Skin Cancer Diagnosis Using Dermoscopic Images and Patient Information

Leandro José Pereira de Almeida

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

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

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Dra. Ana Catarina Fidalgo Barata
Member of the Committee: Prof. Paulo Luís Serras Lobato Correia

October 2020

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.

Acknowledgments

I would like to thank my parents and my sister for their friendship, encouragement and support throughout all my career, without whom it would not be possible to go through tough moments. I would also like to thank my parents for giving me the means to obtain the best academic education.

To my supervisors, Professor Jorge Marques and Dr. Catarina Barata, for all their support, guidance, assistance, critical feedback during this project. Their help and availability for me were instrumental in keeping me motivated throughout all the project.

I would like to thank all my friends and colleagues that helped me and followed me during these last 5 years, namely to Trivial group. Without their friendship, support, all the advice, and motivation to finish all the projects, this first adventure together would not be as gratifying as it was. They made the course seem easier than it is. My Erasmus friends also deserve thanks for the excellent semester they provided me.

Last but not least, to Beatriz Travassos, who always believed in me, gave me strength to achieve all the success, and was always there for me during the good and bad times.

Abstract

Skin cancer is the most common type of cancer worldwide. Early detection leads to an increased survival rate. Computer-Aided Diagnosis (CAD), which processes dermoscopic images, can improve the early detection rates.

In recent years, different CAD systems have been developed. However, almost all of these systems ignore additional patient metadata (e.g., age, region of the body, and gender), which is also taken into account by dermatologists when diagnosing the lesions.

This thesis aims to answer the following question: ”Does combining patient information with dermoscopic images for skin lesion diagnosis lead to further improvements over just dermoscopic images?”. The goal is to understand if there are any performance improvements when incorporating the patient’s clinical information (age, sex, body region) in the decision system. Thus, different strategies based on Deep Neural Networks, that combine these covariates with images, are proposed. These strategies are compared against models trained just with images.

Experiments conducted on the ISIC 2019 dataset verified that metadata improves the results, since the strategies that incorporate patient’s metadata reach a higher Balanced Accuracy (BACC). The best-evaluated configuration achieved a BACC of 77.76% for the validation set and 56.01% for the test set, and it led to an improvement of 3.14% and 3.79%, respectively, over the model without metadata. In this configuration, the fusion of the image network and the metadata network is performed by multiplying their outputs.

Lastly, the relevance of each combination of metadata is explored, and a website application is developed to be used by dermatologists.

Keywords

Skin Lesion Diagnosis, Computer-aided Diagnosis, Dermoscopic Images, Deep Neural Networks, Metadata

Resumo

O cancro de pele é o tipo de cancro mais comum em todo o mundo. A deteção precoce leva a um aumento da taxa de sobrevivência. O diagnóstico assistido por computador, que processa imagens dermoscópicas, pode levar a melhorar as taxas de deteção precoce.

Nos últimos anos, diferentes sistemas de diagnóstico assistidos por computador foram desenvolvidos. No entanto, quase todos estes sistemas ignoram os metadados do paciente (por exemplo, a idade e o género), que são tidos em consideração pelos dermatologistas no diagnóstico.

Esta tese tem como objectivo responder à seguinte questão: ”A utilização de informações clinicas do paciente com imagens dermoscópicas para o diagnóstico de lesões de pele pode levar a melhorias em relação ao uso de apenas imagens dermoscópicas?”. O objetivo é entender se há melhorias ao incorporar covariáveis clínicas (idade, género e região corporal) no sistema de decisão. Assim, diferentes estratégias baseadas em redes neuronais profundas, que combinam essas covariáveis com imagens, são testadas. Essas estratégias são comparadas com vários modelos treinados apenas com imagens. Experiências feitas no conjunto ISIC 2019 demonstram que os metadados melhoram os resultados. A configuração com melhor desempenho atingiu uma BACC de 77.76% no conjunto de validação e 56.01% no conjunto de teste, e levou a melhorias de 3.14% e 3.79%, respectivamente, em relação ao modelo sem metadados. Nesta configuração, a fusão da rede de imagens e dos metadados é feita multiplicando as suas saídas.

Por fim, é estudada a influência de cada combinação de metadados e é apresentado um site desenvolvido para ser usado por dermatologistas.

Palavras Chave

Diagnóstico de Lesões de Pele, Diagnóstico Assistido por Computador, Imagens Dermoscópicas, Redes Neuronais Profundas, Metadados

Contents

List of Figures

List of Tables

Acronyms

Adam	Adaptive Moment Estimation
AK	Actinic Keratosis
BACC	Balanced Accuracy
BCC	Basal Cell Carcinoma
BKL	Benign Keratosi
CAD	Computer-Aided Diagnosis
CNN	Convolutional Neural Network
DIA	Dermoscopy Image Analysis
DF	Dermatofibroma
FCL	Fully-Connected Layer
FN	False Negative
FP	False Positive
ISIC	International Skin Imaging Collaboration
MEL	Melanoma
NV	Melanocytic Nevus
ReLU	Rectified Linear Unit
SCC	Squamous Cell Carcinoma
SE	Sensibility
SP	Specificity
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VASC	Vascular Lesion

1 Introduction

Contents

1.1 Motivation

Skin cancer is the most common type of cancer worldwide, and the number of cases and deaths has been increasing in the past years [11]. The World Health Organization (WHO) estimates that one in three diagnosed cancers is skin cancer [12]. Skin cancer can be divided into non-melanoma and melanoma. Melanoma is less common but more dangerous. Globally, between 2 and 3 million non-melanoma and 132,000 melanoma skin cancers are detected each year [12]. In Portugal, approximately 700 cases of melanoma occur annually [13]. According to the Skin Cancer Foundation [11], in the U.S, every day, more than 9,500 people are diagnosed with skin cancer. It is estimated that one in every five will develop skin cancer by the age of 70 [11].

Early detection and treatment are critical to reducing the mortality rate of this disease, as early detection leads to an increased survival rate. When melanoma is detected on an early stage, the 5-year survival rate is 99% [14]. However, this value drops to about 14% if detected in its latest stages. Despite the early detection and diagnosis of melanoma can increase the survival rate of patients with the disease, the diagnostic accuracy of melanoma from visual inspection is only about 60% [15].

The diagnosis of melanoma can benefit from image analysis and machine learning methods to increase the diagnostic accuracy. Since pigmented lesions occur on the surface of the skin, melanoma is susceptible to early detection with expert visual inspections or with automated detection (image analysis) [16]. CAD, which can take advantage of dermoscopic images from high-resolution cameras, can allow doctors and patients to detect skin lesions earlier and can be of great value in reducing the number of deaths.

Recently, deep learning models have been achieving good results in different medical image analysis tasks. Convolutional Neural Network (CNN) models have become the main approach to solving this kind of problem. The evolution of CNNs for classification problems is linked to the ImageNet challenge [17]. The good results achieved by the deep learning models also extend to skin cancer detection, since these models have also been adopted to tackle skin cancer classification, based on dermoscopic images. Significant improvements already done are linked with the International Skin Imaging Collaboration (ISIC) challenge [16].

In addition to dermoscopic images, patient's information (such as the patient's age, gender, anatomic site, family history, among others) is also taken into account by dermatologists when diagnosing the lesions [18]. However, these covariates have been scarcely used in CAD systems [19]. Therefore, it is crucial to know whether this information is an important clue to be incorporated in a CAD system to achieve a more accurate diagnosis. Taking into consideration not only dermoscopic images but also patient information, it may be possible to build a more robust system. This system can help to act as a quick and efficient diagnostic tool to help doctors to detect and treat cancerous patients earlier and help to save many lives.

1.2 Medical diagnosis of skin lesion

Skin cancer is the most common cancer, with melanoma being the most deadly form. Skin lesions can be divided into non-melanocytic and melanocytic, if it is formed from other cells or melanocytes, respectively [1]. They can be further grouped into benign and malignant. As far as melanocytic lesions are concerned, the malignant lesion is Melanoma (MEL), and benign is Melanocytic Nevus (NV). With regard to non-melanocytic skin lesions, the malignant lesions are Basal Cell Carcinoma (BCC), Squamous Cell Carcinoma (SCC) and Actinic Keratosis (AK). The benign lesions are Dermatofibroma (DF), Benign Keratosi (BKL) and Vascular Lesion (VASC) [1] [11]. This hierarchy is outlined in fig. 1.1.

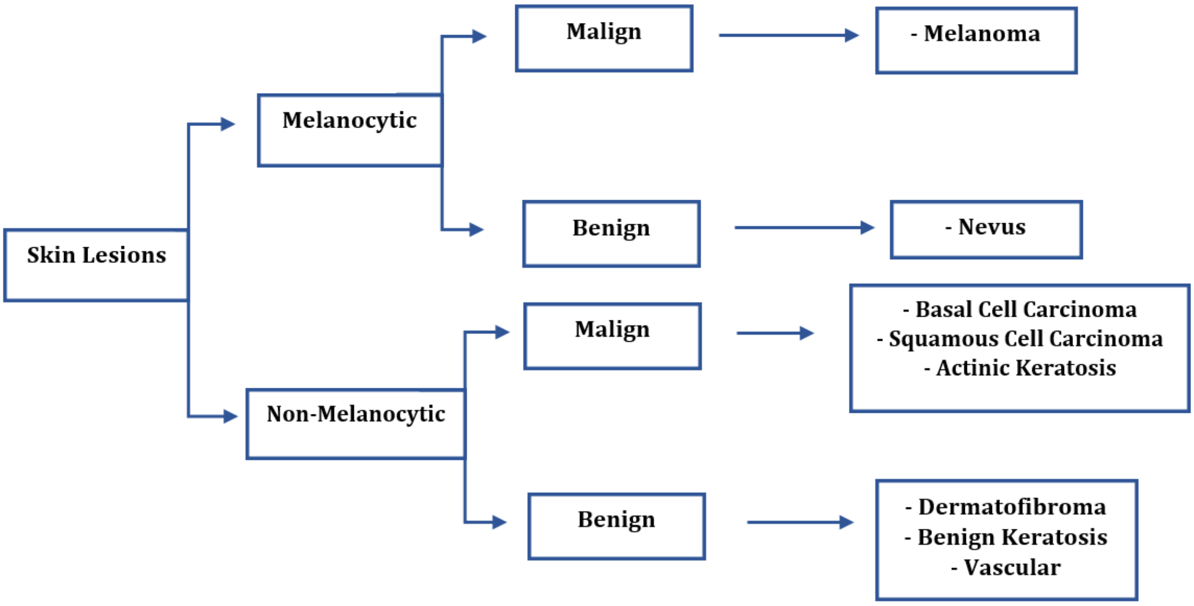


Figure 1.1: Skin lesions hierarchy [1].

Dermatologists use dermoscopy to recognize several surfaces and subsurface structures, which are not visible to the naked eye, and which can be used to diagnose skin lesions [20]. Dermoscopy is an imaging technique that removes the surface reflection of the skin. Thus, the visualization of more profound levels of skin is improved [16]. Dermoscopy leads to an improvement of diagnostic accuracy, in relation to standard photography. Nevertheless, training a dermatologist takes a long time. It also poses a challenge due to the enormous similarity among the different skin lesions. Several medical methods are used to diagnose dermoscopic images, such as pattern analysis, 7-point checklist, Menzies method, and ABCD rule [19]. The first model identifies all the possible criteria and their density inside the lesion. On the other hand, the 7-point checklist and the Menzies method focus only on the criteria associated with melanoma. Finally, the ABCD rule combines the identification of dermoscopic criteria with a global lesion analysis. This analysis takes into account some factors like border sharpness, lesion architecture, color distribution, and the degree of asymmetry [19].

1.3 CAD Systems for Skin Lesion Diagnosis

Medical methods are very subjective. To overcome the limitations of medical diagnosis, CAD systems, based on dermoscopic images, can be used to act as a second opinion tool.

According to [21], the three primary methods of Dermoscopy Image Analysis (DIA) are segmentation (lesion border detection), feature extraction, and classification (different machine learning methods may be used). From the 1970s to the 1990s, when it started to be possible to scan and load medical images into a computer, most researchers have built systems for automated analysis, using low-level image processing methods (edge and line detection, and region growth) and simple mathematical modelings, such as line, circle, and ellipse fitting. The goal of these mathematical models was to build rule-based systems for specific image analysis tasks [22].

Thereafter, supervised classification was used to tackle this problem. Methods based on Machine Learning were introduced in clinical practice and have become the main approaches. Decision trees, Bayesian classifiers, Support Vector Machine (SVM), and artificial neural networks have been used for the diagnosis task [19].

Nevertheless, these classical machine learning techniques required the extraction of handcrafted features. These features were obtained with image processing methods (for example, algorithms that automatically compute the colors of the image). In order to overcome this problem, data-driven CNN models have been used in recent years. Computational techniques that can automatically extract and learn high-level features from images (without previous dermatologists analysis) were developed, providing greater robustness.

CNN models have proven to be effective techniques for skin cancer diagnosis using dermoscopic images. The use of CNNs in dermoscopy is related to the increase in the number of public datasets. The most famous dataset for skin cancer diagnosis is the ISIC dataset. Every year, since 2016, ISIC [16] offers a large dataset of dermoscopic images and promotes a challenge. This archive is the largest publicly available collection of dermoscopic images of skin lesions. The goal of the recurring challenge is to help participants develop image analysis tools to enable the automated diagnosis from a dermoscopic image. One of the tasks is lesion diagnosis classification. In the ISIC 2017 challenge, a ResNet architecture was used in [23]. In 2018, this challenge resulted in several high-performance methods based on CNNs that performed similarly to human experts for the evaluation of dermoscopic images [24]. In the ISIC 2018, a DenseNet 201 was used in classification task in [25]. Several works have used ensemble methods, which combine different architectures. For instance, in [26] an ensemble consisted with ResNet 50, Inception v3, Xception, DenseNet 201 and InceptionResNet v2 was applied. The 2018 challenge winner [27] has also used an ensemble approach. Before feeding the images to the architecture, different pre-processing methods are often used to increase the accuracy and to tackle some generalization problems.

Recently, studies that combine images with the patient's clinical information have started to appear (for example, in [28] and [15]). In 2019, to further improve the diagnostic performance, the ISIC released a new task that aimed at performing lesion diagnosis with dermoscopic images and metadata [16]. In this challenge, the image's information was completed with the patient's information. The winner of the challenge with an ensembling strategy was Gessert [24]. This work combined the images network with the metadata network by concatenating outputs at the feature level. A similar approach was used by other challenge participants. However, it is not yet clear whether metadata helps or not to improve the diagnosis. This leads to the challenge of this thesis: understand if the patient's information is beneficial to skin lesion classification. Moreover, it is also necessary to understand what is the best strategy for combining metadata with images, and this study is missing in the literature. Both questions motivated this thesis, which is a new contribution to literature.

1.4 Objetives

To diagnose the skin lesion of a given patient, the dermoscopic image of the lesion and the patient's information can be used. In this thesis, the patient's information considered is the age, gender, and the anatomical site. An example of a patient record presented in the dataset contains a dermoscopic image illustrated in fig. 1.2. The corresponding patient's information is: age between 45 and 50 years old, the gender is female, and the anatomic site is the posterior torso.

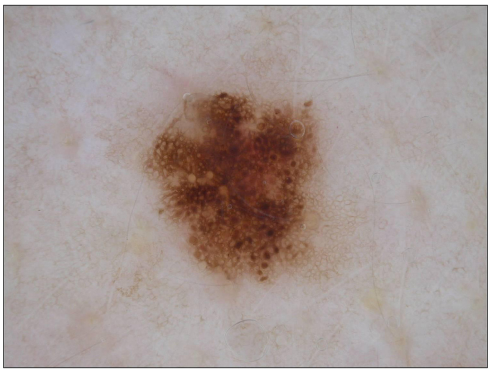


Figure 1.2: Example of a dermoscopic image presented in the dataset.

The main goal of this thesis is to answer the question: "Does combining patient information with dermoscopic images for skin lesion diagnosis lead to further improvements over just dermoscopic images?". In other words, it aims to understand if there are any performance improvements when incorporating the patient's information (age, sex, body region) in the decision system. To answer this question, different strategies that include these covariates with images are proposed and compared. These strategies are also compared against models trained just with images. The relevance of each combination of metadata is also explored (to check which combination has the most influence on the classification) separately, by training a selected architecture with all the different possible combinations of metadata features.

Lastly, a website application will be developed to be used by dermatologists, where the main goal is that they can upload an image and insert the patient's information, and immediately receive the skin lesion classification.

1.5 Organization of the Document

This thesis is organized as follows. Chapter 2 focuses on the important background of CNNs, as well as some popular CNN architectures that have performed well in image classification at the ISLRVC challenge. Chapter 3 introduces the dataset as well as an analysis of the available patient's clinical information. This analysis includes graphical analysis. Chapter 4 presents all of the methods and techniques used to classify the skin lesions with and without metadata. Chapter 5 shows the experimental evaluation of the proposed methods and draws some conclusions. Chapter 6 introduces a website application. Lastly, chapter 7 presents the conclusions and future work.

2

Background

Contents

This chapter presents an explanation about CNNs in general and the training of the models. Afterward, some popular architectures that outperformed in a large scale image classification task (ImageNet) are described and compared.

2.1 CNN concepts

A CNN is a class of deep neural networks used with several image-related problems. A CNN allows the extraction of features by applying convolutional operators that progressively learn more abstract features.

CNN comprises convolutional layers, Fully-Connected Layer (FCL) layers, and pooling layers.

Figure 2.1 aims to exemplify the basic blocks of a CNN. It classifies a 24×24-pixel grayscale image into two categories, y_1 and y_2. The model consists of two convolution layers and two pooling layers. The output of the last pooling layer is fed into a fully-connected layer and followed by the output layer that produces the classification [2].

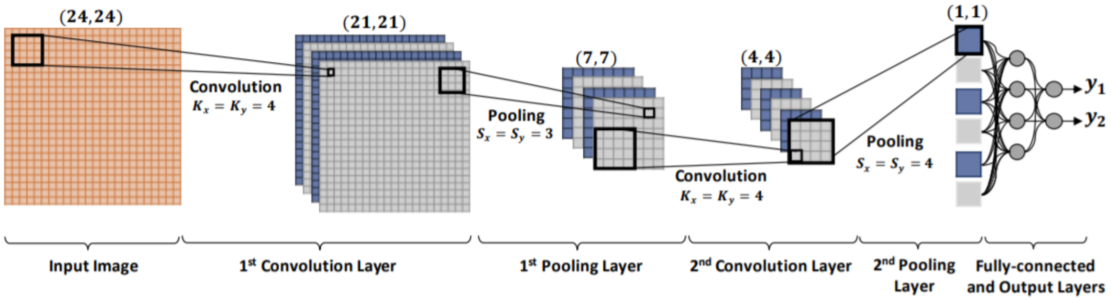


Figure 2.1: The illustration of a CNN composed of 2 convolution layers, 2 pooling layers and one FCL. Image retrieved from [2].

2.1.1 Convolutional Layer

The main building block of a CNN is the convolutional layer [29]. A convolutional layer is composed of a set of convolutional kernels/filters. The input image is converted into feature-maps, using the convolution operation. Each feature-map represents the output of the convolutional operation between the input and a given kernel. In fully-connected layers, neurons can be represented as vectors. In convolutional layers, kernels can be represented as a 3-dimensional tensor (with shape equal to width × height × number of channels) [29]. Each kernel has a specific width and height but has a depth equal to the number of channels of the input. If the input is an RGB image, it will have 3 channels (red, green, blue), and the kernels used in the convolutional layer have a depth equal to 3.

In CNNs, neurons that belong to different kernels detect different image features. On the other hand, neurons presented on the same kernel detect the same image feature, at different spatial locations [29].

Each kernel slides along the spatial dimensions of the input tensor with a certain stride, and it continues until the filter can not slide further [30]. At each location, the kernel computes dot products. The resulting value is placed in the filtered image (it is just one pixel of the resulting feature-map) [29]. This kernel is evaluated at every possible location, resulting in a 2-dimensional feature-map.

By applying several kernels in the same convolutional layer, the output of the convolutional layer is a stack of feature-maps [29]. The depth of this stack is equal to the number of kernels used. If a given convolutional layer contains 2 kernels, 2 feature-maps are generated. Each feature-map is a new image, and a nonlinear activation function is applied to each pixel of the feature-map. If a CNN is composed of several convolutional layers, the input of the next layer is the output of the last one.

