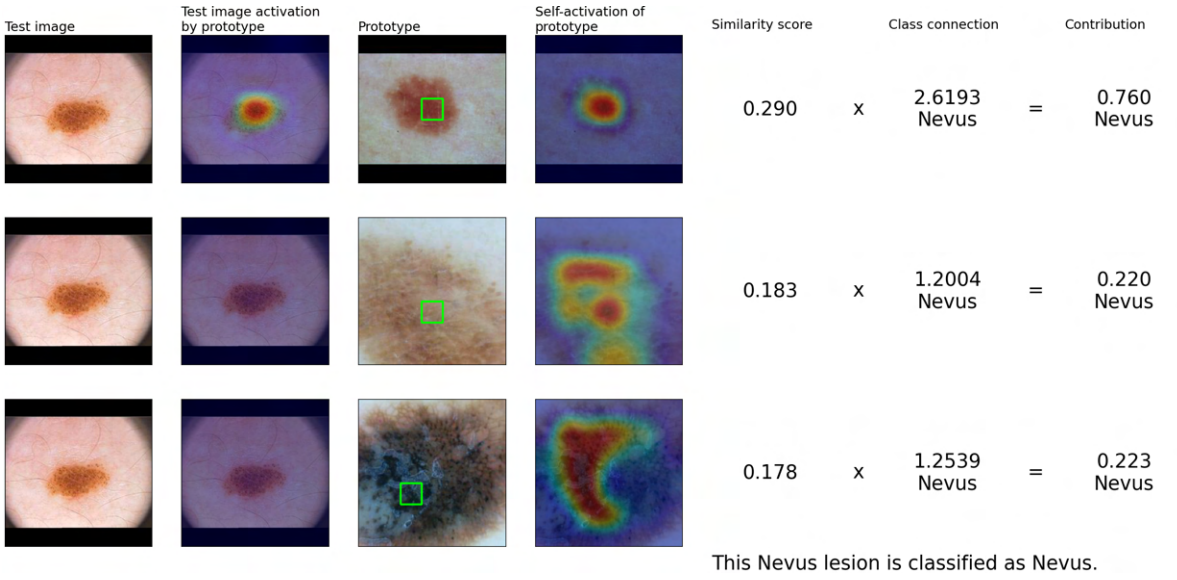


Figure A.8: Example of the explanation provided by the interpretable model in the $L_P + L_M$ scenario for the RN-18 architecture, considering the binary problem melanoma vs nevus (section 3.1). The lesion, belonging to the PH^2 test set [15], is of the nevus class and is correctly classified.

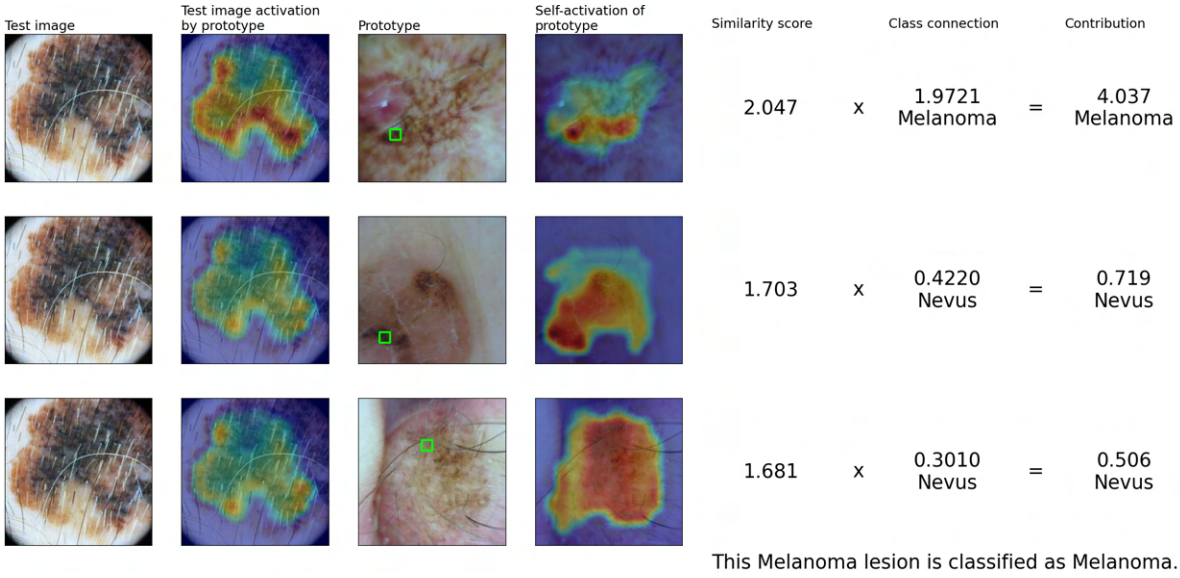


Figure A.9: Example of the explanation provided by the interpretable model in the $L_P + L_M$ scenario for the VGG-16 architecture, considering the binary problem melanoma vs nevus (section 3.1). The lesion, belonging to the ISIC 2019 validation set [1–3], is of the melanoma class and is correctly classified.