2.1.2 Pooling layer

The role of the convolutional layer is to extract and detect local incidences of features from the previous layer. On the other hand, the pooling layer merges similar features into one, since the relative positions of the similar features can vary somewhat [31]. A pooling layer operates on blocks of the feature-map and combines the feature activations [30]. The size of the pooled region (a window with width and height) and the stride need to be specified. Thus, the pooling layer reduces the spatial size of the image (it reduces the width and the height but the depth remains the same), while retaining the most important information [29]. The pooling operation works as follows: a window slides across the input feature-map with a specific stride [30], and for each location, it combines the neighboring pixels of the image into a single representative value (this output value is usually the average or maximum within the window). The output size depends on the width and height from both feature-map and the filter, and the stride. The depth is the same as the feature-map depth. It is highly beneficial to include pooling layers for relieving the computational load.

2.1.3 Fully Connected Layer

After convolutions and pooling layers, a CNN has FCL layers. The output of the convolutions and pooling layers are fed in one or more fully connected layers [32]. In FCL each neuron is connected to all the input units. The input of the first FCL is a one-dimension vector, results from a flattening operation. For instance, if the previous layer has a size $7 \times 7 \times 2048$, it can be flattened in an array of size $7 \cdot 7 \cdot 2048$. Another way to flatten is to apply a global average pooling layer before the FCL. If a global spatial average is applied, the $7 \times 7 \times 2048$ layer will be transformed into a one-dimension vector of size 2048. The output of the FCL is a vector of size equal to the number of neurons of the layer, resulting in a linear combination of the input with weights. It can be represented as a multiplication followed by adding a vector of bias terms and applying an element-wise nonlinear activation function f [30]. It is given by:

$$y = f(W^T x + b) \tag{2.1}$$

where f is the activation function, x is the input flatten vector, y the output vector, W the weight's matrix, and b the bias term vector [30]. The last FCL is used to predict the class label [32]. This layer has M neurons, in order to generate a vector of size M (where M is the number of classes) that gives the final probability for each label.

2.1.4 Activation Functions

The purpose of the activation function is to introduce a nonlinear behavior into the network, and it allows a neural network to learn nonlinear mappings [30]. A nonlinear function can be referred to a switching, which decides whether a neuron will fire depending on the inputs [30]. The activation functions used in deep learning are differentiable in order to allow the backpropagation optimization [30]. The activation functions are applied to convolutional layers and FCL. The most popular nonlinear function is the Rectified Linear Unit (ReLU) [31], since it helps in overcoming the vanishing gradient problem and allows the network to converge very quickly. ReLU is defined by:

$$\text{ReLU}(z) = max(0, z). \tag{2.2}$$

Other activation functions are commonly used, such as $tanh(z)$ and $sigmoid$. $Tanh(z)$ is given by:

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{2.3}$$

$Sigmoid$ is given by:

$$sigmoid(z)\frac{1}{1 + e^{-z}}. \tag{2.4}$$

Nevertheless, the ReLU typically learns much faster in networks with many layers [31].

In the output of the FCL is common to use a *Softmax* activation function. In *Softmax*, the sum of the outputs is equal to 1 and, therefore, it can be in interpreted as a probability distribution. The *Softmax* activation function, $\sigma(x)$ (with M classes and x the vector of inputs with size M), is given by:

$$\sigma(x)_i = \frac{e^{x_i}}{\sum_{k=1}^{M} e^{x_k}}, i = 1, 2, ...M, \tag{2.5}$$

where $\sigma(x)_i$ represents the probability to belong to the class i.

2.2 Training the model

In supervised methods, the estimation of the network parameters assumes that the input-output pairs are known (training set). A loss function is used to evaluate the quality of predictions made by the network on the training data [30].

During the training, the main goal is to minimize the loss function, which computes the difference between the network's output and the ground truth. There are different loss functions to perform this task. Categorical cross-entropy is the most common loss function in classification problems. This function measures the difference between two probability distributions (the network's output and the ground truth). For a single observation o, the cross-entropy loss function is given by:

$$-\sum_{c=1}^{M} y_{o,c} log(p_{o,c}), \tag{2.6}$$

where M is the number of classes, y is an binary indicator (that it is equal to 1 if class label c is the correct classification for observation o), and p is the probability predicted by the model for observation o with respect to class c [30]. Cross-entropy loss increases as the network's output diverges from the ground truth. A perfect model would have a loss of 0.

The parameters of the network are optimized with the gradient descent method. The general equation is given by:

$$\theta_t = \theta_{t-1} - \eta \frac{\partial L}{\partial \theta}, \tag{2.7}$$

where θ_{t-1} represents a network parameter at step $t-1$, θ_t is the update at step t, η is the learning rate, and $\frac{\partial L}{\partial \theta}$ is the backpropagated gradient of a loss function with respect to the trainable parameters [33].

During the train, the gradient, $\frac{\partial L}{\partial \theta}$, is computed using the backpropagation method, which is a practical application of the chain rule [31]. Backpropagation involves forward and backward steps. In the first, the input is forward through the network, and it outputs a predicted value. After computing the loss function based on the predicted value, the backward steps are performed (by using the chain rule) to compute the gradient, and the weights are further updated with the chosen optimizer, in order to reduce the value of the loss function [31]. The optimizer defines the way that the weights are updated in order to minimize the loss function. Different variants of the gradient descent are used as optimizers, such as: Stochastic Gradient Descent (SGD), SGD with momentum, Adaptive Moment Estimation (Adam), Adaptive Delta (AdaDelta), etc. [30]. Adam is the most common optimizer. It uses estimations of the first and second moments of the gradient to apply an individual adaptive learning rate for each parameter. As it is shown in [34], in the step t, each parameter is updated as:

$$\theta_t = \theta_{t-1} - \alpha_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{2.8}$$

where α_t is the global learning rate, ϵ a small constant and $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected estimators for the first and second moments of the gradient, in the step t. This algorithm is computationally efficient with little memory requirements and is suitable for large problems (large dataset or a large number of parameters) [34].

2.3 Popular CNN architectures

Since 2010, the ImageNet Large Scale Visual Recognition Challenge serves as a benchmark for object category classification and detection on hundreds of object categories and millions of images [17]. The database used in the challenge contains millions of images belonging to 1000 classes [17]. The development of popular CNN architectures for classification is often linked with this challenge.

With the recent availability of larger datasets of images (mainly because of the ImageNet challenge), CNN models have achieved significant improvements compared with traditional shallow methods in image classification.

In recent years, most of the innovations in CNN architectures have been made in relation to depth and width, resulting in models a with different number of parameters. The most popular architectures are those that participated in the ImageNet challenges. Based on [35], table 2.1 summarizes some architectural details. The results report the performance on the ImageNet challenge. The top-5 error on the validation set were taken from the ImageNet leaderboard [36] or from the respective papers. The results correspond to the best of each architecture, without ensembling methods (single model evaluation). The reported year corresponds to the participation in the ImageNet challenge and may not be the same as the year of publication of the associated paper.

Table 2.1: Comparison of the recent popular architectures that have participated in the ImageNet Challenge. Top-5 error on validation set.

Name	Year	Author	Nr. Parameters [Millions]	Depth	Top-5 Error(%)	Version
AlexNet	2012	Alex Krizhevsky et al. [3]	60	8	15.4	7 CNNs
VGG	2014	Karen Simonyan et al. [4]	144	19	8.0	VGG19
GoogleNet	2014	Christian Szegedy et al. [5]	4	22	6.67	
Inception	2015	Christian Szegedy et al. [7]	23.6	159	5.6	InceptionV3
ResNet	2015	Kaiming He et al. [6]	25.6	152	4.49	ResNet-152
Inception-ResNet	2016	Szegedy et al. [8]	55.8	572	4.9	Inception-ResNet V2
Xception	2017	Chollet et al. [9]	22.8	126	5.5	
DenseNet	2017	Gao Huang et al. [10]	25.6	190	6.12	DenseNet-264

In the following sections, we succinctly describe the most popular architectures.

2.3.1 AlexNet

AlexNet [3] won ImageNet challenge in 2012. This network has 60 million parameters and 650,000 neurons. It consists of eight layers: five convolutional layers and three fully-connected layers. In [3] the input image size used was 224×224×3. Figure 2.2 shows the AlexNet architecture.

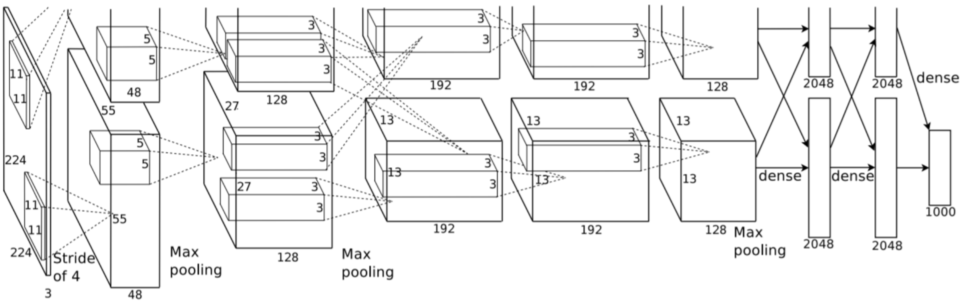
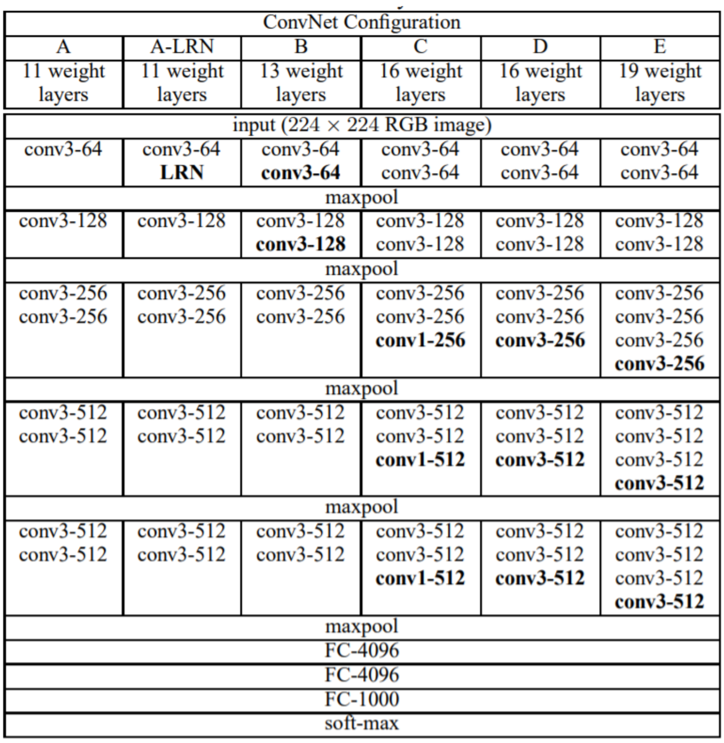


Figure 2.2: AlexNet architecture. Taken from [3]

2.3.2 VGG

VGG ranked second in the ImageNet challenge in 2014, showing that it is possible to train deeper networks to achieve better results. In [4] it is investigated the effect of the convolutional network depth on the accuracy in the large-scale image recognition. An architecture with very small (3×3) convolution filters were used, showing that significant improvements may be achieved by increasing the depth to 16–19 weight layers (more convolutional layers), with very small filters. The advantages of using stacks of convolution layers with small filters rather than using a single one with a relatively large support field are: the incorporation of few nonlinear rectification layers instead of a single one and the decrease in the number of parameters. In fig. 2.3 all the configurations studied in [4] are presented.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.3: ConvNet configurations (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). Image retrieved from [4].

2.3.3 GoogleNet

GoogleNet [5] won ImageNet challenge in 2014. This architecture is also based on very deep convNet and small filters, but it is more complex since it uses a new structure called inception module.

Because of the huge variation in the location of the information, the authors have realized that choosing the right kernel size for the convolution operation becomes tough. Larger filters spread out features of higher dimension. Small filters capture local details. Thus, instead of choosing one size for the filters in each layer, the inception module uses different size filters, as well as max-pooling. Afterwards, a concatenation of the feature-maps from each filter into one big feature-map is performed. The concatenation result is further fed to the next inception module. In fig. 2.4 the inception module (naïve version) is depicted.

2.3.4 ResNet

ResNet [6] revolutionized the CNN architectural race by introducing the concept of residual learning, to train even deeper networks. It was the winner of the 2015 challenge. With deeper networks, a degradation problem has been exposed: with the increase of network depth, accuracy gets saturated

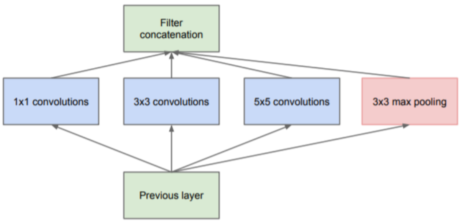


Figure 2.4: Inception Module. Taken from [5].

and then degrades rapidly. Therefore, adding more layers to a previous trained deep model leads to a decrease of the training accuracy [6]. In order to overcome this problem, the traditional convolution blocks were replaced by residual connections, using the blocks depicted in fig. 2.5.

ResNet has shown less computational complexity than previously proposed networks. In the ImageNet challenge, this network has achieved a top-5 error equal to 4.9% in single model evaluation. It is the best result depicted in table 2.1.

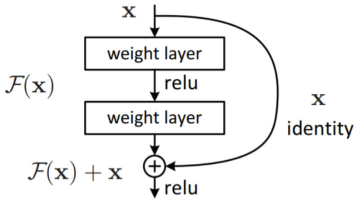

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

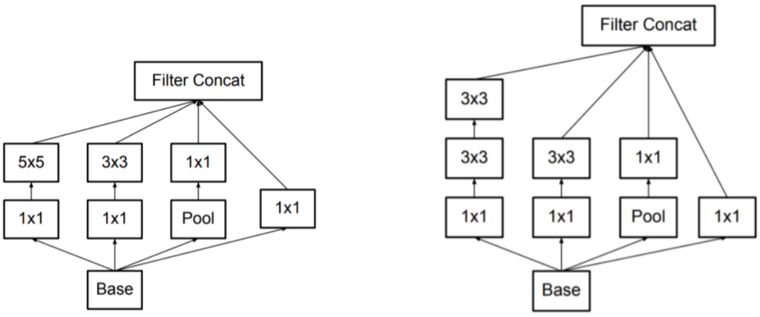
Figure 2.5: Residual learning: a building block. Image retrieved from [6].

2.3.5 Inception v2/v3

For scaling up convolution networks in efficient ways, dimensional reduction and parallel structures of Inception modules are introduced in [7]. Different approaches to factorize convolutions in various settings were performed, especially in order to increase the computational efficiency of the solution. The reduction in computational cost and a reduced number of parameters makes it possible to get faster training. In the first approach, a large convolutional layer is replaced by a multi-layer network with fewer parameters with the same input size and output depth. In fig. 2.6 is depicted a 5×5 convolution that is replaced by two layers of 3×3 convolutions.

In a further approach, asymmetric convolutions are used (replacing a n × n convolution by a 1 × n convolution followed by a n × 1 convolution). It is said in [7] that very good results can be achieved by using 1 × 7 convolutions followed by 7 × 1 convolutions.

Inception-V3 has reached a 5.6% top-5 error on ImageNet for single frame evaluation.



(a) Original module

(b) Module where each 5 × 5 convolution is replaced by two 3 × 3 convolution

Figure 2.6: Inception Module. Taken from [7].

2.3.6 Inception-ResNetV2

Inception-ResNetV2 [8] combines the Inception architecture with residual connections. This architecture has achieved a top-5 error equal to 4.9% on ImageNet. Using residual connections accelerates the training of Inception networks significantly. This allows Inception to have all the benefits of the residual approach while retaining its computational efficiency [8]. An example of a Inception-ResNet block used in Inception-ResNetV2 is depicted in fig. 2.7.

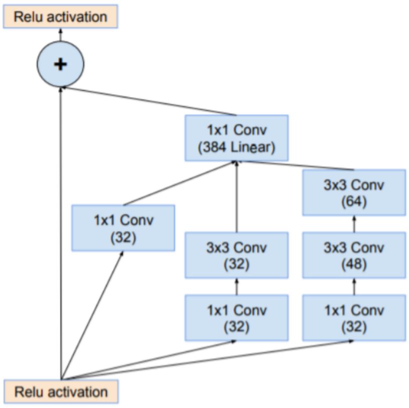


Figure 2.7: Schema for one Inception-ResNetV2 block. Taken from [8].

2.3.7 Xception

In [9] is proposed a convolutional neural network architecture based entirely on depthwise separable convolution layers - Xception. This architecture has achieved 5.5% top-5 error on ImageNet. It is a stronger version of the Inception architecture, which stands for “Extreme Inception” [9]. This architecture replaces the original Inception modules by an “extreme” version, which first applies a 1×1 convolution to map cross-channel correlations, and then separately maps the spatial correlations of every output channel (there is a spatial convolution per output channel of the 1×1 convolution). This module is shown in fig. 2.8.

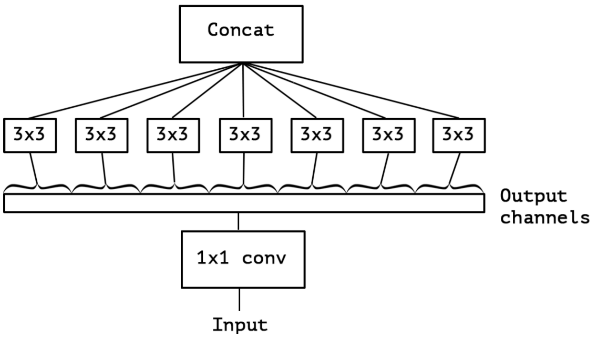


Figure 2.8: An “extreme” version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution. Taken from [9].

2.3.8 DenseNet

DenseNet was proposed to further improve the information flow between layers in deep networks [10]. To accomplish it all layers are connected (with matching feature-map sizes) directly with each other, as can it be seen in fig. 2.9.

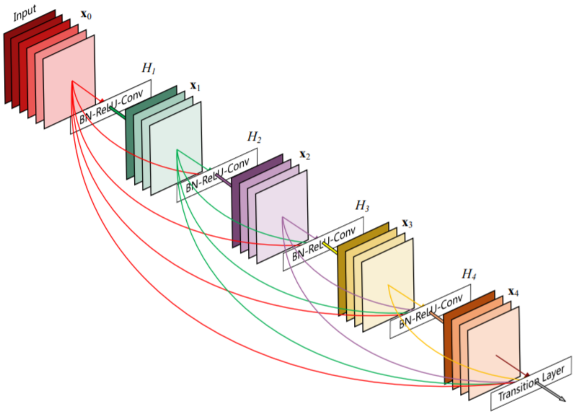


Figure 2.9: A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input. Taken from [10].

Each layer obtains additional inputs from all preceding and passes on its own feature-maps to all subsequent layers. Instead of combining features through summation before they are passed into a layer (characteristic of ResNet), in DenseNet the features are concatenated.

DenseNets has several advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

DenseNet has achieved a 6.12% top-5 error on ImageNet for single model evaluation.

3

Dataset Analysis

Contents

This chapter introduces the dataset (dermoscopic images and their corresponding metadata) and its division into the training and validation set. An exploratory analysis is carried out, focused on the patient's information. This analysis aims to identify relationships between the patient's information and the respective lesions.

3.1 Image dataset

ISIC has developed an international repository of dermoscopic images. The goal of this archive is to simultaneously support the development of automatic classification methods, as well as to aid in clinical training. The dataset used in this thesis was collected from the 2019 ISIC challenge [16]. Each record contains an image and the corresponding metadata. The dataset is composed of 8 types of skin lesions, which were introduced in section 1.2. These lesions are presented in section 1.2: MEL, NV, BCC, AK, BKL, DF, VASC and SCC. Figure 3.1 shows an example of each type of skin lesion.

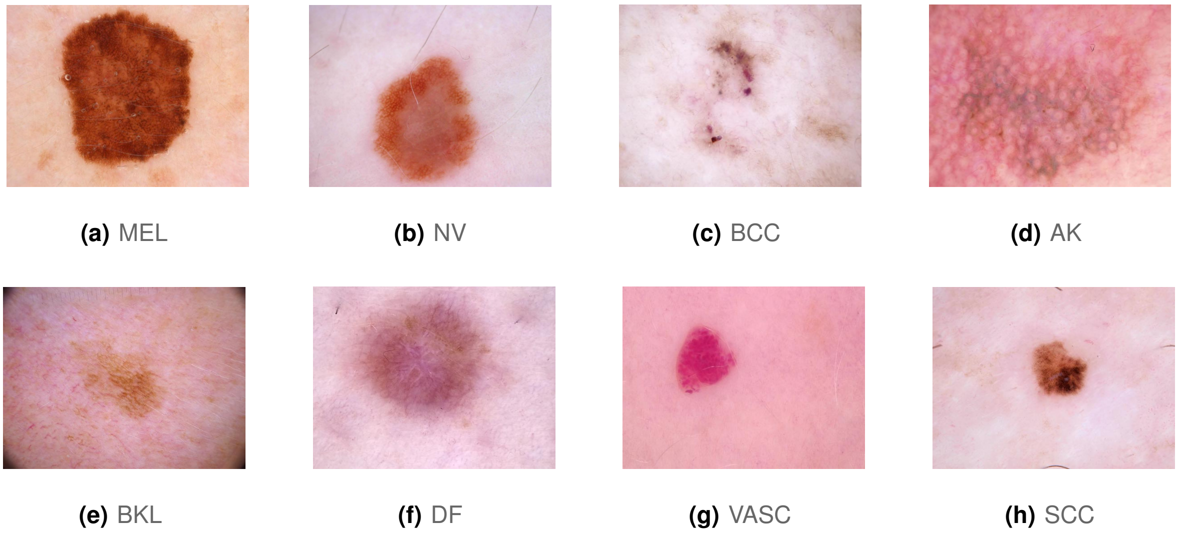
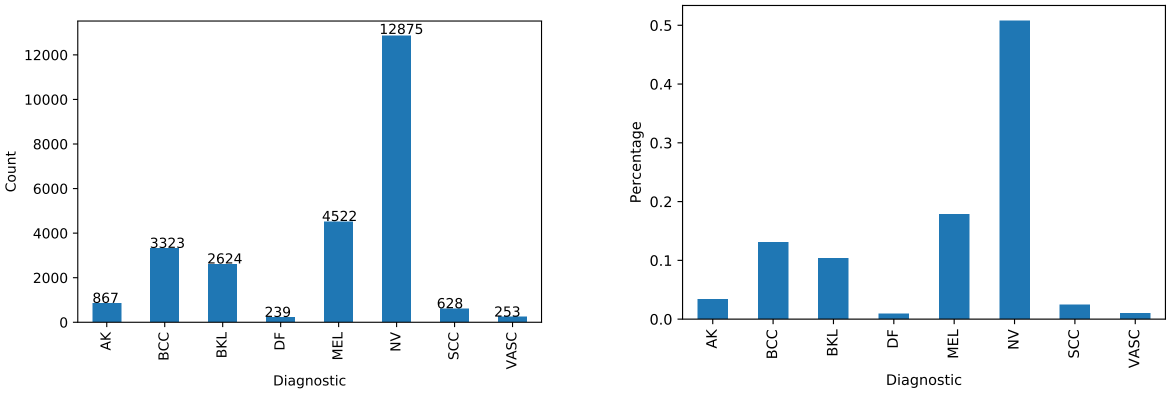

(a) MEL **(b)** NV **(c)** BCC **(d)** AK **(e)** BKL **(f)** DF **(g)** VASC **(h)** SCC

Figure 3.1: An example of each type of skin lesion represented in ISIC dataset.

The dataset comprises 25,331 images with ground truth labels for training and a held-out testing set of 8,238 images. The labels of the testing set are not available. It is important to note that the testing dataset also contains an additional outlier class not represented in training data. However, this thesis does not address the problem of outlier detection. As stated in [16], the ISIC 2019 dataset comes from different hospital sources: HAM10000 [37], BCN 20000 [38], and MSK [39]. The size of the images varies, depending on the source of the dataset.

As outlined in fig. 3.2, the training dataset provided is highly class-imbalanced in which more than 50% images belong to NV class. DF and VASC class images contain only a few images.



(a) Diagnostic count

(b) Diagnostic Percentage

Figure 3.2: Frequency of each lesion per diagnostic in the training dataset provided.

To perform the various experiments, it was necessary to divide the original training set into a smaller training set (80%) and a validation set (20%). The models are trained with the training set. The validation set is used to choose and adjust the hyperparameters and the best architectures that will be shown in chapter 4. All the choices aim to improve the results in the validation set. Table 3.1 summarizes the number of images records for the training, validation, and testing sets, split by all the eight different classes. Recall that in the case of the testing set, as the labels are not available, there is no information regarding the number of samples per class.

Table 3.1: The total number of samples in training, validation and testing sets. The number of samples per class in the training and validation set.

Dataset	Total	MEL	NV	BCC	AK	BKL	DF	VASC	SCC
Train	20265	3654	10241	2678	698	2084	195	209	506
Validation	5056	868	2634	645	169	540	44	44	122
Test	8238								

3.2 Metadata

In addition to the images, the dataset also contains metadata for most of the samples. The metadata is composed of the patient's age and gender, and the region of the body where the skin lesion is located. Data analysis with metadata was performed in order to better understand the potential influence of the patient's information on the classification of skin lesions. It is important to refer that the age is represented in intervals of 5 years (the value 0 in age represents an interval between 0 and 5 years old). There are 18 different age intervals, between 0 and 90 years old, and there are 8 different regions of the body in total.

In fig. 3.3, the age distribution for each lesion is represented, where the black horizontal line corresponds to the median age for each diagnostic. The blue box shows the quartiles of the dataset, for each lesion, while the whiskers extend to show the rest of the distribution, except for points that are determined to be outliers. The black points indicate the records that are much less frequent than the others, for each lesion. Regarding the patient's age box plot, it can be noticed that the median age for NV is lower than all the others. There seems to be no records of patients under the age of 35 years old (and from 35 to 45 years old few records exist) for the SCC lesion. There are also a few cases of patients with AK under the age of 40 years old. Based on these observations, age may be helpful to differentiate some classes.

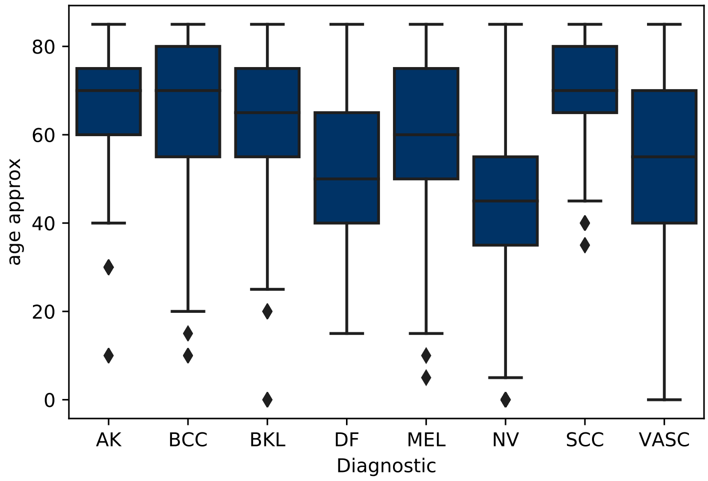


Figure 3.3: Boxplot of the age distribution per diagnostic.

Figure 3.4 represents the number of samples by gender, for each lesion. As can be seen, the frequency for both genders is almost the same for all lesions. Nonetheless, there are some discrepancies, for instance in BCC, BKL, MEL, and in SCC.

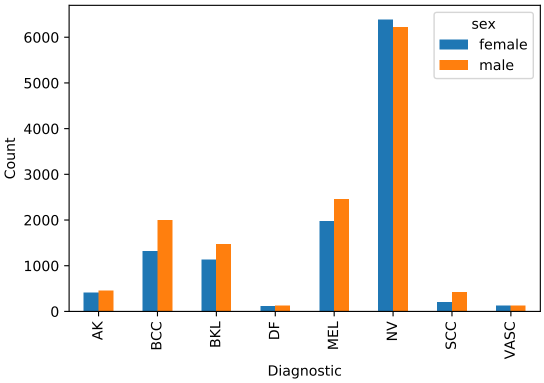
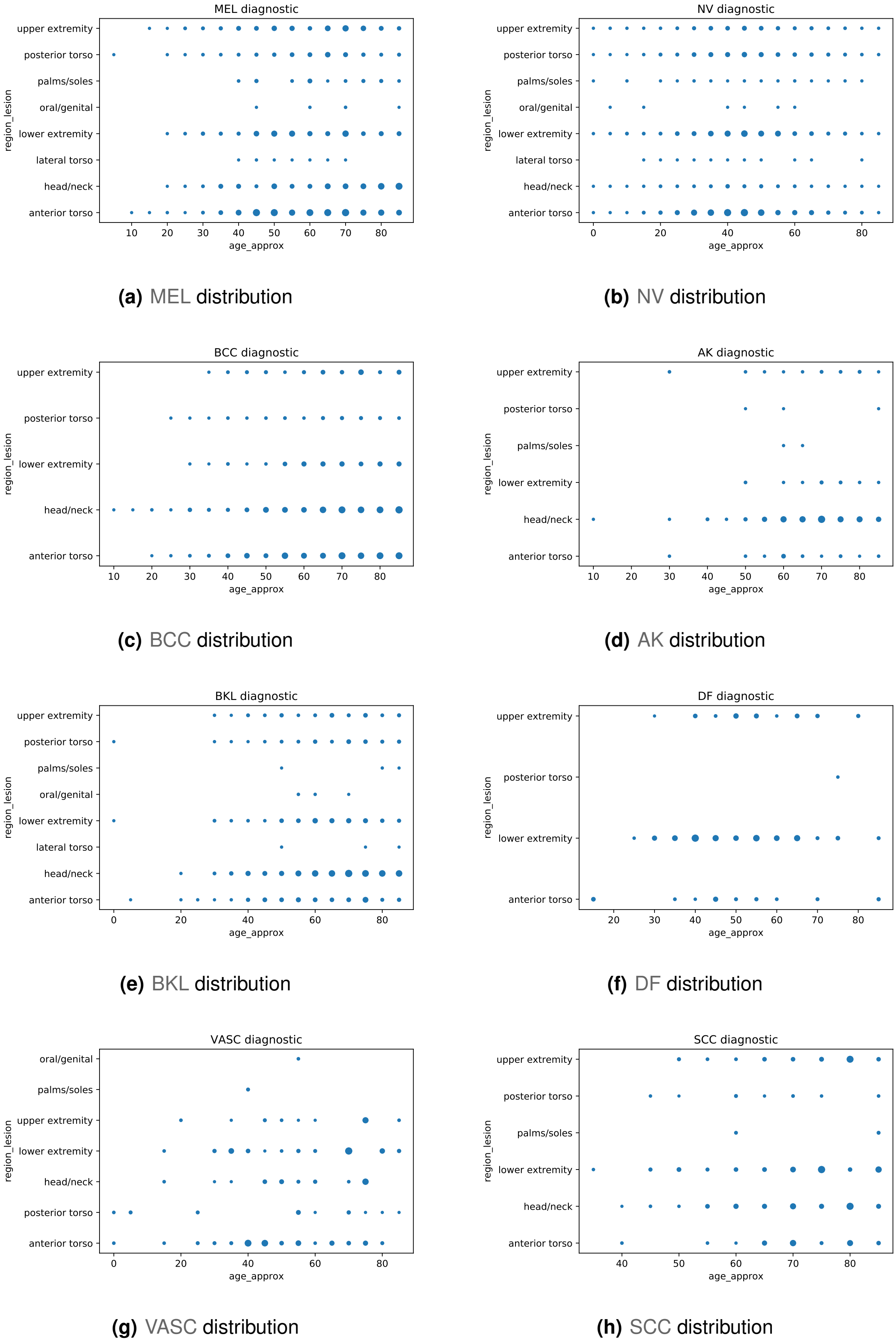


Figure 3.4: Distribution of each skin lesion by gender in dataset.

Finally, the relationship between each lesion and the region of the body is analyzed. In this case, the patient's age and the patient's gender is taken into account. In fig. 3.5 the size of the markers represent the two-dimensional distribution of occurrences according to the patient's age and the body region of the lesion, for each type of lesion. It is important to note that each point represents an age range. For example, a point in age equal to 55 years old represents an age between 55 and 60 years old. It is possible to draw some conclusions from fig. 3.5. Some lesions appear more frequently in specific body regions. For instance, MEL and NV share similar regions (more frequent in anterior torso), BKL and AK are more frequent in head/neck and DF in lower extremity. DF appears only in 4 different parts of the body. Thus, the anatomic site may be helpful to distinguish some lesions. Taking into account not only the region of the lesion but also the patient's age, it may be observed that the preferences for some regions of the body can be restricted to some age ranges. For example, in NV the most frequent region is the anterior torso between 30 and 55 years old, and the lower extremity between around 35 and 60 years old. Regarding BKL and AK, the prevalence to head/neck is higher for ages over around 55 years old. DF is more frequent in the lower extremity between 40 and 70 years old. In the posterior torso, DF just has records with 75-80 years old. VASC contains some points that stand out from the others, for instance lower extremity/70-75 years old, and two points in the anterior torso: 40-45 years old and 45-50 years old.

Following the same reasoning, in fig. 3.6, the bi-dimensional distribution per diagnostic is represented, taking into account the combination of gender and region of the body. Note, in general, there are no significant differences between the genders. However, it can be observed that VASC is more frequent in female's anterior torso. It seems to be found in palms/soles just in males, and in oral/genital only in females. Regarding AK and DF, there are cases in the posterior torso only in males (and not in females). SCC seems to be more frequent in male's head/neck, male's anterior torso, and in the lower extremity (in males and females).

Lastly, in fig. 3.7, the bi-dimensional distribution by diagnostic is depicted, with the variables age and gender. In this case, it is visible that MEL is more frequent in males between 65 and 75 years old, NV is more frequent between 35 and 55 years old, BCC and SCC are more frequent in males over 70 years old and BKL in males over 65 years old. VASC is found with more frequency in males with 75-80 years old.



(a) MEL distribution

(b) NV distribution

(c) BCC distribution

(d) AK distribution

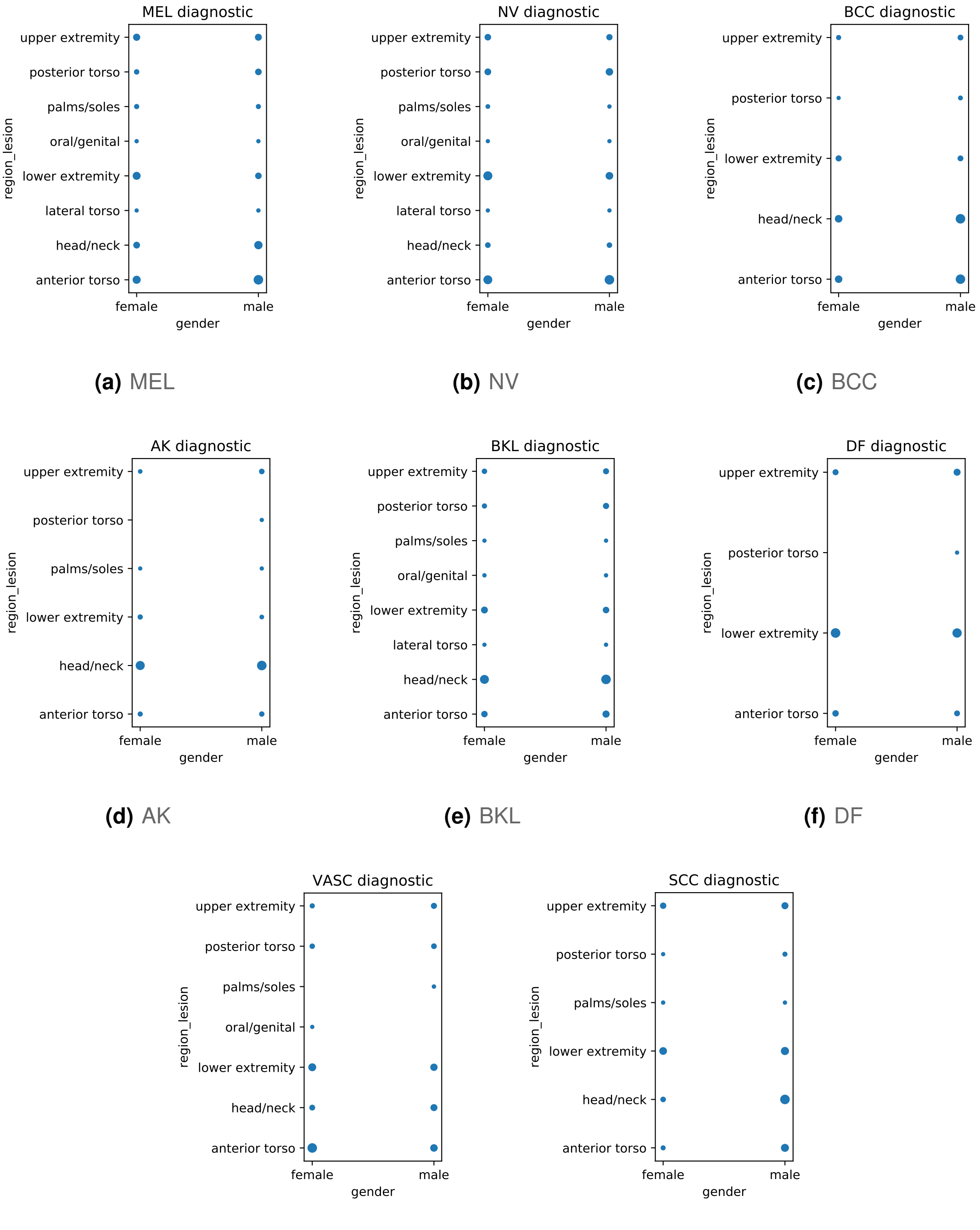
(e) BKL distribution

(f) DF distribution

(g) VASC distribution

(h) SCC distribution

Figure 3.5: Bi-dimensional distribution per diagnostic, with variables age and the body region of the lesion. The mark's area represents the probability, where the sum of the area of the all marks in each lesion is equal to 1.



(a) MEL

(b) NV

(c) BCC

(d) AK

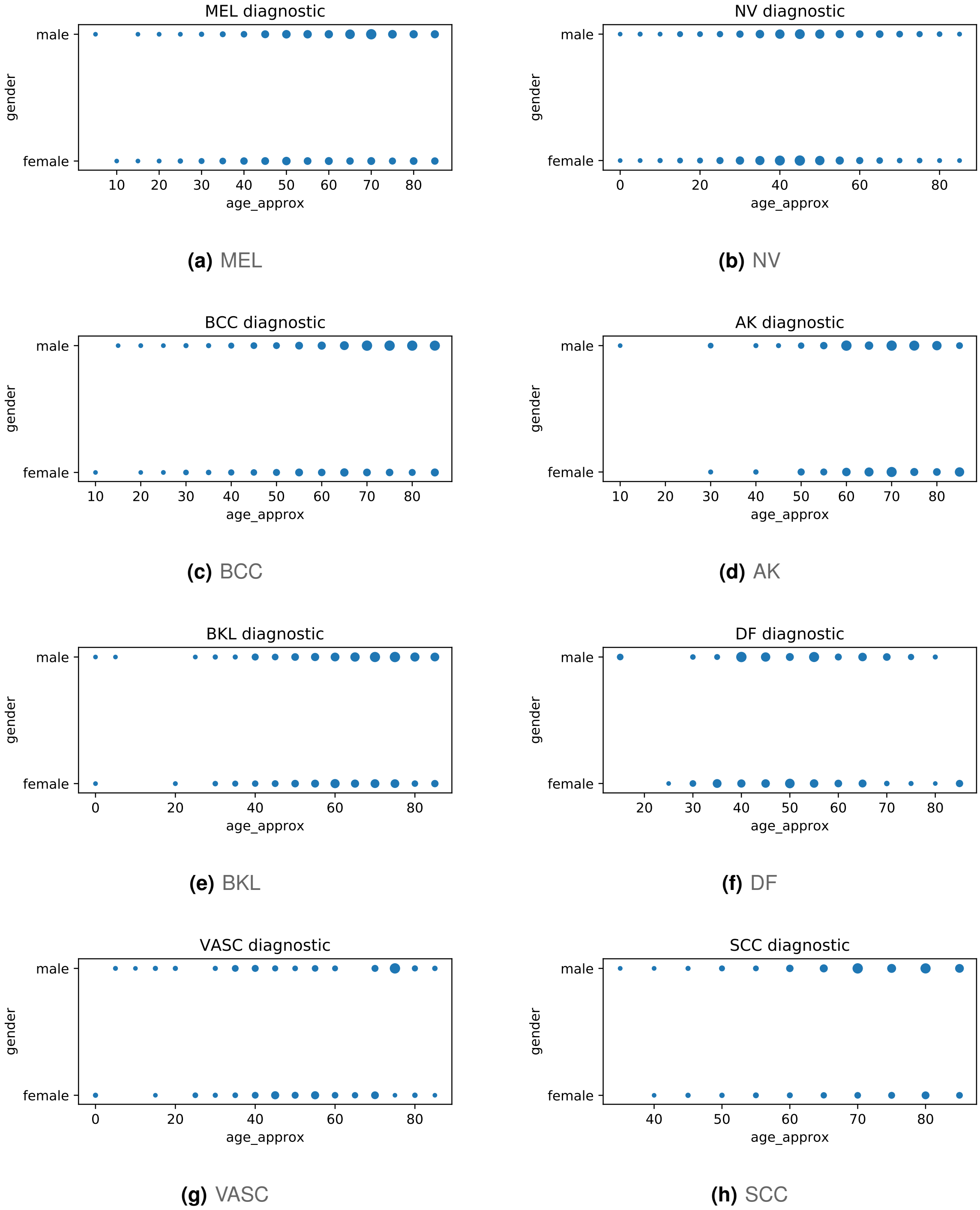
(e) BKL

(f) DF

(g) VASC

(h) SCC

Figure 3.6: Bi-dimensional distribution per diagnostic, with variables gender and the body region of the lesion. The mark's area represents the probability, where the sum of the area of the all marks in each lesion is equal to 1.