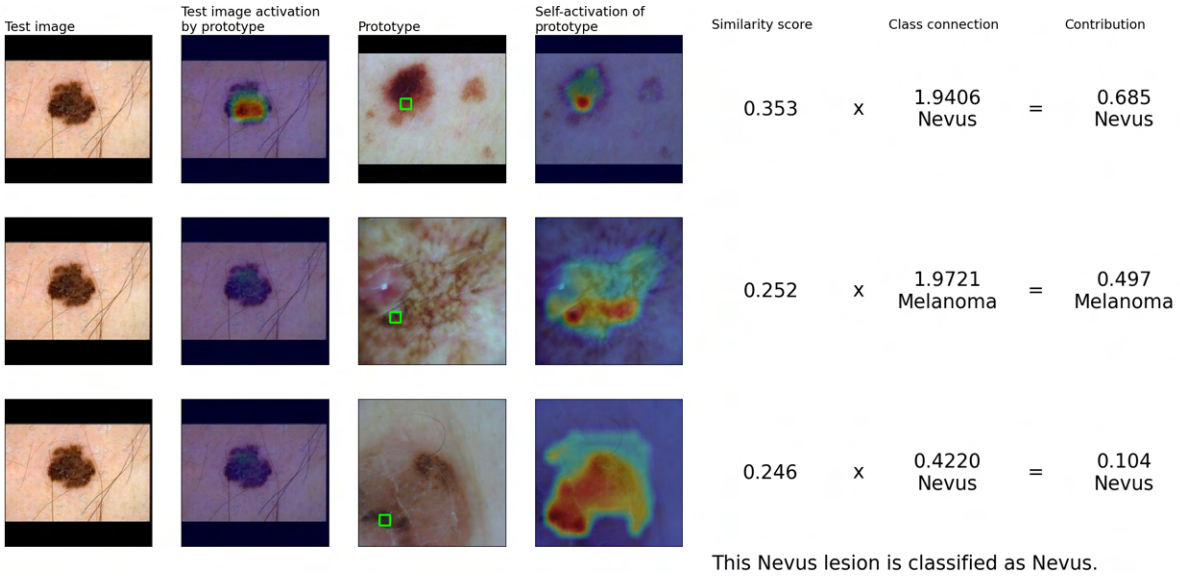


Figure A.10: Example of the explanation provided by the interpretable model in the $L_P + L_M$ scenario for the VGG-16 architecture, considering the binary problem melanoma vs nevus (section 3.1). The lesion, belonging to the Derm7pt test set [9] , is of the nevus class and is correctly classified.

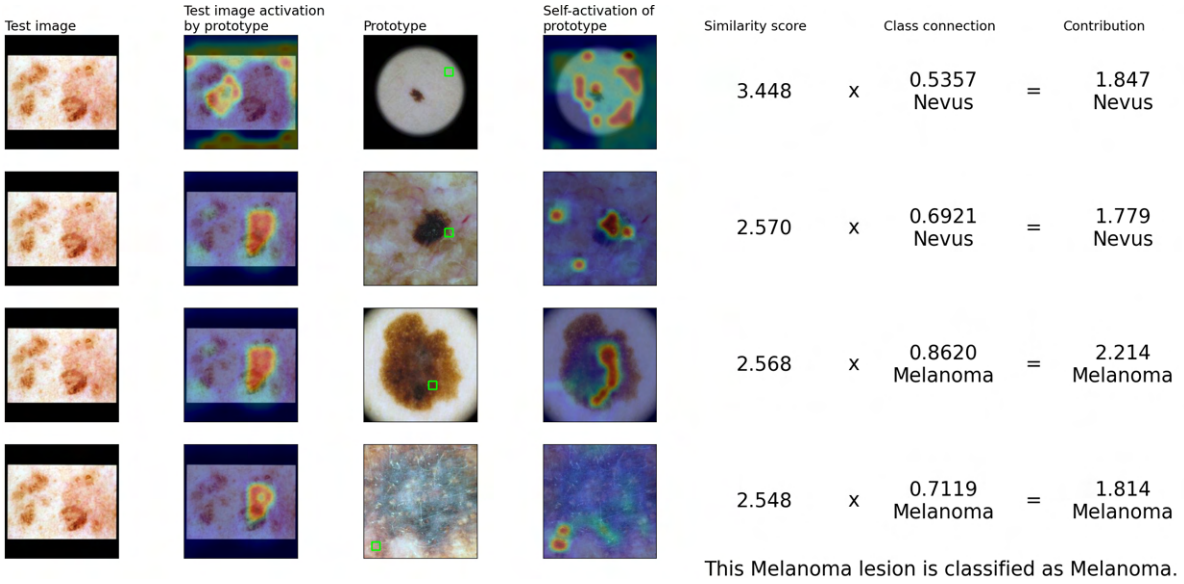


Figure A.11: Example of the explanation provided by the interpretable model in the $L_P + L_R$ scenario for the VGG-16 architecture, considering the binary problem melanoma vs nevus (section 3.1). The lesion, belonging to the Derm7pt test set [9] , is of the melanoma class and is correctly classified.

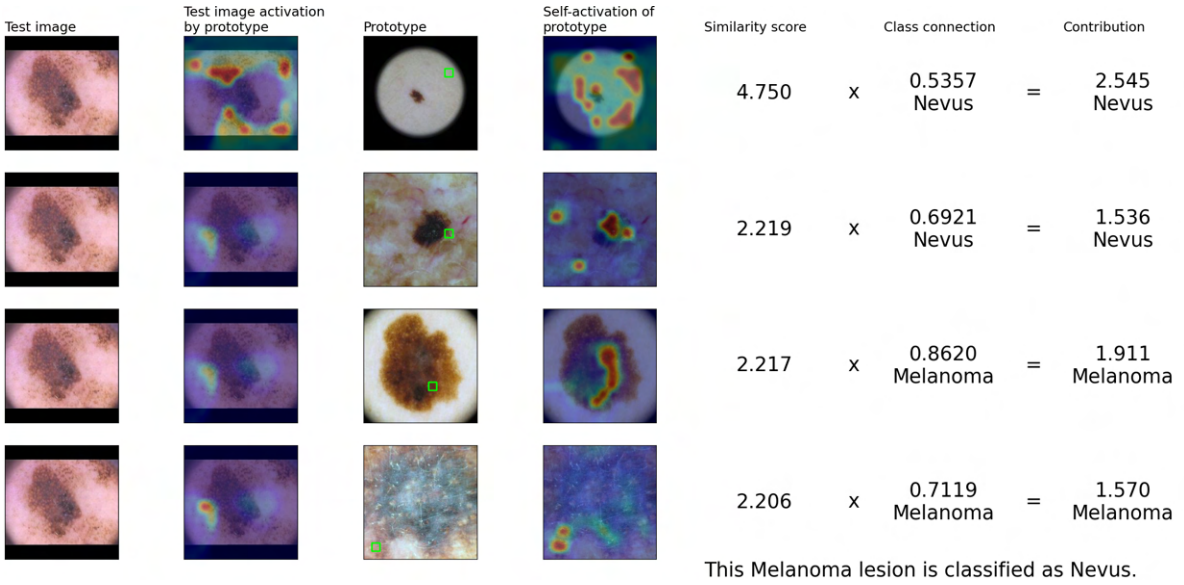


Figure A.12: Example of the explanation provided by the interpretable model in the $L_P + L_R$ scenario for the VGG-16 architecture, considering the binary problem melanoma vs nevus (section 3.1). The lesion, belonging to the PH^2 test set [15] , is of the melanoma class and is incorrectly classified.

A.3 Additional information for section 5.1.2.A

Table A.2: Best hyperparameter configuration for the results in Table 5.2.

Model	Approach	Hyperparameters	
		D	top-k
ResNet-18	L_{P}	128	16
	$L_{\text{P}} + L_{\text{M}}$	128	13
	$L_{\text{P}} + L_{\text{R}}$	512	3
	$L_{\text{P-Masked}}$	128	13
ResNet-50	L_{P}	256	40
	$L_{\text{P}} + L_{\text{M}}$	128	13
	$L_{\text{P}} + L_{\text{R}}$	256	7
	$L_{\text{P-Masked}}$	128	13
EfficientNet B3	L_{P}	512	49
	$L_{\text{P}} + L_{\text{M}}$	128	10
	$L_{\text{P}} + L_{\text{R}}$	128	16
	$L_{\text{P-Masked}}$	128	10
Densenet-169	L_{P}	256	22
	$L_{\text{P}} + L_{\text{M}}$	256	22
	$L_{\text{P}} + L_{\text{R}}$	256	22
	$L_{\text{P-Masked}}$	256	22
VGG-16	L_{P}	256	88
	$L_{\text{P}} + L_{\text{M}}$	512	28
	$L_{\text{P}} + L_{\text{R}}$	512	52
	$L_{\text{P-Masked}}$	512	28

A.4 Additional information for section 5.1.2.B

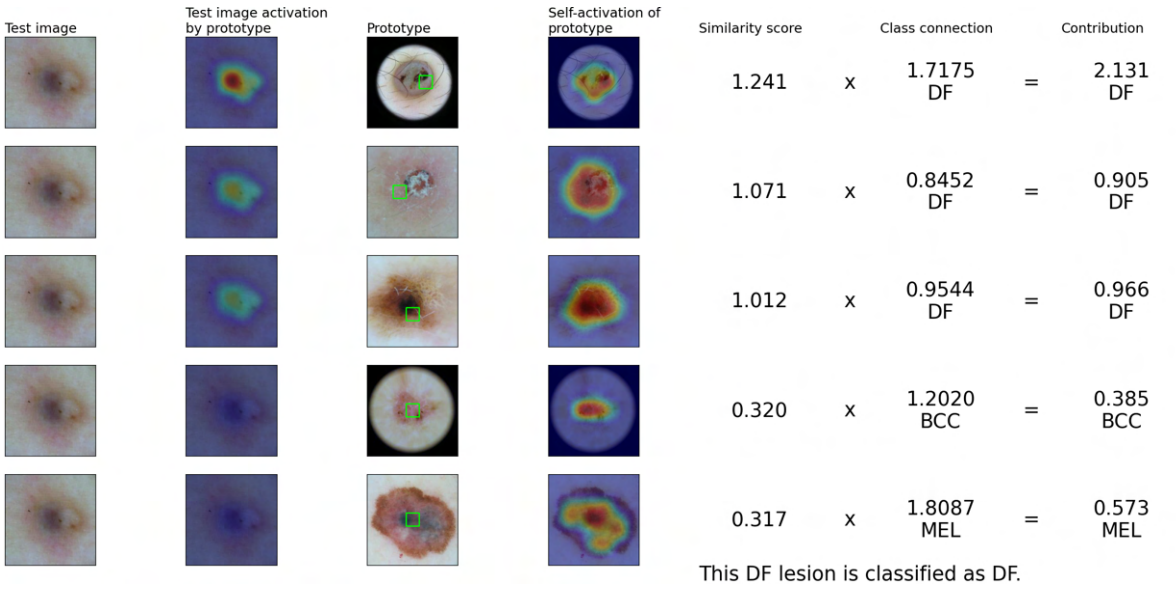


Figure A.13: Example of the explanation provided by the interpretable model in the $L_P + L_M$ scenario for the EN-B3 architecture and for the multiclass problem (sections 3.1 and 5.1.2). The lesion, belonging the ISIC 2019 validation set [1–3], is of the DF class and is correctly classified.

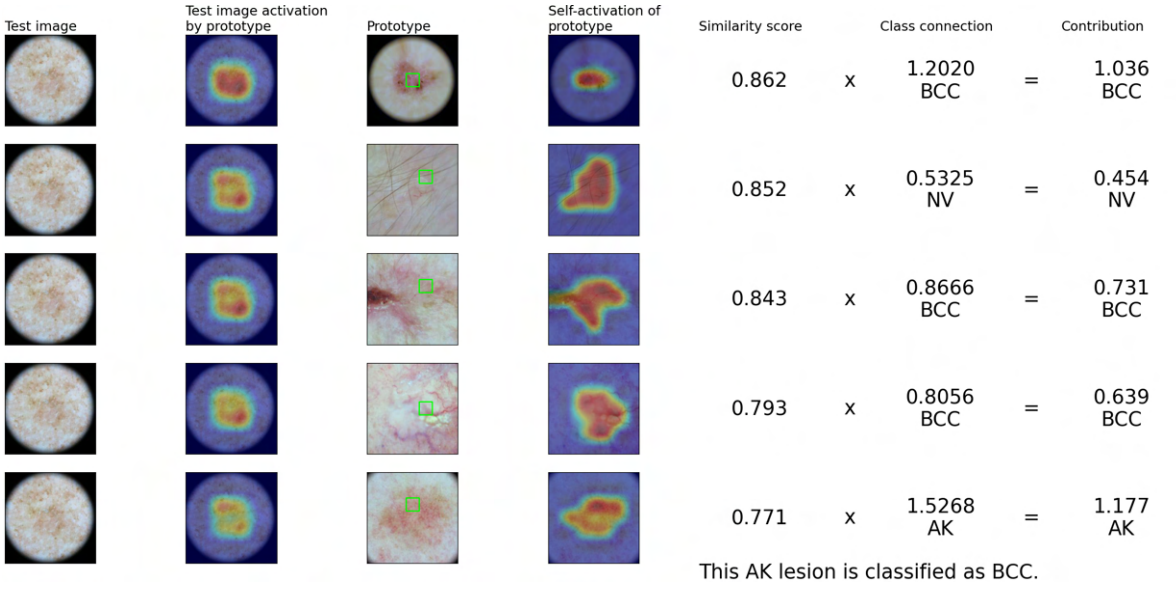


Figure A.14: Example of the explanation provided by the interpretable model in the $L_P + L_M$ scenario for the EN-B3 architecture and for the multiclass problem (sections 3.1 and 5.1.2). The lesion, belonging the ISIC 2019 validation set [1–3], is of the AK class and is incorrectly classified.