(a) MEL

(b) NV

(c) BCC

(d) AK

(e) BKL

(f) DF

(g) VASC

(h) SCC

Figure 3.7: Bi-dimensional distribution per diagnostic, with variables age and gender. The mark's area represents the probability, where the sum of the area of the all marks in each lesion is equal to 1.

4

Proposed System

Contents

This chapter introduces the proposed system that aims to answer the question of this thesis: Does combining patient information with dermoscopic images for skin lesion diagnosis can lead to further improvements over just dermoscopic images?

4.1 Overview of the Proposed System

The main purpose of this thesis is to understand if the patient's clinical information is useful for diagnosing skin lesions. To address this question, different diagnostic systems were designed and evaluated: systems based only on dermoscopic images, systems with metadata only, and systems with both. The main steps of the systems that combine images and patient information are illustrated in fig. 4.1. It is important to refer that the Pre-processing block is common to all methods.

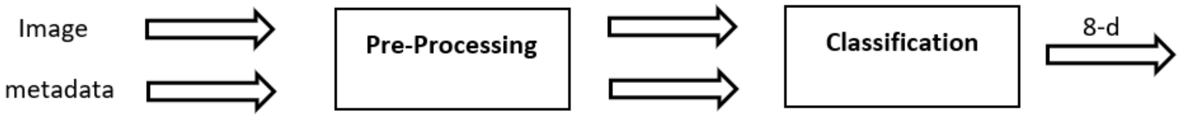


Figure 4.1: The main blocks of the proposed system.

Before being fed in the different models to perform the classification, the image and the metadata are pre-processed. The final output of the system is an 8-d vector because there are 8 different lesion classes. The output represents a probability vector of the different classes. The classification is performed according to the highest probability of the vector. In this chapter, the blocks shown in fig. 4.1 are described in detail. Two different CNNs architectures were used to process the dermoscopic images, and five different methods were investigated to process the metadata and combine this information with the one from the images.

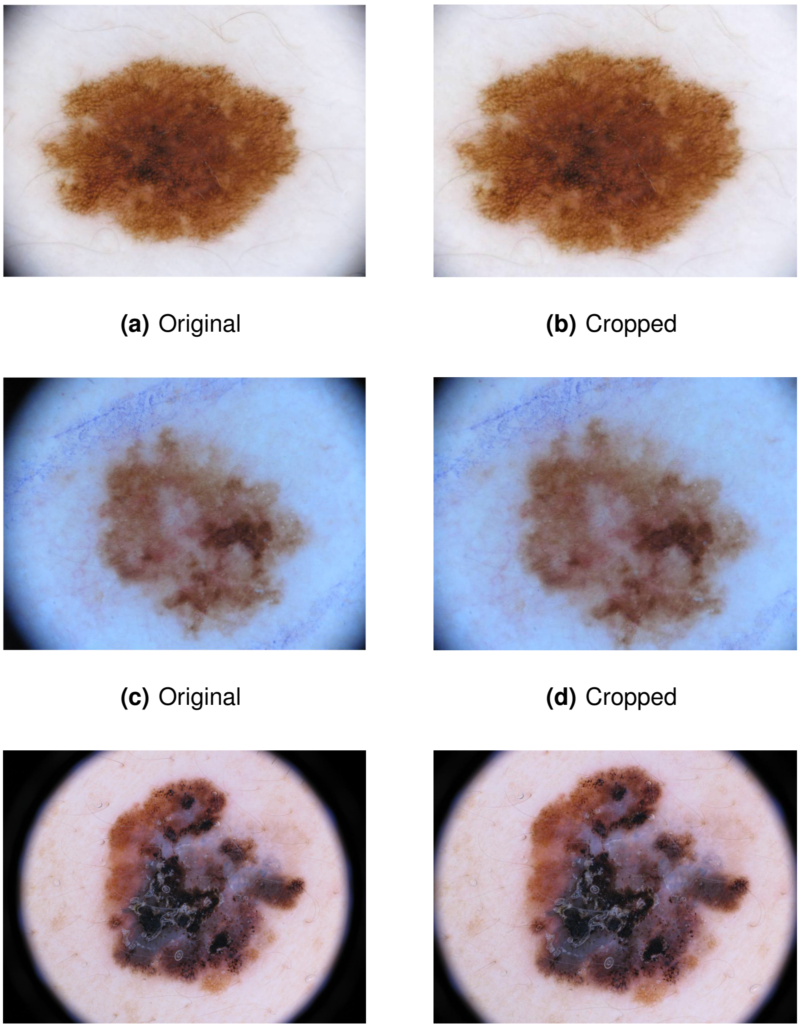
4.2 Pre-processing

Before feeding the images and the metadata into the models, data pre-processing is required. As mentioned in chapter 3, ISIC dataset comes from different medical centers: HAM10000 [37], BCN 20000 [38], and MSK [39], and was acquired using different equipments. For this reason, the size, the color and the aspect ratio of the images are different. To overcome these differences, pre-processing operations that compensate the color and normalize the dimensions were performed.

As far as metadata is concerned, since the metadata contains categorical features, one-hot encoding technique was applied.

4.2.1 Image pre-processing

Data variability was addressed by applying cropping and a color constancy algorithm. As a first step, a central cropping strategy is used, since some of the images often show a black area in the borders. This strategy aims to reduce this black area or eliminate it. Some examples of this technique are shown in fig. 4.2.



(a) Original

(b) Cropped

(c) Original

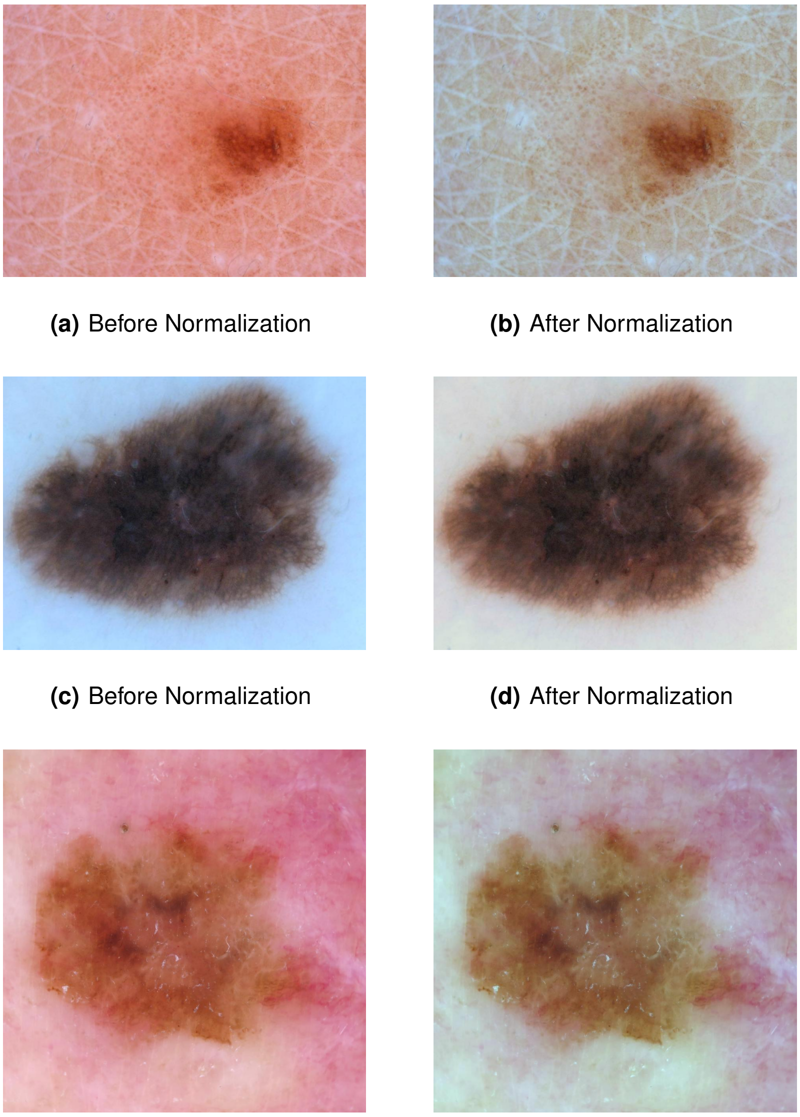
(d) Cropped

(e) Original

(f) Cropped

Figure 4.2: Examples of crop transformation in dermoscopic images.

If a system operates with multisource images (with different acquisition devices and illumination conditions), there may be significant changes in the colors of the acquired images, leading to alterations in the values of the color features in CAD systems. This may reduce the performance of the systems [40]. In [40], it is shown that it is important to normalize the colors of dermoscopic images (before training and testing) with color constancy algorithms. Color constancy is meant to transform the colors of an image, acquired using an unknown light source, to identical colors under a canonical light source. In this work, the color constancy algorithm Shades of Gray with Minkowski norm p = 6 is used, as proposed in [40]. This method automatically estimates the color of the illuminant, since part of the reason why images look so different is the color of the light source. After estimating the color of the illuminant, the image is transformed, based on this value, to the canonical light source. The same technique with the same value p was used in [24]. In fig. 4.3, the original and resulting images, after the color normalization, are depicted.



(a) Before Normalization

(b) After Normalization

(c) Before Normalization

(d) After Normalization

(e) Before Normalization

(f) After Normalization

Figure 4.3: Examples of color normalization with color constancy algorithm - Shade of Gray.

As it can be seen, the resulting images are similar in terms of color. After applying these techniques, since the images have different sizes depending on the source, all the images are resized to the size 224×224 or 229×229, depending on the CNN used to process them.

4.2.2 Metadata pre-processing

The metadata for each image consists of age, gender, and anatomical site. These data are encoded as a feature vector, using a one-hot encoding strategy. The gender is represented by two binary features, where one of them is zero and the other is one, the anatomical site by 8 features, and the age by 18 features (one for each age interval, since the age is represented in intervals of 5 years). For each type of information, just one feature will be 1, and all the others will be 0. Thus, by concatenating all features, the final feature vector has a size 28. In some of the examples, one or more type of metadata may be missing. As such, all of the features associated with that data will be zero. For instance, if the gender is missing, $Fem.$ = 0 and $Male$ = 0 in the one-hot encoding vector. Table 4.1 shows an example, where the patient is a male, the anatomical site is in the lower extremity, and the patient is between 85-90 years old.

Table 4.1: An example of metadata in format one-hot encoding, where the patient is a male between 85-90 years old, and the anatomic site is in the lower extremity.

Fem.	Male	Anterior	Head/	Lateral	Lower	Oral/	Palms/	Posterior	Upper	Age$_0$	Age$_5$	...	Age$_{85}$
		torso	neck	torso	extremity	genital	soles	torso	extremity	0-5	5-10		85-90
0	1	0	0	0	1	0	0	0	0	0	0	0..0	1

4.3 Skin lesion classification

This thesis considers three types of models: a CNN for the diagnosis of dermoscopic images, a multi-layer perceptron for diagnosis based on metadata only, and a deep learning model that integrates both images and metadata. In this section, all the different methods are described.

4.3.1 Classification using only dermoscopic Images

The first diagnostic model proposed in this thesis uses only dermoscopic images. The diagnostic is performed using a CNN. The image is first pre-processed, as previously described, and then fed into the CNN Model block, which comprises convolutional and pooling layers, and a global average pooling layer block. The Convolutional and Pooling Layers block, outlined in fig. 4.4, is a stack of convolutional and pooling layers. A global average pooling layer is applied to the output of this block, to obtain a vector of size 2048 (because it is the number of filters of the last convolutional layer), that will be fed into a

FCL with 8 neurons, which performs the decision. This overall scheme is outlined in fig. 4.4, where it is assumed that the image is already pre-processed.

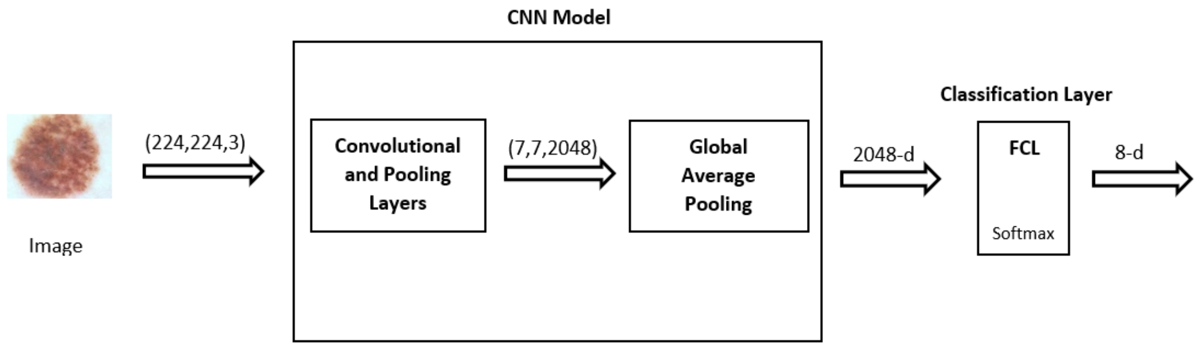


Figure 4.4: The model used to classification with dermoscopic images, where a CNN Model block and a Classification Layer modules are defined, to be used in the next examples. The Convolutional and Pooling Layers block depend on the model used.

The configuration of the Convolutional and Pooling Layers block depend on the architecture used. Two different networks from section 2.3 were chosen to be used as CNN Model block, since they have outperformed in a large scale image classification task (ImageNet Challenge). The best three top-5 error results presented in section 2.3, among the studied architectures, are ResNet, Inception-ResNet and Xception. ResNet-101 [6] and Xception [9] were chosen . Xception is based on Inception module. Since Inception-ResNet combines Inception architecture with residual connections, it was excluded from the experiments.

In ResNet, the input image has size 224×224, while for Xception the size is 229×229. In both cases, the network ends with a global average pooling layer (the input of this layer is a feature-map with size (7,7,2048), and the output is a vector of size 2048, where each position of the vector represents the average of each 7x7 channel). In the next subsections, CNN Model block will be shown in the block diagrams. After this block, there is an 8-way fully-connected layer (because there are 8 classes/lesions) with $Softmax$ as the activation function, that performs the classification. This is called as Classification Layer block and will appear in the next subsections with that name.

4.3.2 Classification with metadata

Although metadata contains little information about skin lesion classes, classifiers using only metadata inputs were designed. These models are composed of a stack of FCLs (multi-layer perceptron), varying among them the number of hidden FCL, the number of neurons in each FCL, and the initial learning rate. The best configuration consists of a single FCL with 500 neurons, followed by a $Softmax$ with 8 neurons. The network input is a vector of size 28 in a one-hot encoding format as described in section 4.2.2.

4.3.3 Classification with both dermoscopic images and metadata

In order to classify lesions using images and metadata, different approaches were compared. In these approaches, there was a fusion between the image network and the metadata network. This fusion can be classified into two main classes depending on how the information from each network is combined. These two main classes are called late fusion and early fusion [41]. In late fusion, the fusion is done at the decision level. Thus, it consists of a combination of the results obtained by different classifiers. Early fusion takes a different approach as the fusion is done at the feature level. Classification is then performed using the combined representation [41].

Five strategies were investigated to combine images and metadata. For each of them, several experiments with different architectures were carried out, and the best five, according to the validation set, are presented.

Method 1

The first method comprises the CNN Model block (the same as the depicted in fig. 4.4), where the input is a dermoscopic image, and a metadata network, with just a FCL, where the input is the metadata. The fusion is carried out by concatenating the output of both networks. This model is illustrated in fig. 4.5.

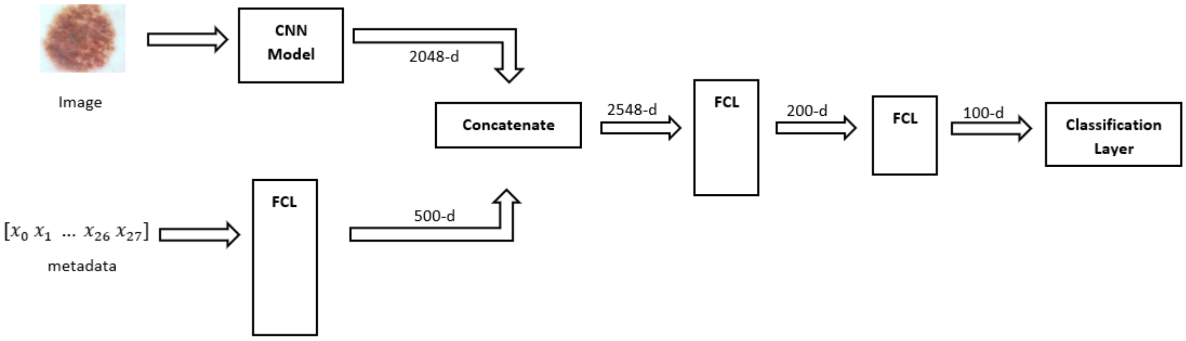


Figure 4.5: Method 1: A fusion of metadata and the CNN image model. Fusing architectures by concatenating outputs at feature level - early fusion.

As can be seen, the output of the CNN Model is the output of the global average pooling: a feature vector of size 2048-d. In relation to the metadata network, the 28-d feature vector is applied to a network with only one layer with 500 neurons, with ReLU activation function. The output of this network is a vector of size 500-d. These two outputs are concatenated, and the output of this operation is a feature vector with dimension 2548. The same way of fusing the networks was performed in [42] and in [24]. This fusion is classified as early fusion, since the fusion is done at the feature level. The concatenation output is followed by two FCL (the first with 200 neurons, and the second with 100 neurons). The network ends with a Classification Layer (same as described in fig. 4.4).

During the training phase, the initialization of the weights in the CNN Model block is not random. A pre-trained model is used to initialize it: the weights obtained from training a CNN for diagnosing dermoscopy images as described in section 4.3. Firstly, tests with only dermoscopic images were performed, and the weights that lead to the best result were saved and used as initial weights here. The metadata network weights and the weights of the remaining FCLs are randomly initialized. During the train, all weights are updated.

Method 2

The second method adopts a different way of fusing the information and was inspired on [15]. The architecture and training of the model are similar to method 1. However, the differences are: this approach does not perform concatenation between the output of the CNN Model block and metadata network. Instead, it multiplies the outputs. For accomplishing it, the dimension of each network output must be equal, since each feature-map of the CNN Model output is multiplied by the corresponding vector element from the metadata network. This is also a type of early fusion. This method is depicted in fig. 4.6.

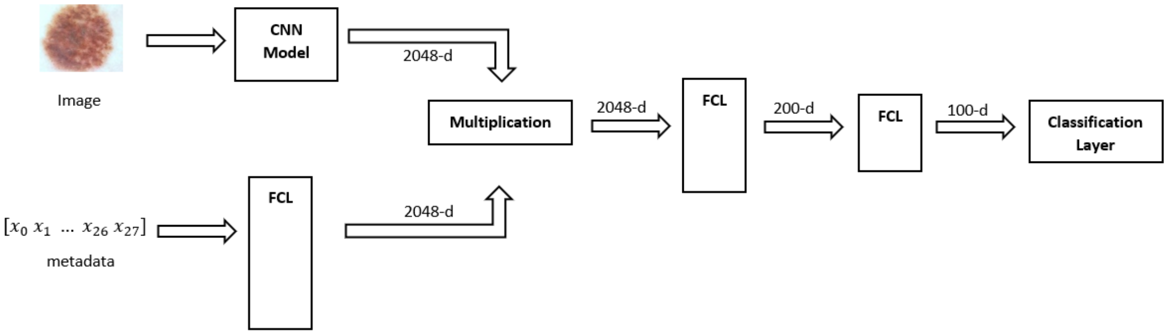


Figure 4.6: Method 2: A fusion of metadata and the CNN image model. Fusing architectures by multiplying the outputs at feature level - early fusion.

With this approach, the metadata controls each feature channel of the CNN Model (for instance, the metadata network can learn which feature-maps are more relevant and give more importance to those feature-maps by assigning higher values in the respective positions, and can disable a specific feature-map by introducing a value 0 in the respective position). As such, the metadata network is composed of a layer with 2048 neurons (instead of 500 neurons) with ReLU activation function. The output of the multiplication layer has size 2048-d. After this layer, everything is the same as method 1: the output is applied to a stack of two FCLs with ReLU activation function and a Classification Layer. As in method 1, the initial weights of the CNN Model block are the values obtained for the CNN trained for image classification, using only dermoscopic images. Then we allow for all the weights to be updated during training.