A.5 Additional information for section 5.2.2

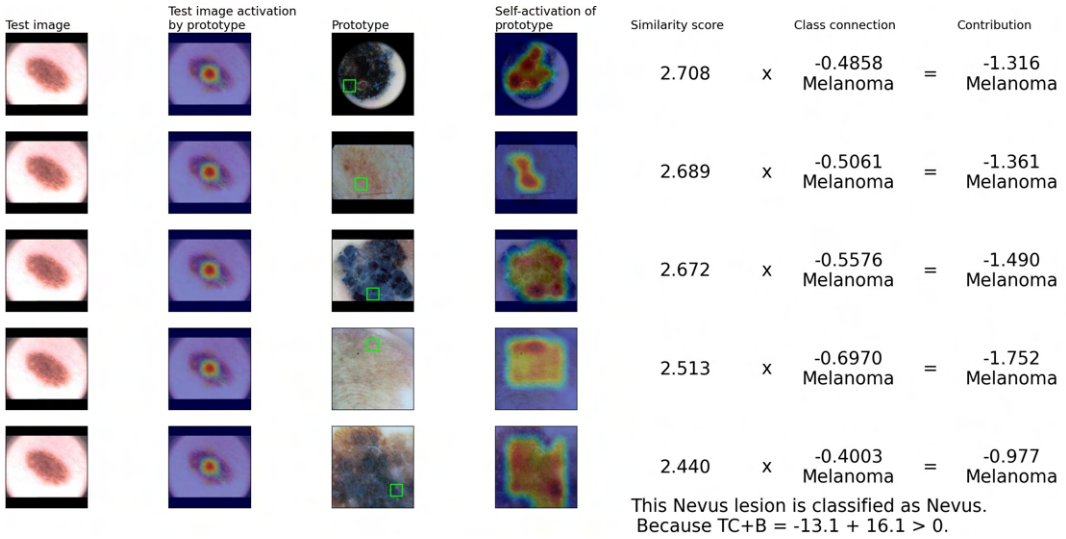


Figure A.15: Explanation generated by the interpretable model in the scenario with prototype-level supervision and promotion of intra-class diversity, denoted as $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ (section 3.2). The CNN backbone used is EN-B3. The test image belongs to the PH^2 [15] dataset and is correctly classified as nevus. Observe how the low resemblance to the melanoma prototypes results in a positive sum of total contribution with the bias ($\text{TC} + \text{B} > 0$), leading to the classification as nevus, which is associated with a positive score. We are presenting only the top 5 prototypes that are most similar to the test image, instead of all 9, for the sake of simplicity. However, it should be noted that the value of TC takes into account all 9 prototypes.

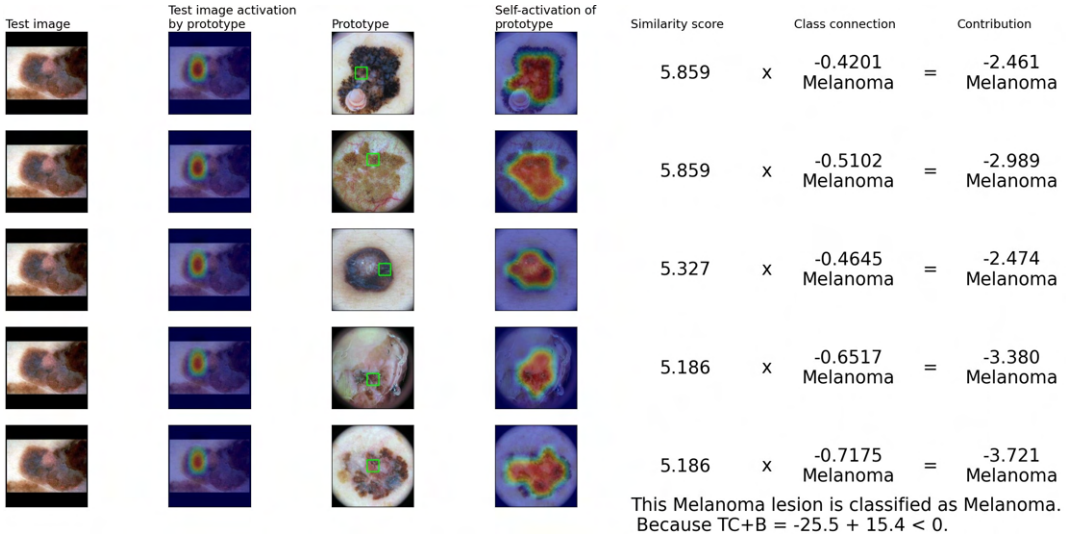


Figure A.16: Explanation generated by the interpretable model in the scenario with prototype-level supervision and promotion of intra-class diversity, denoted as $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ (section 3.2). The CNN backbone used is DN-169. The test image belongs to Derm7pt [9] dataset and is correctly classified as melanoma ($\text{TC} + \text{B} < 0$). We are presenting only the top 5 prototypes that are most similar to the test image, instead of all 9, for the sake of simplicity. However, it should be noted that the value of TC takes into account all 9 prototypes.