Method 3

The third method is similar to method 2. The difference is in the way of combining the image and metadata information. This method does not perform multiplication between the feature-map (2048-d) of the CNN Model and the output of the metadata network (also 2048-d). Instead, it performs an average of both outputs. Once again, it is an early fusion, and the dimension of each network output must be equal. Each feature-map of the CNN Model is averaged with the corresponding vector element from the output of the metadata network. This is the only difference between the two methods. The described architecture is exemplified in fig. 4.7.

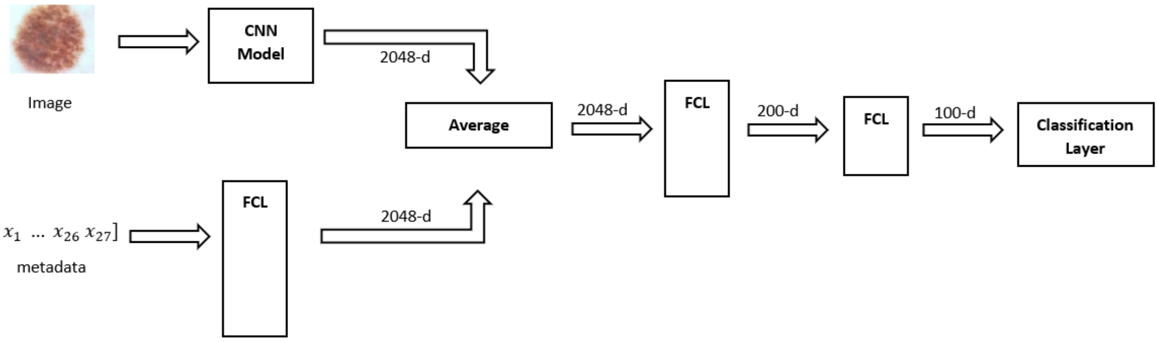


Figure 4.7: Method 3: A fusion of metadata and the CNN image model. Fusing architectures by averaged the outputs at feature level - early fusion.

Method 4

In method 4, the module responsible for combining the outputs performs a squared sum. To accomplish it, the size of the output of both networks is the same. As such, the FCL used in the metadata network contains 2048 neurons. After applying the fusion operation, the output of the fusing layer (with size 2048-d) is fed to a stack of one FCL, with 200 neurons and a ReLU activation function, and a Classification Layer. The scheme of this method is represented in fig. 4.8.

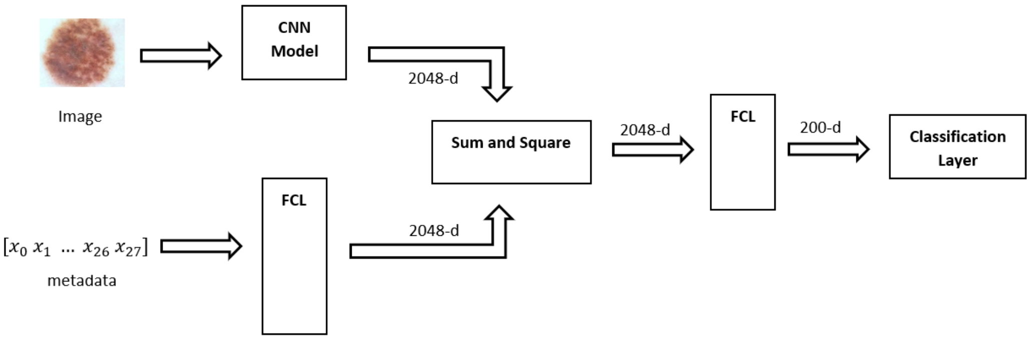


Figure 4.8: Method 4: A fusion of metadata and the CNN image model. Fusing architectures by sum and square the outputs at feature level - early fusion.

Method 5

In method 5, the fusion is done at the decision level, by combining the classifiers of both networks. The output of the Classification Layer (with $Softmax$ activation function, as depicted in fig. 4.4) of the image network has size 8-d, and it is multiplied by $1 - \alpha$, while the output of the Classification Layer of the metadata network (also with $Softmax$ and size 8-d) is multiplied by α. Then, these two outputs are summed, position by position, resulting in an 8-d output vector, where the sum of the output vector is equal to 1 and, therefore, it can be in interpreted as a probability distribution. The method is represented in fig. 4.9.

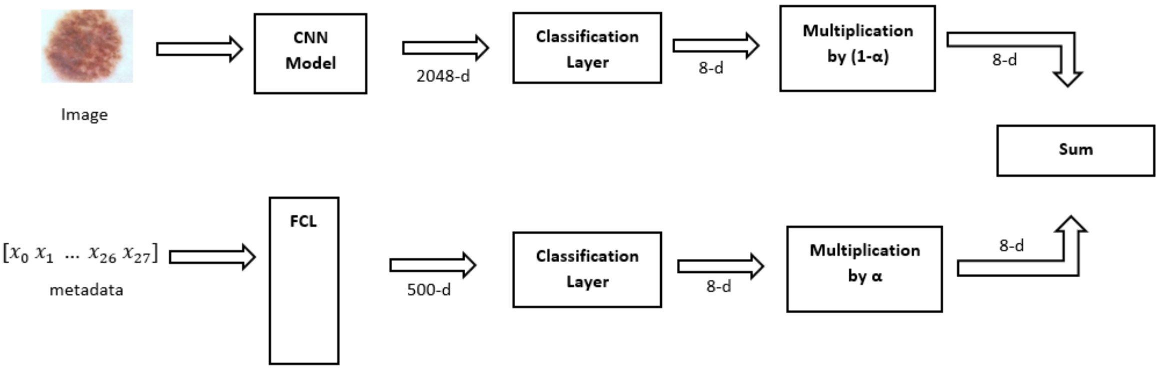


Figure 4.9: Method 5: A fusion of metadata and the CNN image network by combining the classifiers. Nevertheless, all the model is re-trained.

Since the information is combined at the decision level, this approach produces better results when everything is trained end-to-end, instead of just combining the classifiers without training the weights. Thus, it was considered as a late fusion with training. The weights of the image network (CNN Model + Classification Layer) were initialized with the weights obtained by the trained CNN only for image classification, but those weights were allowed to change during the train.

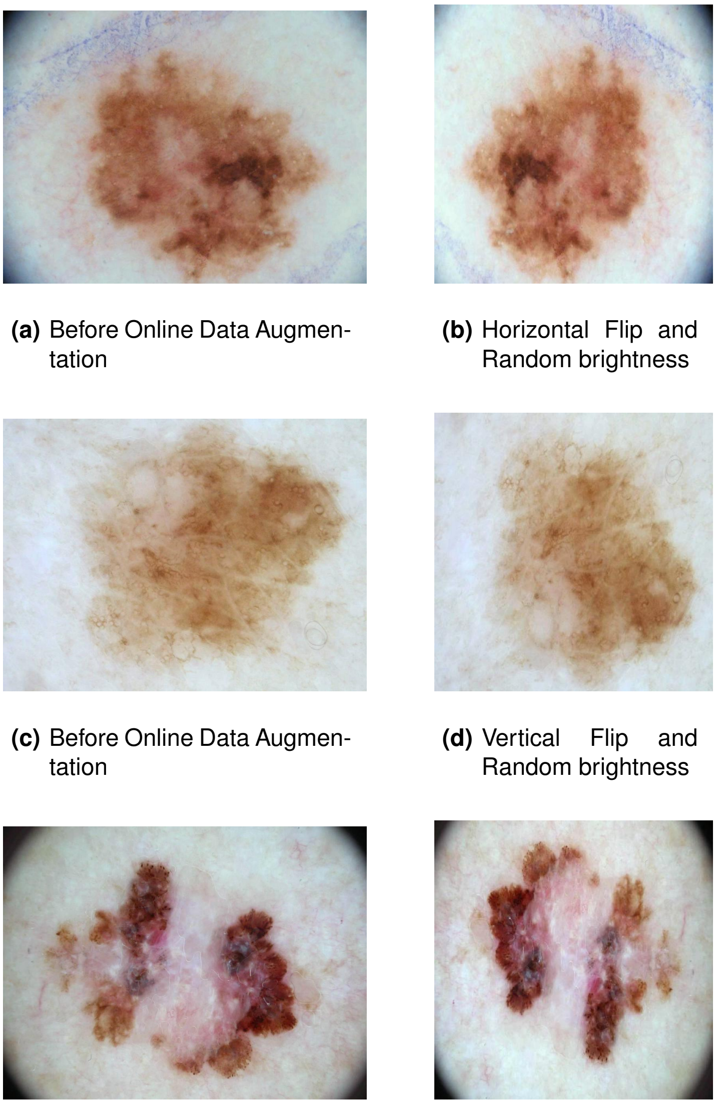
4.4 Training issues

Since the trained models have a large number of parameters, there may be an overfitting problem. Moreover, as discussed in chapter 3, the dataset used in this thesis is highly class-imbalanced, where some lesions classes contain just a few images. In order to overcome these issues, the following strategies were adopted during the training phase.

4.4.1 Data augmentation

Image data augmentation

The number of examples provided in the dataset is limited. Moreover, several images present different orientations, locations, scales, brightness, etc. To help to reduce the overfitting, the network can be trained with additional synthetically modified data. In order to get more variability of data, some geometrical transformations are applied to the training set. In particular, the following forms of data augmentation are performed: randomly flip images horizontally and vertically, and random brightness. Different versions of each original image are fed into the network. In this work, the data augmentation performed is an online data augmentation: whenever an image is used to train or test the network, it is resized and, then, randomly flipped horizontally and vertically are applied, independently, to the original image with probability p = 0.5 (each transformation is applied with probability p). Then, random brightness is applied independently of the other's transformation. For example, the resulting image may be flipped horizontally and vertically, just one of them, or none, and, in addiction, random brightness is applied. Some examples are shown in fig. 4.10. Images on the right (after augmentation) were resized to 224×224, while images on the left have the initial size. In both cases, the images were, previously, cropped and color normalized.



(a) Before Online Data Augmentation

(b) Horizontal Flip and Random brightness

(c) Before Online Data Augmentation

(d) Vertical Flip and Random brightness

(e) Before Online Data Augmentation

(f) Horizontal Flip, Vertical Flip and Random brightness

Figure 4.10: Data Augmentation: random horizontal and vertical flip, and random brightness. All the images were cropped and normalized before. The right images are resized to 224×224.

Metadata augmentation

In all the methods described above that combine images and metadata, the same approach to data augmentation is carried out. This was necessary since not all images contain metadata. There are cases where just one or two types of information are missing (age and gender are missing at the same time, for example) and others where all information is missing. As referred in section 4.2.2, if a certain piece of metadata is missing, all features of that type will be zero. For example, if gender is absent, in the one-hot encoding vector, the features $Fem.$ and $Male$ represented in table 4.1 will be zero. To handle the missing data, augmentation is performed in the metadata. During the train, the model independently encodes each type of metadata as missing with a probability of p = 0.1.

For instance, if for a given patient the gender is provided and he is a male, the gender input will be $Fem.$ = 0 and $Male$ = 1, in the one-hot encoding vector. Nevertheless, since the system randomly encodes each type of metadata as missing with probability p, it can encode the gender feature as a missing value, and, in this case, the one-hot encoding input vector (depicted in table 4.1) will have the first two entries ($Fem.$ and $Male$) equal to zero. If a sample has no missing information, the probability of passing the real one-hot vector to the network is equal to 0.9^3 [$(1-p)\cdot(1-p)\cdot(1-p)$ = $(1-0.1)\cdot(1-0.1)\cdot(1-0.1)$], since there are 3 types of information, and the probability of each being considered as a miss is independent of the others. This way, the system is prepared to handle with lack of metadata. In this case, the system is expected to learn to adapt to the missing values by solely relying on the dermoscopy image.

4.4.2 Class weights in Loss Function

In order to address the class imbalance problem, class weights are applied to the loss function. These weights are used in all of the experiments. The weights in the loss function are inversely proportional to the class frequencies in the training data:

$$class\_weight_i = \frac{N_{total}}{N_i}, \tag{4.1}$$

where $class\_weight_i$ is the class weight for class i, N_{total} is the number of samples in the training set, and N_i is the number of samples of class i in the training set. Thus, there are different cost weights in the loss function, according to the number of samples of the class, where the less frequent classes have a higher weight in relation to the others. As such, it is possible to place more emphasis on the minority classes such that the final model goal is a classifier that can learn equally from all classes.

4.4.3 Dropout

In order to handle overfitting problems, Dropout is applied to all FCL - (before $Softmax$ layer for example), once it is in these layers that exist more weights. In this technique, each neuron is activated with a fixed probability. In other words, it consists of setting to zero a subset of hidden neuron randomly chosen with probability p. The value of p used in this work is 0.5. Activation dropout works really well for regularization purposes [30].

4.4.4 Transfer Learning

For all CNN architectures pre-trained models are used. It consists of taking features learned on a problem and leveraging them on a new problem [43]. In other words, the initial weights used in our CNN model were obtained from models trained for the classification of the ImageNet dataset. For instance, when ResNet is used to this purpose, the weights obtained with ResNet architecture on ImageNet are used for initialization. After loading the pre-trained weights to the model, fine-tuning is performed, which consists of re-training the model on the new data (the pre-trained features will adapt to the new data). Some experiments without transfer learning were performed, and it was concluded that this method leads to faster convergence and better results. Therefore, this technique was adopted in all the experiments that only used image data. When both dermoscopic images and metadata are combined, the CNN weights are initialized with the values obtained for the CNN fine-tuned to classify a dermoscopic image.

5

Experiments and results

Contents

This chapter starts by re-introducing the dataset, and then it describes the metrics used to evaluate all the experiments. Afterwards, it presents the experimental results and a discussion of the methods proposed in chapter 4 to classify the skin lesion with and without metadata.

5.1 Dataset

As stated in chapter 3, the dataset comprises 25,331 images with ground truth labels for training and a held-out test set of 8,238 images. The labels of the test set are not available. As mentioned in [16], the ISIC 2019 dataset comes from different hospital sources: HAM10000 [37], BCN 20000 [38], and MSK [39].

The original training dataset is divided into the training set (80%) and the validation set (20%). Table 5.1 summarizes the number of images and metadata records for each of the training, validation, and test sets, split by all the eight different classes (in the case of the test set, as the labels are not available, there is no information regarding the number of samples per class).

Table 5.1: The total number of samples in training, validation and test sets. The number of samples per class in the training and validation set.

Dataset	Total	MEL	NV	BCC	AK	BKL	DF	VASC	SCC
Train	20265	3654	10241	2678	698	2084	195	209	506
Validation	5056	868	2634	645	169	540	44	44	122
Test	8238								

In addition to the images, the dataset also contains metadata for most of the examples. The metadata is composed of the patient's age and gender, and the body region where the skin lesion is located.

5.2 Evaluation Metrics

5.2.1 Confusion Matrix evaluation

In order to compute the performance metrics, the confusion matrix must be defined. The confusion matrix is a matrix with dimension $k \times k$, where k is the number of classes. This matrix can be presented in a normalized version, such that the element i, j of the matrix represents:

$$P_{i,j} = Prob(Predict\_Class = j | True\_Class = i). \tag{5.1}$$

In other words, each matrix entrance, i, j, corresponds to the probability of predicting class j when the real class is the class i.

5.2.2 Binary problems

In binary problems (only 2 classes), the confusion matrix has size 2 $\times$ 2. Therefore, it is possible to define the concepts of: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which are the different scores that can be extracted from the confusion matrix. If class 1 is considered a positive class, the different concepts will be defined as:

- TP: It is predicted positive (class 1), and it is true.
- FP: It is predicted positive (class 1), and it is false (it belongs to class 0).
- TN: It is predicted negative (class 0), and it is true (it belongs to class 0).
- FN: It is predicted negative (class 0), and it is false (it belongs to class 1).

5.2.3 Multi-class problems

The concepts introduced above can be extended to the case of multi-class (with k classes). In this setup, to compute each of these concepts for each class, it is necessary to define a one-vs-all strategy, for each class. As an example, let's consider that there are k = 3 classes (0,1 and 2) and we are evaluating these parameters with class 0 as a positive class. The one-vs-all strategy consists of assuming class 0 vs classes 1 and 2. Then, it is possible to extract the following concepts:

- TP: It is predicted positive (class 0), and it is true.
- FP: It is predicted positive (class 0), and it is false/rest (it belongs to class 1 or class 2).
- TN: It is predicted negative/rest (class 1 or 2), and it is true (it belongs to class 1 or 2). In this case, other classification errors (e.g., predicting class 1 when it was 2) are not considered, since we are analyzing it from the point of view of class 0.
- FN: It is predicted negative/rest (class 1 or 2), and it is false (it belongs to class 0).

These parameters are taken from the confusion matrix as represented in fig. 5.1, from the point of view of class 0. Each component represents the sum of the cells of the same corresponding color.

This method is repeated through the 3 classes, where each class has its scores. In the case of this thesis, there are 8 classes. Thus, the confusion matrix is 8$\times$8. The metrics that will be defined next will be computed for the 8 classes.

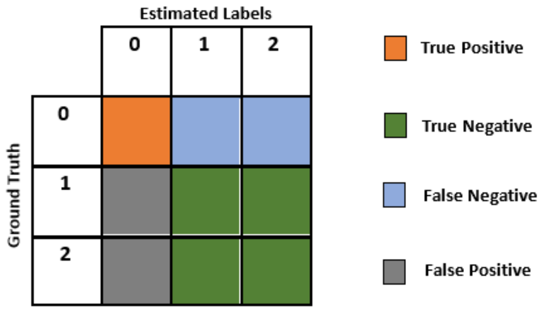


Figure 5.1: Example of multi-classes confusion Matrix for 3 classes, where the positive class is the class 0. Each component corresponds to the sum of the cells of the corresponding color.

Sensibility (SE) and Specificity (SP)

SE and SP are computed for each class. SE is the true positive rate and it corresponds to the percentage of positive samples correctly classified.

The SE for each class i, SE_i, is given by:

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

SP is the true negative rate. It represents the percentage of negative samples that were correctly classified. The SP for each class i, SP_i, is given by:

$$\mathsf{SP}_i = \frac{\mathsf{TN}_i}{\mathsf{TN}_i + \mathsf{FP}_i}. \tag{5.3}$$

Balanced Accuracy (BACC)

Since the dataset is unbalanced, instead of using the weighted accuracy, we will use the BACC. Thus, the same importance is given to all classes, independently of the number of examples. On the other hand, the computation of the weighted accuracy would favor the classes with more samples. BACC is the average of the SE obtained for each class. In this case, it is given by:

$$\mathsf{BACC} = \frac{\sum_{i=0}^{7} \mathsf{SE}_i}{8}. \tag{5.4}$$

Precision

The $precision$ determines of all the records predicted positive, what fraction are actually positive. It is computed by:

$$precision = \sum_{i=0}^{7} \frac{\mathsf{TP}_i}{\mathsf{TP}_i + \mathsf{FP}_i}. \tag{5.5}$$

5.3 Computational conditions

All the code was implemented in Python. The deep learning architectures were implemented based on the frameworks Tensorflow, and its high-level API for building and training deep learning models: Keras [43]. With these libraries, all models were built and trained. The data analysis and the data manipulation were performed using the library Pandas. The library scikit-learn was also very useful to different kinds of computations, such as the confusion matrix and the metrics. The image pre-processing was carried out by using the library open-cv. All of the experiments were performed on a computer with a processor Intel(R) Core(TM) I7-770 CPU @ 3.60 GB, GPU NVIDIA GeForce GTX 1060 6G, and 16 GB RAM.

5.4 Skin lesion classification

In this section, the results of all experiments carried out, with and without metadata, are presented. All the experiments have in common the following conditions:

- The loss function is the categorical cross-entropy eq. (2.6), with Adam Optimizer algorithm eq. (2.8).
- The batch size is equal to 8 (except for the model that only uses just metadata).
- The training was performed during 40 epochs (except when it is used just metadata).
- Class weights in loss function (described in section 4.4.2) are used.
- Dropout with p = 0.5 in all FCL.