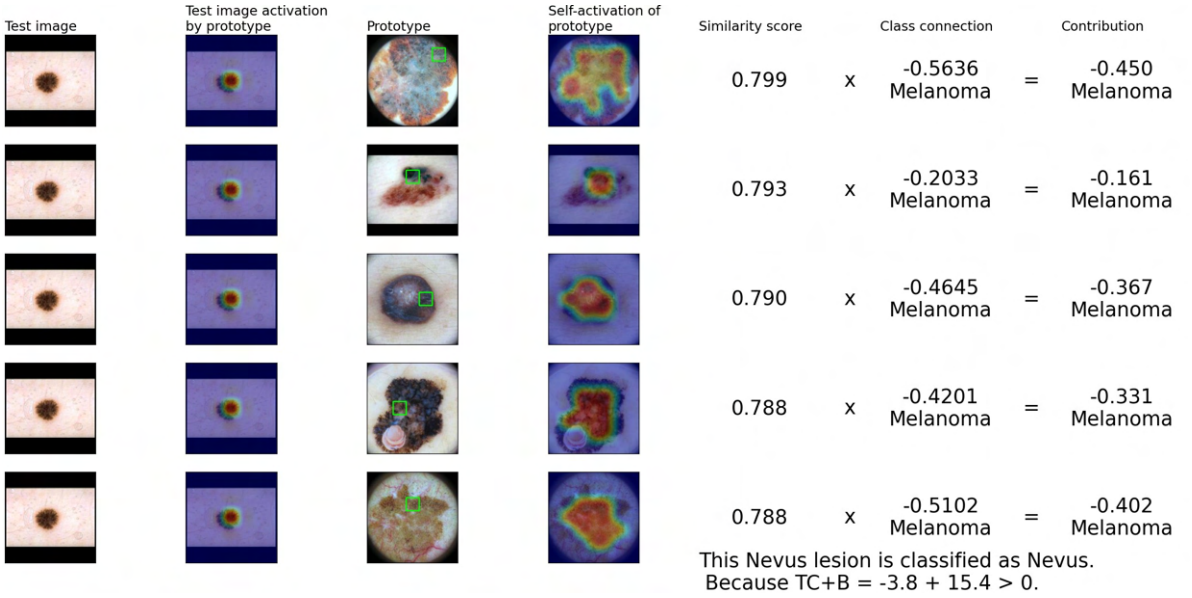


Figure A.17: Explanation generated by the interpretable model in the scenario with prototype-level supervision and promotion of intra-class diversity, denoted as $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ (section 3.2). The CNN backbone used is DN-169. The test image belongs to the Derm7pt [9] dataset and is correctly classified as nevus (TC + B > 0). We are presenting only the top 5 prototypes that are most similar to the test image, instead of all 9, for the sake of simplicity. However, it should be noted that the value of TC takes into account all 9 prototypes.

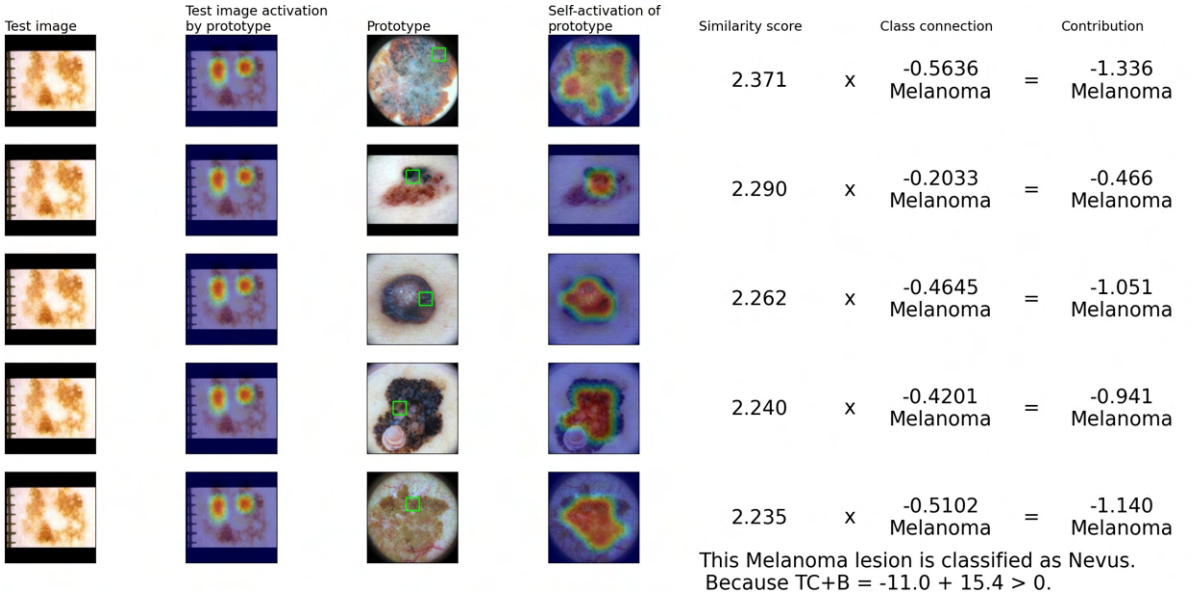


Figure A.18: Explanation generated by the interpretable model in the scenario with prototype-level supervision and promotion of intra-class diversity, denoted as $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ (section 3.2). The CNN backbone used is DN-169. The test image belongs to the Derm7pt [9] dataset and is incorrectly classified as nevus (TC + B > 0), it is in fact a melanoma lesion. We are presenting only the top 5 prototypes that are most similar to the test image, instead of all 9, for the sake of simplicity. However, it should be noted that the value of TC takes into account all 9 prototypes.