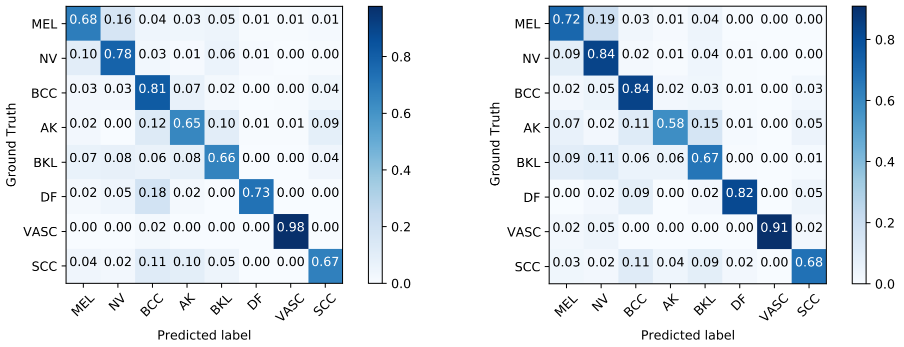
The other hyperparameters were adjusted in order to obtain the best possible value of BACC in the validation set. In all the examples, after training the model, the weights that led to the best value of BACC in the validation set are chosen and loaded to compute the metrics.

The comparison of the results achieved in the validation and the test set to all the methods are presented in section 5.4.4.

5.4.1 Classification with dermoscopic images only

The experiments without metadata were performed using ResNet 101 and Xception architectures (described in section 4.3). In both cases, the initial learning rate is 1^{-5} and it decreases by a factor of 0.75 if the validation loss function does not decrease during 5 consecutive epochs. In the case of ResNet, the best value of BACC was achieved at epoch number 17, while in Xception it was the epoch number 36.

The confusion matrices regarding the validation set for both experiments are depicted in fig. 5.2.



(a) ResNet

(b) Xception

Figure 5.2: Normalized confusion matrices using configurations just with images, in the validation set.

The diagonal of the matrix represents the SE by class, which would ideally be equal to 1, meaning that all samples in that class are correctly classified. The other entries on each line represent the classifier's errors. In this case, regarding ResNet architecture, the most significant confusion occurs in the class AK and BKL. AK obtained a SE equal to 65%, and it was misdiagnosed 12% of the time with BCC, and 10% with BKL. BKL has been confused with AK and NV 8% of the time, each, and 7% with MEL. As far as Xception is concerned, the most significant errors occur in the AK class, which was only well classified in 58% of cases. This class was classified as BKL and BCC in 15% and 11% of the cases, respectively. In both cases, the most accurate class is VASC.

5.4.2 Classification with metadata only

In this case, where just metadata is used as input, the batch size is set to 20 and the initial learning rate is equal to 5^{-5}. The learning rate decreases by a factor of 0.75 if the validation loss function does not decrease during 3 epochs in a row. The training was performed during 50 epochs. The confusion matrix for the validation set is depicted in fig. 5.3.

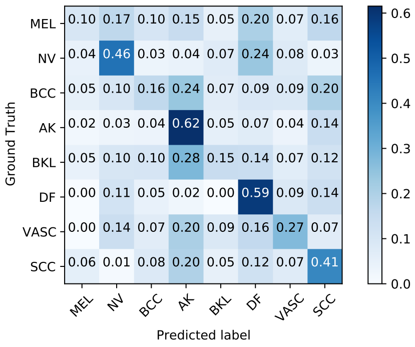


Figure 5.3: Confusion matrix using a network just with metadata, in the validation set.

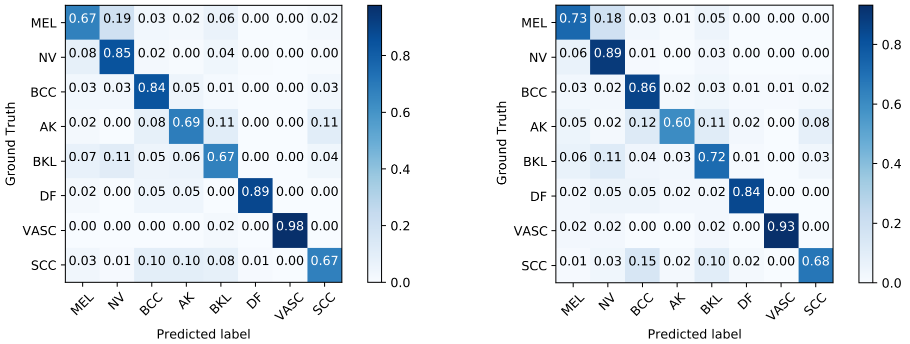
In the validation set, the BACC obtained is 34.41%, the average SP is 90.49% and the $Precision$ is 24,27%. As it can be observed in fig. 5.3 the lesions with the highest SE are: AK, DF and NV. It may be related to the statements presented in chapter 3 (for example, AK is more frequent in head/neck, DF in the lower extremity and the median age for NV is lower than all the others). The most problematic class is MEL, which is only correctly diagnosed in 10% of the cases. Therefore, it can be concluded that metadata alone is not sufficient to achieve a reasonable classification result.

5.4.3 Classification with images and metadata

As described in section 4.3.2, experiments with different fusion methods were carried out. For each method, ResNet-101 and Xception architectures were compared as well. In all the methods, the initial learning rate used is equal to $5 \cdot 10^{-5}$, and it decreases by a factor of 0.75 if the validation loss function does not decrease during 2 epochs in a row.

Method 1

Concerning ResNet, the best model was achieved in epoch 22, and in the epoch 33, for Xception. For each architecture, the confusion matrix for the validation set is depicted in fig. 5.4.



(a) ResNet

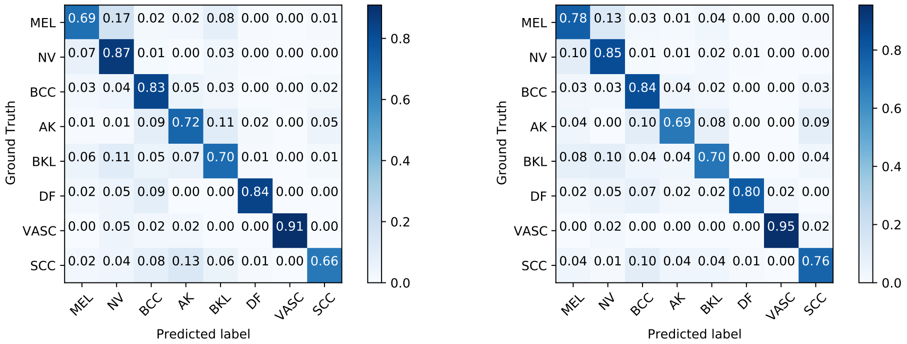
(b) Xception

Figure 5.4: Normalized confusion matrices to method 1 with image and metadata, in the validation set.

By looking at fig. 5.4, in ResNet architecture, the most problematic classes are MEL, BKL and SCC. MEL is confused with NV 19% of the time, BKL 11 % with NV, and SCC 10% with BCC and 10% with BKL. In Xception, the class that deviates more from 1 is AK, which is very confused with BCC and BKL. In both cases, the class whose SE is closest to 1 is VASC.

Method 2

The saved weights belong to epoch number 29, for ResNet, and epoch 19, for Xception. For each architecture, the confusion matrix for the validation set can be seen in fig. 5.5.



(a) ResNet

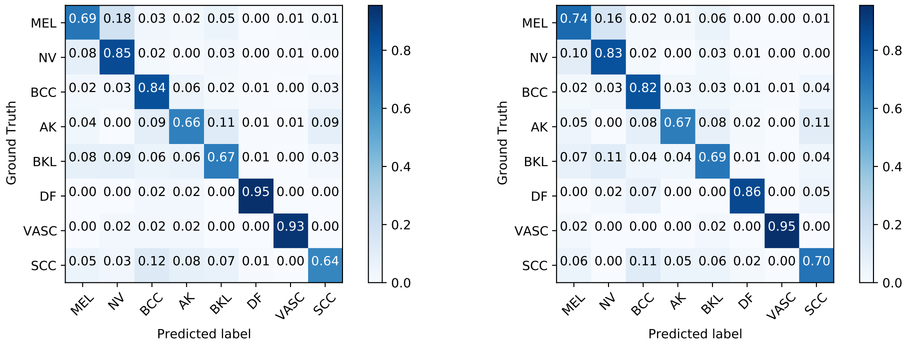
(b) Xception

Figure 5.5: Normalized confusion matrices to method 2 with image and metadata, in the validation set.

In ResNet architecture, the class that deviates more from 1 is SCC. The most significant error, in this class, is in AK class, since SCC is confused as AK in 13% of the cases. Regarding Xception, the biggest confusion is in AK, which is more confused with SCC, BKL and BCC. Once again, in both cases, the most accurate class is VASC.

Method 3

Regarding the ResNet architecture, the best model was achieved in epoch number 22, and in the epoch 14, for Xception. The confusion matrices obtained to the validation set are depicted in fig. 5.6.



(a) ResNet

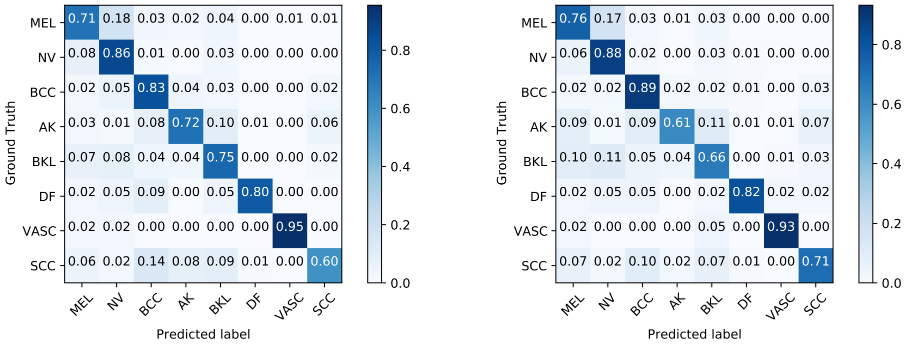
(b) Xception

Figure 5.6: Normalized confusion matrices to method 3 with image and metadata, in the validation set.

When analyzing the confusion matrices, SCC is, again, the class that generates more confusion in ResNet, and AK is also, again, the most problematic class in Xception. SCC is more confused with BCC, and AK is more confused with SCC, BKL and BCC. In Xception the most accurate class is VASC. Nevertheless, in this case, the class whose SE comes closest to 1, in ResNet, is DF.

Method 4

The best model was achieved in epoch number 24, in ResNet architecture, and the epoch 23, for Xception. The confusion matrices obtained to the validation set are outlined in fig. 5.7.



(a) ResNet

(b) Xception

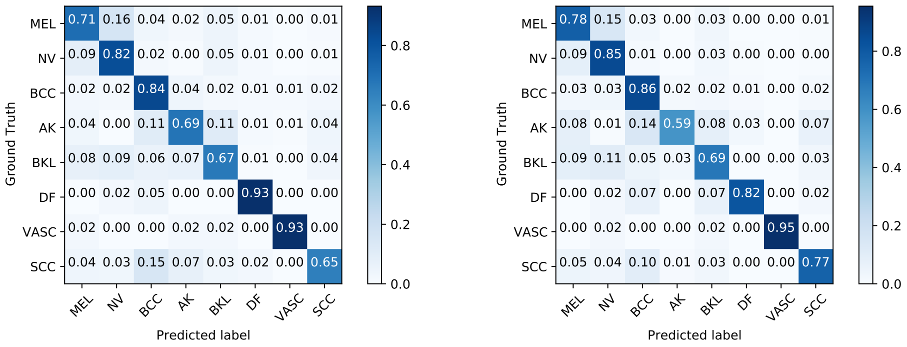
Figure 5.7: Normalized confusion matrices to method 4 with image and metadata, in the validation set.

Again, in this method SCC and AK are the classes with the largest deviation from 1 in ResNet and Xception, respectively, and VASC is the class with the least confusion. The class that is more confused with SCC, in ResNet, is BCC, and with AK, in Xception, is BKL, MEL and BCC.

Method 5

The best value of the hyperparameter α (depicted in fig. 4.9) was 0.2, in both architectures. The best value of BACC was reached in epoch number 19, for ResNet architecture, and in the epoch 25, for Xception. The confusion matrices obtained to the validation set are outlined in fig. 5.8.

In ResNet, the largest deviation from 1 occurs in SCC, and in Xception is in AK. Both classes are more confused with BCC.



(a) ResNet

(b) Xception

Figure 5.8: Normalized confusion matrices to method 5 with image and metadata, in the validation set.

In all methods, SCC is the most challenging class in ResNet (in general, more confused with BCC), and AK is the most challenging class in Xception (in general, more confused with BCC and BKL), with more significant deviations from 1, in relation to the other classes. This misdiagnosed makes sense, since BKL is the benign form of AK and both AK and BCC are Non-Melanocytic and Malign. The VASC is often the most accurate class in both architectures.

5.4.4 Comparison

This section shows the values of the metrics obtained for each method, which were computed from the respective covariance matrices. The results for the validation set are shown in table 5.2 and in table 5.4, and for the testing set in table 5.3 and in table 5.5. The tables contain the values of SE per class and the value of BACC. In addition, the average of SP and the $Precision$ are also shown for the validation set. In order to see the improvements brought by each method, which fuses images and metadata, in relation to the network without metadata, an additional line with improvements is included below the results of each method. The values shown in these lines are the difference between the respective column, with the value obtained without metadata (first line of the table). If the difference is greater than 2%, the difference value will be in green color. On the other hand, if it is under -2%, it is in red.

Table 5.2: Comparison of metrics between all methods that combine images and metadata and the model with only images as input, in the validation set with the ResNet architecture. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE										
Model	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC	avg. SP	$Precision$
No metadata	68.43	78.28	81.40	65.09	66.11	72.72	97.72	67.21	74.62	96.01	62.40
Method 1	67.17	84.66	84.34	68.64	67.04	88.64	97.73	67.21	78.18	96.46	69.49
Improvements	-1.26	+6.38	+2.94	+3.55	+0.93	+15.92	+0.01	0.00	+3.56	+0.45	+7.09
Method 2	68.78	87.32	83.26	72.19	70.00	84.09	90.91	65.57	77.76	96.71	68.98
Improvements	+0.35	+9.04	+1.86	+7.10	+3.89	+11.37	-6.81	-1.54	+3.14	+0.70	+6.58
Method 3	69.47	85.27	83.57	65.68	67.41	95.45	93.18	63.93	78.00	96.57	67.11
Improvements	+1.04	+6.99	+2.17	+0.59	+1.30	+22.73	-4.54	-3.28	+3.38	+0.56	+4.71
Method 4	71.20	86.41	83.26	71.60	75.00	79.55	95.45	59.84	77.79	96.76	71.69
Improvements	+2.77	+8.13	+1.86	+6.51	+8.89	+6.83	-2.27	-7.37	+3.17	+0.75	+9.29
Method 5	71.31	82.31	84.19	68.64	66.67	93.18	93.18	64.75	78.03	96.47	64.81
Improvements	+2.88	+4.03	+2.79	+3.55	+0.56	+20.46	-4.54	-2.46	+3.41	+0.46	+2.41

Table 5.3: Comparison of metrics between all methods that combine images and metadata and the model with only images as input, in the testing set with the ResNet architecture. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE								
Model	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC
No metadata	55.35	72.46	71.29	45.72	41.26	48.90	55.44	27.38	52.22
Method 1	61.18	80.40	72.55	45.45	43.31	51.11	53.46	33.12	55.07
Improvements	+5.83	+7.94	+1.26	-0.27	+2.05	+2.21	-1.98	+5.74	+2.85
Method 2	64.10	81.19	69.10	50.80	45.98	57.78	50.50	28.66	56.01
Improvements	+8.75	+8.73	-2.19	+5.08	+4.72	+8.88	-4.94	+1.28	+3.79
Method 3	61.91	77.52	74.95	45.45	43.46	53.33	48.51	29.30	54.30
Improvements	+6.56	+5.06	+3.66	-0.27	+2.20	+4.43	-6.93	+1.92	+2.08
Method 4	61.75	79.63	74.63	37.97	48.66	45.56	57.43	27.39	54.13
Improvements	+6.40	+7.17	+3.34	-7.75	+7.40	-3.34	+1.99	+0.01	+1.91
Method 5	58.75	74.36	72.13	53.21	43.62	53.33	53.47	29.94	54.85
Improvements	+3.40	+1.90	+0.84	+7.49	+2.36	+4.43	-1.97	+2.56	+2.63

Table 5.4: Comparison of metrics between all methods that combine images and metadata and the model with only images as input, in the validation set with the Xception architecture. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE										
Model	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC	avg. SP	$Precision$
No metadata	71.54	83.68	83.88	57.99	66.67	81.82	90.91	68.03	75.56	96.33	69.41
Method 1	72.93	89.29	85.74	60.36	71.85	84.09	93.18	68.03	78.18	96.97	73.09
Improvements	+1.39	+5.61	+1.86	+2.37	+5.18	+2.27	+2.27	0.00	+2.62	+0.64	+3.68
Method 2	78.23	84.74	84.34	68.64	70.00	79.55	95.45	76.23	79.65	96.90	71.25
Improvements	+6.69	+1.06	+0.46	+10.65	+3.33	-2.27	+4.54	+8.20	+4.09	+0.57	+1.84
Method 3	73.96	83.26	82.48	66.86	69.07	86.36	95.45	70.49	78.49	96.57	67.50
Improvements	+2.42	-0.42	-1.40	+8.87	+2.40	+4.54	+4.54	+2.46	+2.93	+0.24	-1.91
Method 4	75.58	87.51	89.15	60.95	66.30	81.82	93.18	71.31	78.22	96.94	70.51
Improvements	+4.04	+3.83	+5.27	+2.96	-0.07	0.00	+2.27	+3.28	+2.66	+0.61	+1.10
Method 5	78.23	85.50	86.05	58.58	69.07	81.82	95.45	77.05	78.97	96.86	73.04
Improvements	+6.69	+1.82	+2.17	+0.59	+2.40	0.00	+4.54	+9.02	+3.41	+0.53	+3.63

Table 5.5: Comparison of metrics between all methods that combine images and metadata and the model with only images as input, in the testing set with the Xception architecture. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE								
Model	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC
No metadata	62.56	77.05	69.31	24.60	43.31	54.44	41.68	31.21	50.52
Method 1	59.81	82.50	70.15	31.28	45.83	62.22	51.49	40.76	55.50
Improvements	-2.75	+5.45	+0.84	+6.68	+2.52	+7.78	+9.81	+9.55	+4.98
Method 2	63.86	77.60	71.40	39.57	41.10	54.44	51.49	38.89	54.79
Improvements	+1.30	+0.55	+2.09	+14.97	-2.21	0.00	+9.81	+7.68	+4.27
Method 3	63.13	76.93	65.87	33.42	44.88	53.33	47.52	42.04	53.39
Improvements	+0.57	-0.12	-3.44	+8.82	+1.57	-1.11	+5.84	+10.83	+2.87
Method 4	62.24	80.30	70.88	34.22	42.80	58.89	52.48	36.94	54.84
Improvements	-0.32	+3.25	+1.57	+9.62	-0.51	+4.45	+10.80	+8.73	+4.32
Method 5	62.07	76.80	72.13	24.87	41.42	52.22	45.54	35.03	51.26
Improvements	-0.49	-0.25	+2.82	+0.27	-1.89	-2.22	+3.86	+3.82	+0.74

Xception and Resnet extract features with different image properties, since Xception has inception modules and ResNet residual modules. This may justify the different performances achieved with both methods. In the classification using only images, ResNet architecture achieves BACC equal to 74.62% in the training set and 52.22% in the testing set. Xception reached 75.56% and 50.52%. In both cases, VASC is the class with the best SE, in the validation set. Nevertheless, for the testing set, the best SE is in NV class, for both architectures.

All the methods that combine images with metadata lead to improvements in the BACC scores, both for the validation and testing sets. This improvement was observed for both architectures. Additionally, in both architectures SP has reached small improvements.

Regarding ResNet architecture (table 5.2), the higher SE improvements happen in DF and NV lesions. In the method 3, DF improved 22.73% (it has reached a SE = 95.45%). In the testing set, there is also an improvement in those 2 classes in all the methods (except to DF in method 4). In the validation set, there is also a reasonable improvement in AK. The class SCC got worse with the introduction of metadata (the same is not observed on the testing set). As VASC has a great SE in the classification with images only on the validation set, just method 1 led to minimal improvements in the SE of this class. All the other methods made it worse. $Precision$ achieved great improvements.

In the case of Xception, the scores of the MEL, AK and VASC classes seem to improve in all the methods for the validation set. Although MEL does not exhibit the same behavior in the testing set, AK and VASC do. SCC also shows significant improvements in both the validation and testing sets. There were improvements in the $Precision$ (except for method 3).

Therefore, it is possible to conclude that the incorporation of the metadata does not benefit the lesions in the same way. Moreover, it seems to depend on the CNN architecture used to process the dermoscopic images. While in ResNet the classes with the most significant improvements are NV, DF and AK, in Xception these classes are MEL, AK, VASC, SCC. BKL does benefit in both cases. This may be due to the features extracted by both architectures, which may be different. The improvements for each class also depends on the method used to incorporate the metadata (for instance, in Xception, method 3 led to a 4.54% improvement in DF, while method 2 led to a 2.27% decrease).