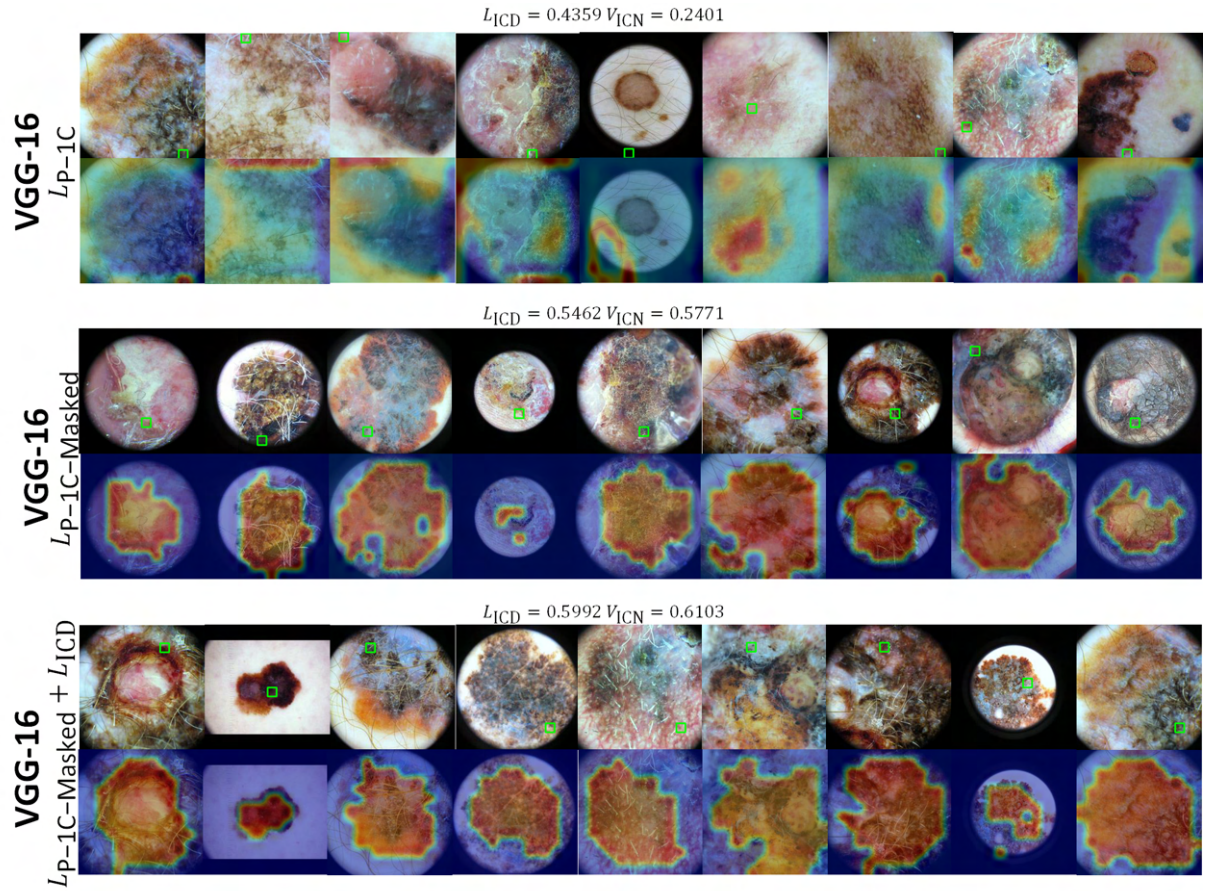


Figure A.19: Nine prototypes learned exclusively for the malignant class of melanoma for VGG-16 architecture: in three distinct scenarios, namely, $L_{\text{P-1C}}$, $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ (sections 3.2 and 5.2). The quantitative metrics for intra-class diversity, L_{ICD} and V_{ICN} are presented for each scenario.

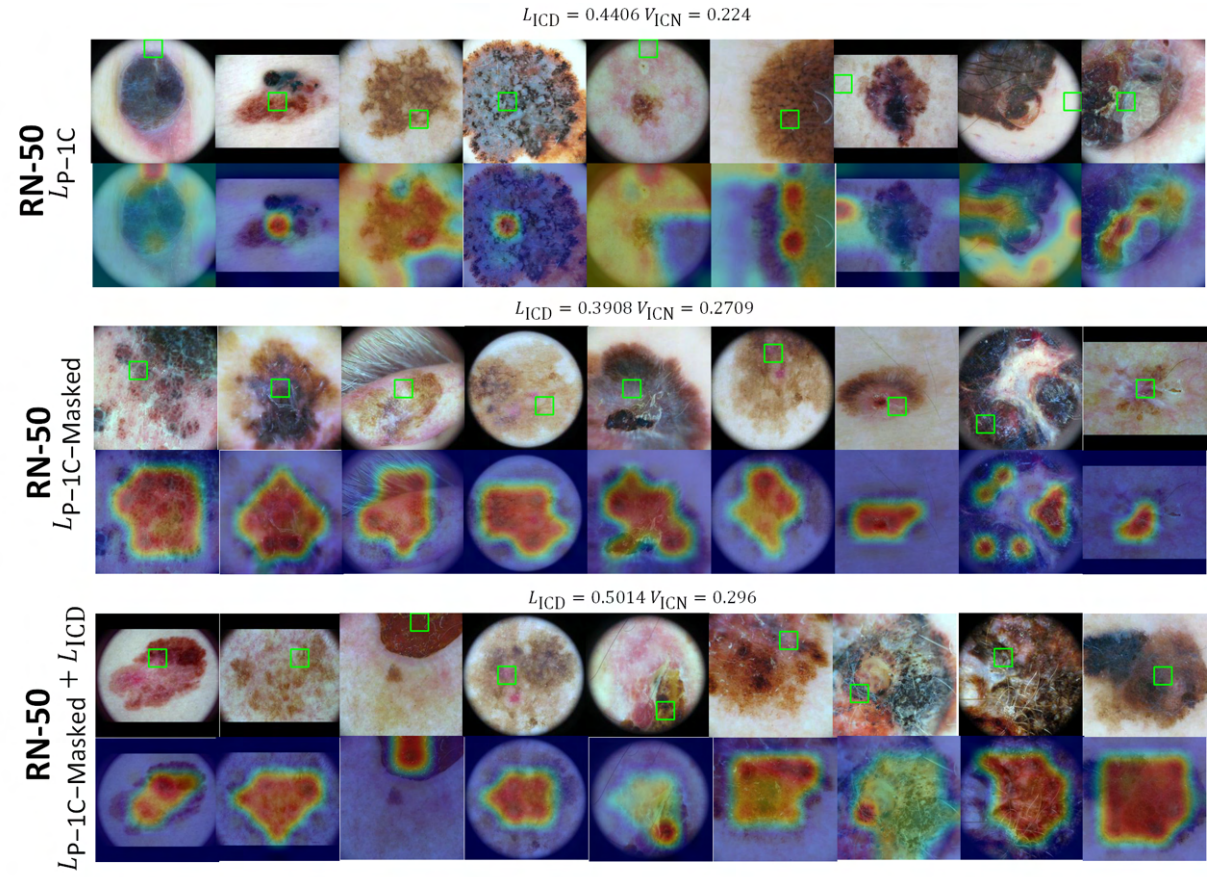


Figure A.20: Nine prototypes learned exclusively for the malignant class of melanoma for ResNet-50 architecture: in three distinct scenarios, namely, $L_{\text{P-1C}}$, $L_{\text{P-1C-Masked}}$ and $L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ (sections 3.2 and 5.2). The quantitative metrics for intra-class diversity, L_{ICD} and V_{ICN} are presented for each scenario.

Table A.3: Impact of prototype removal in terms of BA on the ISIC 2019 [1–3] validation set for the interpretable scenario with prototype-level supervision and promotion of intra-class diversity $L_{\text{P-1C–Masked}} + L_{\text{ICD}}$ for the binary problem of melanoma vs. nevus, see section 3.2. There are 9 prototypes of the melanoma class. The average difference in BA when each prototype is individually removed is denoted by μ, and the standard deviation by σ.

Model	**Scenario:$L_{\text{P-1C-Masked}} + L_{\text{ICD}}$ Difference in BA on the ISIC 2019 dataset for each removed prototype**									
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	$\mu \pm \sigma$
RN-18	-4.8	-4.8	-4.1	-4.1	-5.1	-5.7	-4.	-3.8	-3.8	-4.5 $\pm$ 0.6
RN-50	-3.4	-2.8	-4.9	-4.4	-2.8	-2.1	-2.0	-1.9	-2.4	-3.0 $\pm$ 1.0
EN-B3	-3.6	-3.0	-3.0	-4.2	-2.5	-3.5	-1.8	-4.2	-3.5	-3.3 $\pm$ 0.7
DN-169	-4.0	-4.0	-4.0	-3.5	-2.4	-2.6	-1.9	-3.9	-1.0	-3.0 $\pm$ 1.0
VGG-16	-4.0	-5.8	-7.0	-5.1	-6.0	-5.0	-5.2	-6.8	-5.9	-5.6 $\pm$ 0.9

A.6 Additional information for section 5.3

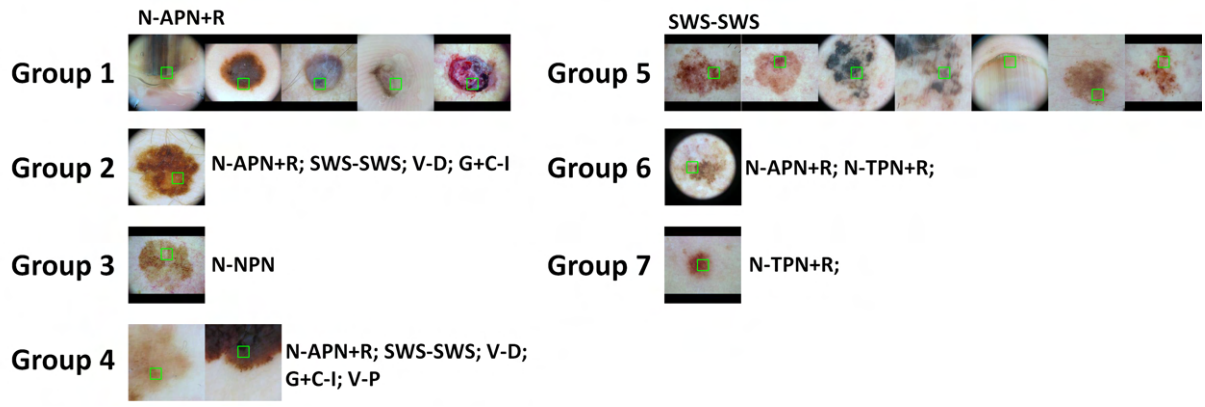


Figure A.21: Seven prototype groups pertaining to the $L_P + L_M$ scenario in the binary problem for the EN-B3 architecture. Each group represents a set of prototypes that, according to the analysis conducted in section 5.3, exhibit the same concepts. The first four groups pertain to melanoma prototypes, while the remaining ones pertain to nevus prototypes. The identified concepts are as follows: "Network - Atypical pigment network + Reticulation" (N-APN+R), "Globules + Clods - Irregular" (G+C-I), "Shiny white structures - Shiny white streaks" (SWS-SWS), "Vessels - Dotted" (V-D), "Vessels - Polymorphous" (V-P), "Network - Negative pigment network" (N-NPN),"Network - Typical pigment network + Reticulation" (N-TPN+R).

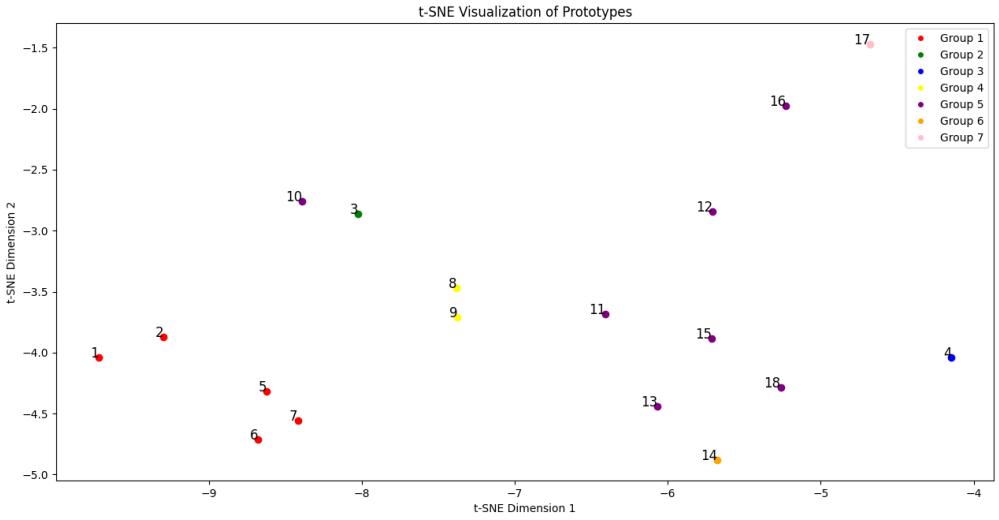


Figure A.22: Representation of the 18 prototypes in a 2D space using t-SNE, with 9 melanoma prototypes indexed from 1 to 8 and 9 nevus prototypes indexed from 10 to 18. The prototypes are highlighted with colors corresponding to the groups identified in the analysis conducted in section 5.3. Prototypes belonging to the same group and therefore sharing the same color exhibit identical dermatological concepts, as illustrated in fig. 5.8. The t-SNE parameters used were as follows TSNE(n_components=2, perplexity=10, n_iter=1000).