Table 5.6 summarizes the results obtained with all methods. In order to compare the results with the state-of-the-art, the ISIC leaderboard [44] was analyzed. In ISIC, the classification is based on weighted accuracy (weighted average of the SE). The winner of the challenge presents the best weighted accuracy. Nevertheless, since in this thesis the same importance is given to all the classes, even if they contain a different number of examples, the comparisons are made using the BACC score, and without taking into account the class unknown. The winner of the 2019 challenge presents a BACC equal to 50.93%. This value was obtained by computing the average SE of the 8 classes. In this thesis, only the Xception architecture, with just images, got worse than 50.93%.

Table 5.6: Summary of BACC across all methods for the test set.

Architecture	Just Image	Method 1	Method 2	Method 3	Method 4	Method 5
ResNet	52.22	55.07	56.01	54.30	54.13	54.85
Xception	50.52	55.50	54.79	53.39	54.84	51.26

As can be seen, ResNet seems to generalize better than Xception, because in almost all methods ResNet achieves a better BACC in the test set, even when Xception gets a better result in the validation set. Method 2 with ResNet seems to be the most robust method.

5.5 Effect of each type of metadata feature

In chapter 3, some conclusions were drawn when the metadata was analyzed. The improvements obtained when metadata is taken into consideration may be related to hypotheses defined in chapter 3. The networks may take advantage of some relationships in metadata to improve the distinction of some classes. For instance, in ResNet there were noticeable improvements in NV, DF, AK. It may be related to one of the many facts highlighted in chapter 3, such as that DF is more frequent in the lower extremity, AK in head/neck, and NV in the anterior torso. Moreover, the median age associated with NV is lower than the others.

In order to study the influence of each combination of metadata feature separately and analyze the hypotheses proposed in chapter 3, all the different combinations of metadata were tested. These experiments were performed with method 2, for both architectures, since the best result was obtained with this approach. Therefore, the input size from the metadata network depends on the features being used. For example, if only age is used, the size will be 18, if age and gender is used at the same time, the size will be 20. The remaining training conditions were the same as those used with all features.

Table 5.7 and table 5.9 show the SE per class obtained with ResNet and Xception architectures, respectively for the validation set. Table 5.8 and table 5.10 illustrate the results for the testing set. As specified before, the improvements are in relation to the first line.

Table 5.7: Comparison of metrics between all the metadata combinations and the model with only images as input, in the validation set with ResNet architecture, using method 2. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE								
Features	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC
No metadata	68.43	78.28	81.40	65.09	66.11	72.72	97.72	67.21	74.62
Age	75.81	82.00	81.24	66.86	70.19	72.73	90.91	70.49	76.28
Improvements	+7.38	+3.72	-0.16	+1.77	+4.08	+0.01	-6.81	+3.28	+1.66
Gender	68.54	76.69	78.91	57.40	75.19	79.55	90.91	70.49	74.71
Improvements	+0.11	-1.59	-2.49	-7.69	+9.08	+6.83	-6.81	+3.28	+0.09
Site	65.55	82.42	80.16	59.76	75.19	81.82	95.45	68.85	76.15
Improvements	-2.88	+4.14	-1.24	-5.33	+9.08	+9.10	-2.27	+1.64	+1.53
Age + Gender	69.59	85.73	85.12	60.95	70.56	86.36	93.18	64.75	77.03
Improvements	+1.16	+7.45	+3.72	-4.14	+4.45	+13.64	-4.54	-2.46	+2.41
Age + Site	67.86	84.70	80.47	64.50	72.41	72.73	97.72	71.31	76.46
Improvements	-0.57	+6.42	-0.93	-0.59	+6.30	+0.01	0.00	+4.10	+1.84
Gender + Site	62.33	84.17	79.84	67.46	67.41	81.82	97.73	68.03	76.10
Improvements	-6.10	+5.89	-1.56	+2.37	+1.30	+9.10	+0.01	+0.82	+1.48
Age + Gender + Site	68.78	87.32	83.26	72.19	70.00	84.09	90.91	65.57	77.76
Improvements	+0.35	+9.04	+1.86	+7.10	+3.89	+11.37	-6.81	-1.54	+3.14

Table 5.8: Comparison of metrics between all the metadata combinations and the model with only images as input, in the testing set with ResNet architecture, using method 2. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE								
Features	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC
No metadata	55.35	72.46	71.29	45.72	41.26	48.90	55.44	27.38	52.22
Age	63.45	71.40	65.76	44.12	46.93	44.44	49.50	42.04	53.46
Improvements	+8.10	-1.06	-5.53	-1.60	+5.67	-4.46	-5.94	+14.66	+1.24
Gender	57.86	66.21	68.78	34.76	53.39	45.56	49.50	37.58	51.71
Improvements	+2.51	-6.25	-2.51	-10.96	+12.13	-3.34	-5.94	+10.20	-0.51
Site	58.10	74.48	70.35	41.98	48.97	38.89	49.50	27.39	51.21
Improvements	+2.75	+2.02	-0.94	-3.74	+7.71	-10.01	-4.94	+0.01	-1.01
Age + Gender	58.51	78.45	70.50	38.22	46.34	44.44	47.52	30.60	51.82
Improvements	+3.16	+5.99	-0.79	-7.50	+5.08	-4.46	-7.92	+3.22	-0.40
Age + Site	59.89	75.07	69.10	43.32	50.55	43.33	50.50	37.58	53.67
Improvements	+4.54	+2.61	-2.19	-2.40	+9.29	-5.57	-4.94	+10.20	+1.45
Gender + Site	52.11	73.81	66.70	47.69	38.26	45.56	49.50	37.58	51.40
Improvements	-3.24	+1.35	-4.59	+1.97	-3.00	-3.34	-5.90	+10.20	-0.82
Age + Gender + Site	64.10	81.19	69.10	50.80	45.98	57.78	50.50	28.66	56.01
Improvements	+8.75	+8.73	-2.19	+5.08	+4.72	+8.88	-4.94	+1.28	+3.79

Table 5.9: Comparison of metrics between all the metadata combinations and the model with only images as input, in the validation set with Xception architecture, using method 2. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE								
Features	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC
No metadata	71.54	83.68	83.88	57.99	66.67	81.82	90.91	68.03	75.56
Age	77.07	88.50	89.15	60.95	66.48	72.73	90.91	71.31	77.14
Improvements	+5.53	+4.82	+5.27	+2.96	-0.19	-9.09	0.00	+3.28	+1.58
Gender	77.76	79.20	84.03	59.17	65.56	81.82	88.64	67.21	75.42
Improvements	+0.22	-4.48	+0.15	+1.18	-1.11	0.00	-2.27	-0.82	-0.14
Site	73.16	81.36	88.37	61.54	64.44	81.82	93.18	72.95	77.10
Improvements	+1.62	-2.32	+4.49	+3.55	-2.23	0.00	+2.27	+4.92	+1.54
Age + Gender	73.04	86.45	84.34	56.21	73.52	79.55	90.91	72.95	77.12
Improvements	+1.50	+2.77	+0.46	-1.78	+6.85	-2.27	0.00	+4.92	+1.56
Age + Site	75.35	85.46	86.51	65.09	66.48	81.82	95.45	72.13	78.54
Improvements	+3.81	+1.78	+2.63	+7.10	-0.19	0.00	+4.54	+4.10	+2.98
Gender + Site	72.81	86.64	86.67	63.91	73.33	77.27	95.45	71.31	78.42
Improvements	+1.27	+2.96	+2.79	+5.92	+6.66	-4.55	+4.54	+3.28	+2.86
Age + Gender + Site	78.23	84.74	84.34	68.64	70.00	79.55	95.45	76.23	79.65
Improvements	+6.69	+1.06	+0.46	+10.65	+3.33	-2.27	+4.54	+8.20	+4.09

Table 5.10: Comparison of metrics between all the metadata combinations and the model with only images as input, in the testing set with Xception architecture, using method 2. In the Improvements lines, if the difference is greater than 2%, this value will be in green color. On the other hand, if it is under -2%, it is in red.

	SE								
Features	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	BACC
No metadata	62.56	77.05	69.31	24.60	43.31	54.44	41.68	31.21	50.52
Age	63.45	82.71	73.38	29.14	45.35	44.44	47.52	30.57	52.07
Improvements	+0.89	+5.66	+4.07	+4.54	+2.04	-10.00	+5.84	-0.64	+1.55
Gender	66.40	71.86	69.10	26.20	40.47	53.33	39.60	43.31	51.28
Improvements	+3.84	-5.19	-0.21	+1.60	-2.84	-1.11	-2.08	+12.10	+0.76
Site	63.86	73.09	77.04	33.69	38.74	43.33	51.48	40.13	52.67
Improvements	+1.30	-3.96	+7.73	+9.09	-4.57	-11.11	+9.80	+8.92	+2.15
Age + Gender	58.51	78.45	70.46	38.24	46.30	44.44	47.52	30.57	51.81
Improvements	-4.05	+1.40	+1.15	+13.64	+2.99	-10.00	+5.84	-0.64	+1.29
Age + Site	63.78	78.70	70.67	44.12	38.59	54.44	51.49	33.12	54.36
Improvements	+1.22	+1.65	+1.36	+19.52	-4.72	0.00	+9.81	+1.91	+3.84
Gender + Site	63.61	80.34	71.19	30.75	42.99	55.60	50.56	40.13	54.40
Improvements	+1.05	+3.29	+1.88	+6.15	-0.32	+1.16	+8.88	+8.92	+3.88
Age + Gender + Site	63.86	77.60	71.40	39.57	41.10	54.44	51.49	38.89	54.79
Improvements	+1.30	+0.55	+2.09	+14.97	-2.21	0.00	+9.81	+7.68	+4.27

The combination that performed better was with Age, Gender and the anatomical site (the one that combines all the metadata information). As can be seen, not all combinations led to improve the results over the model without metadata.

The comparison of the hypotheses from the chapter 3 with the results obtained for each combination are discussed below. This comparison is based on the validation set, since not all lesions that have improved in the validation set exhibited the same behaviour on the testing set. When each combination was analyzed in chapter 3, some lesions were stood out, since they present a specific distribution (based on the variables of that combination) that can allow the network to learn some relationships between the lesions and the metadata, and improve the distinction of those lesions. After training both networks with all combinations of metadata, the improvements of each class, in relation to the model without metadata, were analyzed and some hypotheses were verified, when some combinations of metadata were used.

- **Age**: When fig. 3.3 was analyzed, it was concluded that the age can be useful to differentiate some lesions, namely NV, SCC, and AK. In the ResNet architecture there were improvements in the SE of NV and SCC, and in Xception in NV, SCC and AK. This seems to support the hypothesis made in chapter 3 (that age can be useful to distinguish these lesions, since their SE improved). Using only age led to improve the BACC in both architectures, in the validation and testing sets.

- **Gender**: The main differences noticed in the gender appear in BCC, BKL, MEL, and SCC (fig. 3.4). Nevertheless, only SCC and BKL have improved in ResNet (both have more male than female samples). Gender alone does not seem to be an informative feature, as there are basically no improvements in BACC.

- **Site**: The use of the site alone led to small improvements in the ResNet (only in the validation set) and in Xception. When the distribution of the site was discussed in chapter 3, MEL, NV, AK, BKL and DF were highlighted, once they appear in specific parts of the body more often. In ResNet the improvements of NV, BKL, and DF are highlighted, and in Xception BCC, AK, VASC, and SCC. Some relationships between this metadata information and some of these lesions seem to be learned by the networks.

- **Age and Gender**: Regarding the combination of age and gender, in both architectures, there were improvements in NV and BKL. Moreover, in ResNet, BCC and DF have improved, while in Xception SCC has improved. In both cases, MEL has improved less than 2%. All those lesions, except DF, have been referred to when the bi-dimensional distribution with the variables age and gender was described. As such, it seems to support the hypotheses made in chapter 3. This combination of features led to improvements, in relation to the case with just images, in both cases in the validation set. In the testing set, it got a worse score for the ResNet architecture.
- **Age and Site**: NV, BKL, DF, AK, and VASC were stood out for being more frequent in specific regions of the body, in specific ranges of age. The first two classes have improved considerably in the case of the ResNet architecture. As such, the hypotheses created (that say that this combination can be useful to distinguish these lesions) seem to be supported with the use of the ResNet. The last two classes have improved considerably in the case of the Xception. This way, the hypotheses created for these two lesions seem to be supported for Xception. This combination of features was helpful to increase their SEs. In both cases, the BACC improved in relation to the method with just images.
- **Gender and Site**: In this combination of features, the highlighted classes were: AK, DF, SCC and VASC. The first two have improved more than 2% in the ResNet experience (it seems to support the hypothesis). With Xception, almost all the classes, except DF, have improved the SE. In both cases, the BACC improved over the method without metadata (in the ResNet test set, it did not improve).

In chapter 3 some hypotheses were created, based on different combinations of features. By training the networks just with those combinations separately, it was possible to analyze which networks learned the hypotheses created, and led to improve the distinction of some lesions, in relation to the model with just images. The introduction of each metadata feature does not benefit the same classes in the same way for the two compared CNN architectures. Recall that the experiments presented in this section were performed using method 2, where metadata controls each feature channel of the CNN model. It works as a feature selector, which assign higher values to the most relevant features. Therefore, metadata can help, for instance, to remove the chance of being a particular lesion. As the CNN architectures extract image features using different strategies, and there are relations between metadata and each lesion, this feature selection process may differ according to the architecture used, resulting in different improvements. Some relationships defined in chapter 3 might be learned by one the architectures, and taken into account in the classification, leading to significant improvements. Nevertheless, it does not mean that the other architecture has learned the same correlations. Therefore, some hypotheses created with a specific combination of features seemed to be supported for one architecture (but it does not mean that were supported by the other).

Not all combinations of metadata led to improve the results, and some of them are more beneficial in one CNN architecture in relation to the other. The combination that led to the best result, in the validation and testing sets, is the one that combines all the metadata features: age, gender and anatomical site. This was observed for both CNN architectures.

6

Web site application

Contents

In this chapter, a website application is presented. All the functionality of the website and the back-end architecture are introduced. Finally, the front-end images are illustrated, with a practical example. A complete example of how the website application works is illustrated in a video here:
https://www.youtube.com/watch?v=cwCnXPRWa1o

6.1 Goal and functionalities

There are many centers that have already begun to research on automated skin lesion diagnosis, but, a centralized, coordinated, and comparative effort across institutions has not yet been implemented [16]. The website developed aims to represent a type of application that can be used by dermatologists in the future, to support them in the detection of skin cancer. It is a simple application, in which the user uploads a dermoscopic image and inserts the patient's information and, as soon as the user submits the information, receives an automatic diagnosis. To create a website application that can be used by different institutions and multiple users at the same time, a scalable and fault-tolerance application is needed. However, as it is not the focus of this work, this website is just a simple example that has not been tested for these specifications. There is room for improvement. The final version of the website was not deployed to the cloud. This means that the website is running locally.

This web application is divided into two main parts: client and server. The client is a front-end that sends the patient's information to the server, receives, and displays the result. When the server is initialized, it builds the diagnostic model based on a deep neural network and loads the weights. After receiving an image and the patient's information, the server feeds the input into the model, performs the prediction, and returns the result to the client.

6.2 Server architecture

The `HTTP` back-end server was developed in `Python`, with the web framework `Django`. In this framework, each URL has associated an event, which facilitates the implementation. The API created is based in `REST` architecture style, where `GET`, `POST`, `PUT` or `REMOVE` requests are made. In this API, two different $endpoints$ have been created. Therefore, the interaction between client and server is done through two different ways, according to the intended action. The $endpoints$ are:

- ***Endpoint***: /app/, **Method**: `GET` - The server receives a `GET` request and returns a web page with the user interface (depicted in fig. 6.2).

- ***Endpoint***: /app/lesion/, **Method**: `POST` - The user sends a $json$ in the body of the HTTP request with the metadata information in a code that is decoded by the server, and with the image. The server returns a $json$ with the output of the $Softmax$, and with the diagnosed lesion. This exchange of information is exemplified in section 6.3

All the interaction between the client and the is depicted in fig. 6.1.

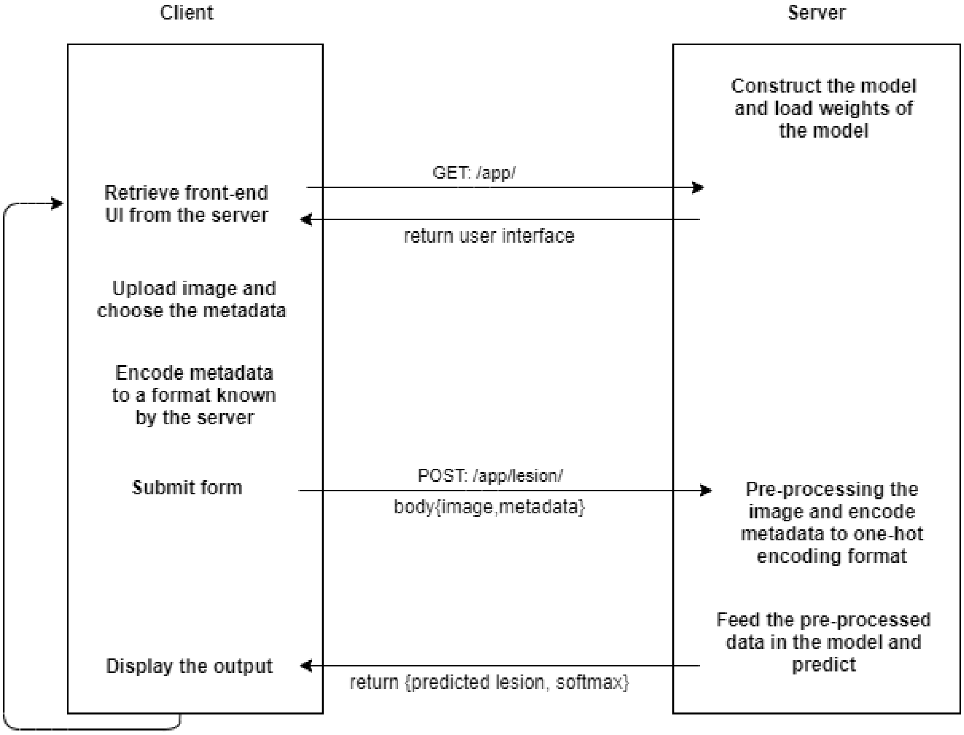

Client
Server
Construct the model and load weights of the model
GET: /app/
Retrieve front-end UI from the server
return user interface
Upload image and choose the metadata
Encode metadata to a format known by the server
Submit form
POST: /app/lesion/
body{image,metadata}
Pre-processing the image and encode metadata to one-hot encoding format
Feed the pre-processed data in the model and predict
Display the output
return {predicted lesion, softmax}

Figure 6.1: Workflow of the interaction between the client and the server.

Initially, when the server is started, the model is created, and the weights are loaded. The chosen model was method 1 with Xception because of the memory. The user opens the browser and executes an `GET` request to the server. The server responds with the user interface, where the user can fill out a form with all the necessary information. When the form is submitted, a `POST` request is performed: the image and the metadata are sent. After receiving the information, the server performs all the pre-processing on the image and in the metadata (build the one-hot encoding vector), feeds the inputs into the model, and makes the prediction. After predicting, the output of the $Softmax$ and the output label are returned to the client, which displays the output label and the output of the $Softmax$.