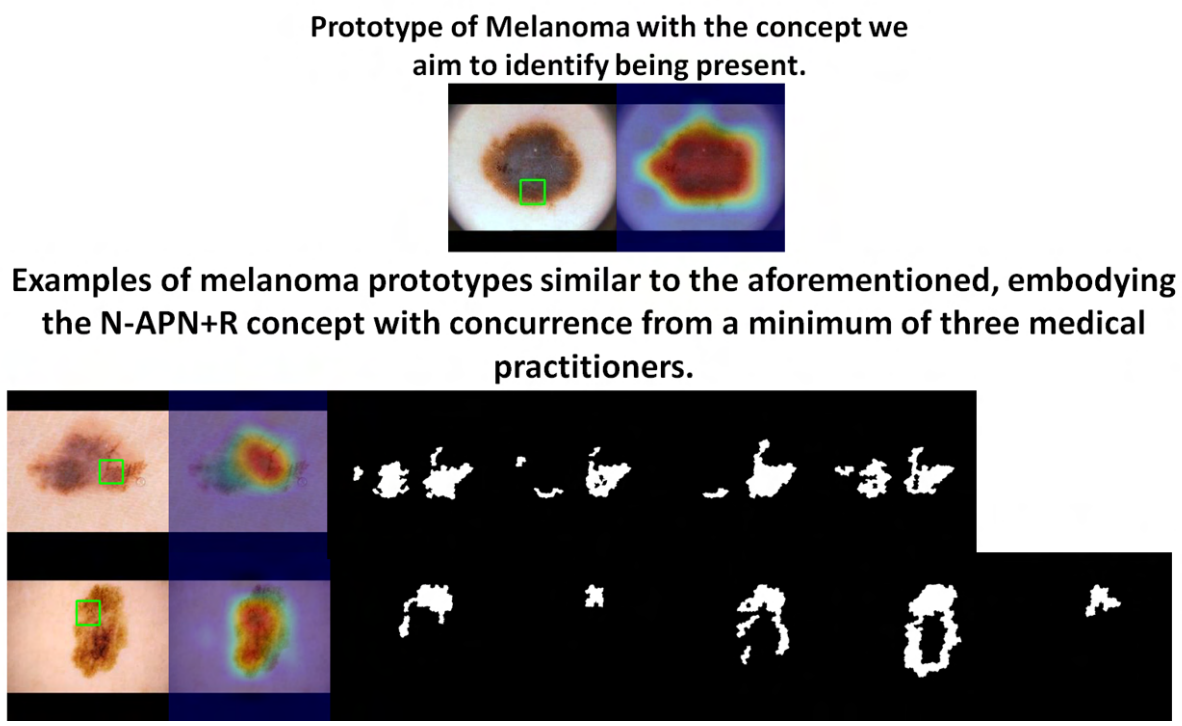


Figure A.23: Example of a prototype where the concept ”Network - Atypical pigment network + Reticulation” (N-APN+R) was identified, along with two of the 5 most similar images from the EASY Dermoscopy Expert Agreement Study dataset that are closest to this prototype. Additionally, the respective annotations made by dermatologists were examined, revealing a minimum agreement rate of 3 regarding the presence of the concept. Notice how the patch outlined by the green square in the figures annotated by the doctors exhibits the concept and is strongly activated by the prototype.

A.7 Additional information for section 5.4

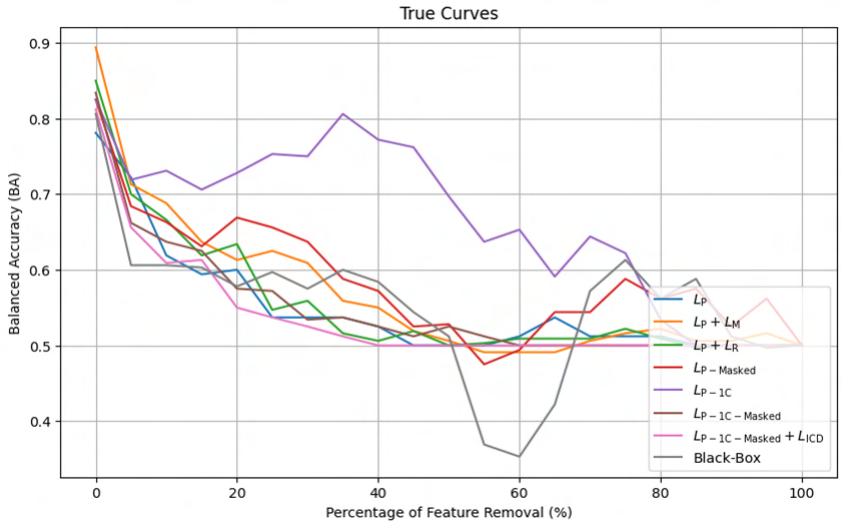


Figure A.24: Evolution of BA occurs by gradually removing important pixels from the images, starting from 0% and increasing in 5% increments up to 100%. This process was carried out for various interpretable scenarios, including L_P, $L_P + L_M$, $L_P + L_R$, $L_{P\text{-Masked}}$, $L_{P\text{-1C}}$, $L_{P\text{-1C-Masked}}$, and $L_{P\text{-1C-Masked}} + L_{ICD}$. Additionally, there was a non-interpretable scenario referred to as the black-box. This was performed for the binary problem. The importance of pixels was determined based on appropriate and non-randomly generated activation maps A_i, or heatmap H_i in the case of the black-box scenario. The test set used for this evaluation was PH^2 [15]. The CNN backbone utilized in this study was EfficientNet B3. These curves were employed to assess the guideline G3-truthfulness [16].

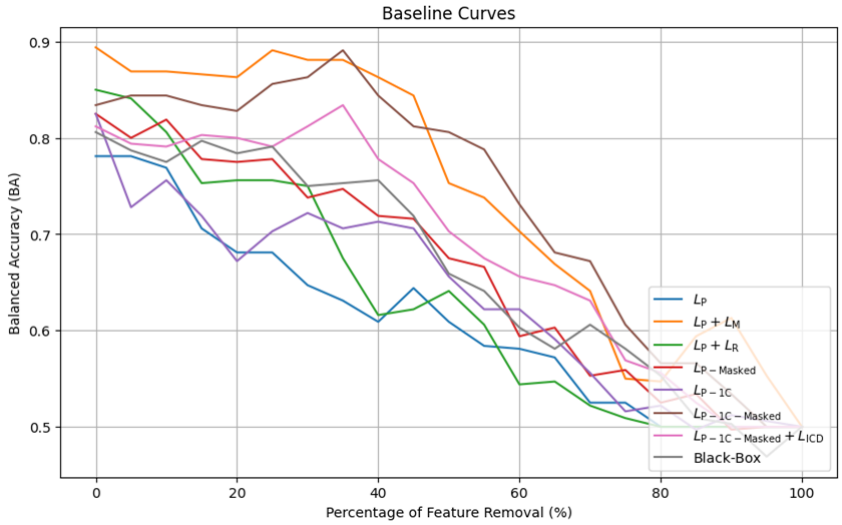


Figure A.25: Evolution of BA occurs by gradually removing important pixels from the images, starting from 0% and increasing in 5% increments up to 100%. This process was carried out for various interpretable scenarios, including L_P, $L_P + L_M$, $L_P + L_R$, $L_{P\text{-Masked}}$, $L_{P\text{-1C}}$, $L_{P\text{-1C-Masked}}$, and $L_{P\text{-1C-Masked}} + L_{ICD}$. Additionally, there was a non-interpretable scenario referred to as the black-box. This was performed for the binary problem. The importance of pixels was determined based on randomly generated baseline heat maps B_i. The test set used for this evaluation was PH^2 [15]. The CNN backbone utilized in this study was EfficientNet B3. These curves were employed to assess the guideline G3-truthfulness [16]