6.3 Front-end and a practical example

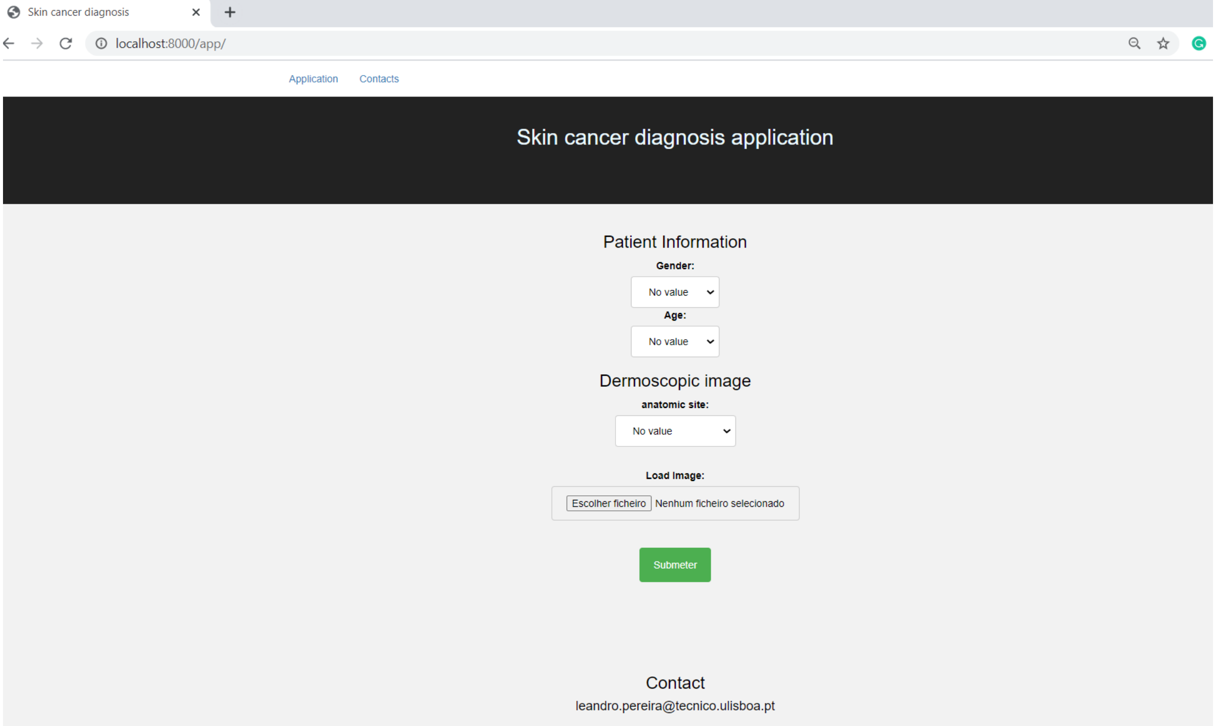
All the front-end was developed with `HTML` and `React`. When the user opens the browser and searches with the URL localhost/app/ (because it is running in localhost mode), a `GET` request to the server is performed. The server returns a webpage, that is depicted in fig. 6.2. Afterward, the user fills out the form, uploads an image and submits the form. It is exemplified in fig. 6.3

As mentioned above, a POST request is carried out when the submit takes place. As far as the body of the POST request goes, a $json$ with the image and with the metadata is sent, inside the body of the request. As already stated in chapter 4, the metadata is represented with a one-hot encoding vector. As such, the value of each field of the $json$ represents the positions of the one-hot encoding vector that has to be activated (value 1). In the example represented in fig. 6.3 the $json$ wil be given by:

```
{
"image": ISIC_000001.jpg (array of the image),
"gender": "0",
"anatomic_site": "2" ,
"age": "16"
}
```

The gender field is ”0” since it is the position associated with the Female gender, and anatomic_site field is 2 once it is the position of the anterior torso in the one-hot encoding vector, which will be built by the server. The same reasoning is applied to the age. Nevertheless, if one of those values were empty, the client would send the value ”100”. After receiving this information and perform all the pre-processing, the server makes the prediction and returns a $json$, inside the body, back to the client, with the output of the prediction, and the diagnosed lesion. In this example, the $json$ will be:

```
 {
 "softmax": {
      "MEL": "1.7210988e-6",
      "NV": "0.9999982",
      "BCC":2.370179e-10" ,
      "AK": "1.3885159e-12",
      "BKL": "6.1323355e-08",
      "DF":"4.098048e-12",
      "VASC": "1.3586243e-12",
      "SCC": "7.9694125e-12"
  },
  "label": {
      "lesion": "NV - Melanocytic nevus"
  }
}
```



Gender: No value

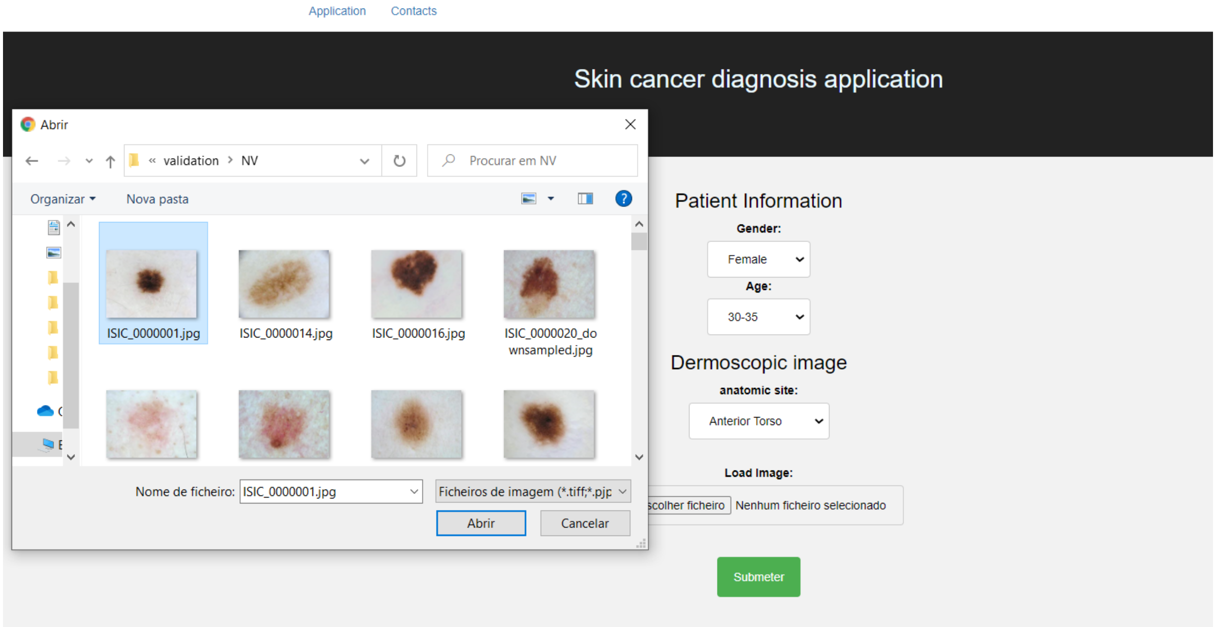
Age: No value

anatomic site: No value

Load Image: Nenhum ficheiro selecionado

leandro.pereira@tecnico.ulisboa.pt

Figure 6.2: User interface. Main page.



Nome de ficheiro: ISIC_0000001.jpg

Gender: Female

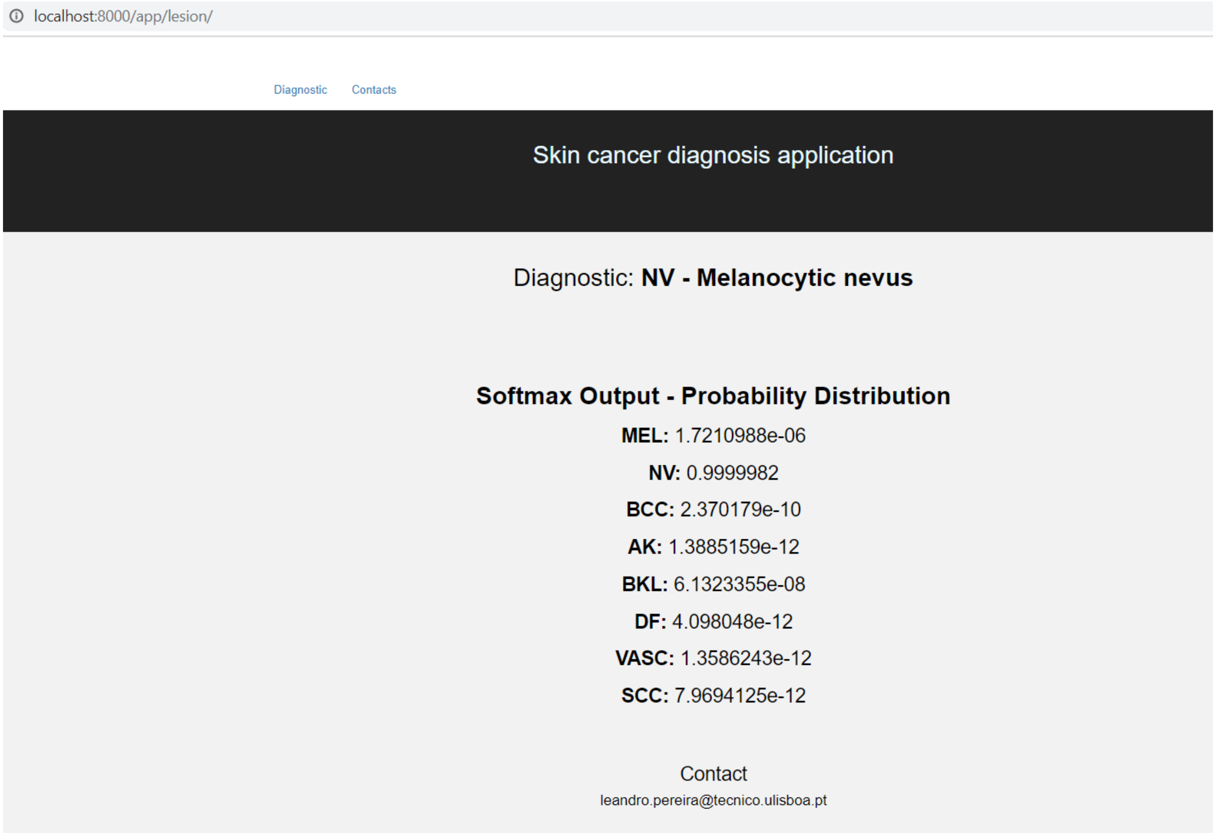
Age: 30-35

anatomic site: Anterior Torso

Load Image: Nenhum ficheiro selecionado

Figure 6.3: Filling the form. Selecting a dermoscopic image.

When the server receives the response, displays the output, as outlined in fig. 6.4.



Skin cancer diagnosis application

Diagnostic: **NV - Melanocytic nevus**

Softmax Output - Probability Distribution

MEL: 1.7210988e-06

NV: 0.9999982

BCC: 2.370179e-10

AK: 1.3885159e-12

BKL: 6.1323355e-08

DF: 4.098048e-12

VASC: 1.3586243e-12

SCC: 7.9694125e-12

Contact

leandro.pereira@tecnico.ulisboa.pt

Figure 6.4: User interface. Output page, where NV was the diagnosed lesion.

7

Conclusions and Future Work

Contents

This chapter presents the conclusions of this work and some suggestions for the future work.

7.1 Conclusions

This thesis aimed to understand if there are improvements when the patient's information (age, sex, body region) are incorporated into an automatic decision system that diagnose skin lesions. To accomplish it, this thesis considers three types of models: a CNN for the diagnosis of dermoscopic images, a multi-layer perceptron for diagnosis based on metadata only, and a deep learning model that integrates both images and metadata. For the diagnosis of dermoscopic images, ResNet-101 and Xception CNN architectures were used. Regarding the combination of images and metadata, five different methods that combine these covariates with images were developed and compared.

Each one of these methods was consisted of combining a CNN, previously trained just with dermoscopic images (using either ResNet or Xception architectures), with a multi-layer perceptron output, used for diagnosis based on metadata only. How this fusion is performed depends on the method. In all experiments performed, the hyperparameters were adjusted in order to select the best performing configuration (according to the metric BACC) in the validation set. Then, it was applied to the testing set.

The results show that using only metadata does not lead to a reasonable classification result. All strategies that combine images and metadata performed better than the respective strategy without metadata, both in the validation set and in the testing set. Thus, it is concluded that patient information improves the performance of the system. Method 2 with ResNet was the best overall method. It achieved a BACC of 77.76% for the validation set and 56.01% for the testing set. It led to an improvement of 3.14% and 3.79% in the validation and the testing set, respectively, compared to the model without metadata. In this configuration, the fusion is performed with a multiplication operation.

The incorporation of metadata did not benefit all the classes in the same way across the two CNN architectures. It seems to depend on the CNN architecture used to process the dermoscopic image, since these architectures extract features differently. This analysis was performed based on the SE of each lesion. The classes with the biggest improvements in ResNet were not the same as for Xception. In addition, it was stood out that the most challenging class, in general, is different between the two CNN architectures.

In order to study the influence of each type of metadata feature, all different combinations of metadata were tested, using method 2, trained with both ResNet and Xception, to analyze which combination has the most influence on the classification, and to analyze the hypotheses proposed in chapter 3. These hypotheses say that some combinations of metadata may be helpful to improve the SE of certain lesions, since they may be correlated. The networks can take advantage of some relationships between the lesions and the patient's information, to improve the distinction of some classes. The combination that performed better was with Age, Gender and the anatomical site (the one that combines all the metadata information). In addition, some hypotheses proposed were supported. For example, since NV is more frequent in some regions of the body in specific ranges of age (in this case, in the anterior torso between 30 and 55 years old), the introduction of the combination Age + anatomical Site seemed to be helpful to diagnose this lesion, since it led to improve the SE of this lesion in the validation set, for Xception and ResNet CNN architectures.

Last but not least, a web site application was developed. This website aims to represent a type of application that can be used by dermatologists in the future, to support them in the detection of skin cancer. It is a simple application, in which the user uploads a dermoscopic image and inserts the patient's information and, as soon as the user submits the information, receives an automatic diagnosis. A complete example of how the website application works is available on:
https://www.youtube.com/watch?v=cwCnXPRWa1o

7.2 Future Work

The results obtained in this thesis show the importance of the metadata in the decision system that diagnoses skin lesions. However, there is room to improve the results. The following points show some contents that can be studied in the future to improve the results and to further improve the analysis of the influence of each combination of the metadata, as well as add some features to the website.

- Ensemble the classifiers of the different strategies used, that combine images with metadata. This will make it possible to take advantage of the properties of the different CNN architectures.
- Increase the dataset size, since some lesions contain only a few images. DF and VASC represent around 0.9% and 1% of the dataset, respectively.
- Further analysis of the influence of each metadata combination: try to find correlations between the lesions and metadata, and further improvement models with all the combinations.
- Deployment of the web site application to the Cloud, in order to be online and accessible to all the dermatologists. Ensure that the application is scalable, fault-tolerant. In addition, add a new feature to the web site application that allows automated retraining, in order for the dermatologists add samples, and the system automatically retrains the model. This way, all the dermatologists can contribute to the further improvement of the centralized system. In other words, there would be an option to add a new sample and the respective label, and the system adds this new sample to the training set and retrain the model with the new sample.

Bibliography

[1] "DermNet NZ – All about the skin — DermNet NZ," 2020. [Online]. Available: https://www.dermnetnz.org/

[2] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D Convolutional Neural Networks and Applications-A Survey," *arXiv preprint arXiv:1905.03554*, 2019. [Online]. Available: https://arxiv.org/abs/1905.03554

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Neural Information Processing Systems*, 2012, pp. 1097–1105.

[4] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*, 2015.

[5] C. Szegedy, W. Liu, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

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

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "Rethinking the Inception Architecture for Computer Vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[9] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[11] “Home - The Skin Cancer Foundation - Skin Cancer Facts & Statistics,” 2020. [Online]. Available: https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/

[12] “Ultraviolet (UV) Radiation and Skin Cancer,” 2020. [Online]. Available: https://www.who.int/news-room/q-a-detail/ultraviolet-(uv)-radiation-and-skin-cancer

[13] “Melanoma : Liga Portuguesa Contra o Cancro,” 2020. [Online]. Available: https://www.ligacontracancro.pt/melanoma/

[14] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks,” *Nature Publishing Group*, vol. 542, no. 7639, pp. 115–118, 2017.

[15] W. Li, J. Zhuang, R. Wang, J. Zhang, and W.-S. Zheng, “Fusing Metadata and Dermoscopy Images for Skin Disease Diagnosis,” in *IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 1996–2000.

[16] “Training Data — ISIC 2019.” [Online]. Available: https://challenge2019.isic-archive.com/data.html

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] C. G. Watts, C. Madronio, R. L. Morton, C. Goumas, B. K. Armstrong, A. Curtin, S. W. Menzies, G. J. Mann, J. F. Thompson, and A. E. Cust, “Clinical Features Associated with Individuals at Higher Risk of Melanoma a Population-Based Study,” *JAMA Dermatology*, vol. 153, no. 1, pp. 23–29, 1 2017. [Online]. Available: https://jamanetwork.com/journals/jamadermatology/fullarticle/2580297

[19] C. Barata, M. E. Celebi, and J. S. Marques, “A Survey of Feature Extraction in Dermoscopy Image Analysis of Skin Cancer,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, pp. 1096–1109, 2018.

[20] C. Barata, M. Ruela, M. Francisco, T. Mendonça, and J. S. Marques, “Two Systems for the Detection of Melanomas in Dermoscopy Images Using Texture and Color Features,” *IEEE Systems Journal*, vol. 8, no. 3, pp. 965–979, 2013.

[21] M. E. Celebi, N. Codella, and A. Halpern, “Dermoscopy Image Analysis: Overview and Future Directions,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 474–478, 3 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8627921

[22] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, “A Survey on Deep Learning in Medical Image Analysis,” *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[23] L. Bi, J. Kim, E. Ahn, and D. Feng, “Automatic Skin Lesion Analysis using Large-scale Dermoscopy Images and Deep Residual Networks,” in *arXiv:1703.04197*, 3 2017. [Online]. Available: https://arxiv.org/abs/1703.04197

[24] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer, “Skin Lesion Classification Using Ensembles of Multi-Resolution EfficientNets with Meta Data,” *MethodsX*, vol. 7, no. 100864, 2019.

[25] S. Nofallah and W. Wu, “Transfer Learning for Automatic Disease Diagnosis with Dermoscopic Images,” Tech. Rep., 2018.

[26] N. Heller, E. Bussman, A. Shah, J. Dean, and N. Papanikolopoulos, “Computer Aided Diagnosis of Skin Lesions from Morphological Features,” Tech. Rep., 2018. [Online]. Available: https://www.researchgate.net/publication/327208960_Computer_Aided_Diagnosis_of_Skin_Lesions_from_Morphological_Features

[27] A. Nozdryn-Plotnicki, J. Yap, and W. Yolland, “Ensembling Convolutional Neural Networks for Skin Cancer Classification,” in *International Skin Imaging Collaboration (ISIC) Challenge on Skin Image Analysis for Melanoma Detection. MICCAI*, 2018.

[28] A. G. C. Pacheco and R. A. Krohling, “The Impact of Patient Clinical Information on Automated Skin Cancer Detection,” *Computers in Biology and Medicine*, vol. 116, no. 103545, 2019.

[29] S. Goes, “Introduction to Convolutional Neural Networks,” in *Learning the Scale of Image Features in Convolutional Neural Networks*, 2017, ch. 3, pp. 23–40. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3A4ff9b2e3-2679-42f8-80cd-bd2cb884a466

[30] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, “A Guide to Convolutional Neural Networks for Computer Vision,” *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, 2 2018.

[31] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” in *Nature*, vol. 521, no. 7553. Nature Publishing Group, 5 2015, pp. 436–444. [Online]. Available: https://www.nature.com/articles/nature14539

[32] W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[33] J. M. Ede and R. Beanland, “Adaptive Learning Rate Clipping Stabilizes Learning,” *Machine Learning: Science and Technology*, vol. 1, no. 015011, 6 2019.

[34] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” in *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[35] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A Survey of the Recent Architectures of Deep Convolutional Neural Networks,” *Artificial Intelligence Review*, 2020.

[36] “ImageNet Leaderboard — Papers With Code,” 2020. [Online]. Available: https://paperswithcode.com/sota/image-classification-on-imagenet

[37] P. Tschandl, C. Rosendahl, and H. Kittler, “The HAM10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions,” *Scientific Data*, vol. 5, no. 180161, 3 2018.

[38] M. Combalia, N. C. F. Codella, V. Rotemberg, B. Helba, V. Vilaplana, O. Reiter, C. Carrera, A. Barreiro, A. C. Halpern, S. Puig, and J. Malvehy, “BCN20000: Dermoscopic Lesions in the Wild,” *arXiv preprint arXiv:1908.02288*, 8 2019. [Online]. Available: http://arxiv.org/abs/1908.02288

[39] 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),” in *Proceedings - International Symposium on Biomedical Imaging*. IEEE Computer Society, 5 2018, pp. 168–172.

[40] C. Barata, M. E. Celebi, and J. S. Marques, “Improving Dermoscopy Image Classification Using Color Constancy,” *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 1146–1152, 5 2015.

[41] C. T. Duong, R. Lebret, and K. Abererécole, “Multimodal Classification for Analysing Social Media,” in *arXiv:1708.02099*, 2017. [Online]. Available: https://arxiv.org/abs/1708.02099

[42] W. Li, R. Wang, and W. Zheng, “Skin Lesion Analysis Towards Melanoma Detection with Meta-Data Using Ensemble of Deep Neural Networks,” 2019. [Online]. Available: https://challenge2019.isic-archive.com/leaderboard.html

[43] “Developer guides,” 2020. [Online]. Available: https://keras.io/guides/

[44] “ISIC 2019,” 2020. [Online]. Available: https://challenge2019.isic-archive.com/