Skin Cancer Diagnosis Using Dermoscopic Images and Patient Information

Leandro José Pereira Almeida
Instituto Superior Técnico - Universidade de Lisboa, Portugal
Email: leandro.pereira@tecnico.ulisboa.pt

***Abstract*—Skin cancer is the most common type of cancer worldwide. Early detection leads to an increased survival rate. CAD systems, which process dermoscopic images, can improve the early detection rates.**

In recent years, different CAD (Computer-Aided Diagnosis) 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 work 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 $BACC$. 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.

I. Introduction

Skin cancer is the most common type of cancer worldwide, and the number of cases and deaths has been increasing in the past years [3]. The World Health Organization (WHO) estimates that one in three diagnosed cancers is skin cancer [4].

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% [12]. However, this value drops to about 14% if detected in its latest stages.

The diagnosis of melanoma can benefit from image analysis and machine learning methods to increase the diagnostic accuracy. CAD systems, which process 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.

In addition to dermoscopic images, patient's information (such as the patient's age, gender, anatomical site, family history, among others) is also taken into account by dermatologists to diagnose [29]. However, these covariates have been scarcely used in CAD systems [6]. 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. The incorporation of patient's information in CAD systems is a great and useful challenge, since some lesions that belong to different classes are very similar, and metadata can act to differentiate them.

Several medical methods are used to diagnose dermoscopic images, such as dermoscopy, pattern analysis, 7-point checklist, Menzies method, and ABCD rule [6]. Nevertheless, 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. Different methods have been proposed to tackle this problem. Firstly, systems using low-level image processing methods (edge and line detection, and region growth), then methods based on Machine Learning (Decision trees, Bayesian classifiers, Support Vector Machines, and artificial neural networks) [6]. Nevertheless, these classical machine learning techniques required the extraction of handcrafted features.

In order to overcome this problem, CNN (Convolutional Neural Network) models have been used in recent years. CNNs have become the main approach to solving this kind of problem. 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. ISIC promotes a challenge 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 [7]. In the ISIC 2018, a DenseNet 201 was used in classification task, in [23]. The latter works have used ensemble methods, which combine different architectures. For instance, in [16] an ensemble consisted with ResNet 50, Inception v3, Xception, DenseNet 201 and InceptionResNet v2 was applied. The 2018 challenge winner [24] has also used an ensemble approach.

Recently, studies that combine images with the patient's clinical information have started to appear (for example, in

[22]). In 2019, to further improve the diagnostic performance, the ISIC challenge came with new tasks to consider: one of them is lesion diagnosis with dermoscopic images and metadata [1]. The image's information was completed with the patient's information. The winner of the challenge with an ensembling strategy was Gessert [13]. This work combined the images network with the metadata network by concatenating outputs at the feature level. 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.

In this work, to diagnose the skin lesion of a given patient, the dermoscopic image of the lesion and the patient's information is used as input. The metadata is composed of the patient's age (18 intervals from 0 to 90 years old) and gender, and the region of the body where the skin lesion is located (8 different parts). The classification/diagnostic is a skin lesion. The dataset is composed of 8 types of skin lesions: MEL, NV, BCC, AK, BKL, DF, $VASC$ and SCC.

This thesis aims to answer the following question: "Does combining patient information with dermoscopic images for skin lesion diagnosis can lead to further improvements over just dermoscopic images?". In other words, the goal is to understand if there are 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.

II. Background

A CNN is a class of deep neural networks used with several image-related problems. CNN allows the extraction of features by applying convolutional operators that progressively learn more abstract features. CNN comprises convolutional layers, Fully Connected Layers (FCL), and pooling layers.

A. Convolutional Layer

The main building block of a CNN is the convolutional layer [14]. 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 convolutional layers, kernels can be represented as a 3-dimensional tensor (with shape equal to width × height × number of channels) [14]. Each kernel has a specific width and height but has a depth equal to the number of channels of the input.

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 [18]. 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) [14]. This kernel is evaluated at every possible location.

By applying several kernels in the same convolutional layer, the output of the convolutional layer is a stack of feature-maps [14]. The depth of this stack is equal to the number of kernels used. Each feature-map is a new image, and a nonlinear activation function is applied to each pixel of the feature-map.

B. Pooling layer

The pooling layer merges similar features into one since the relative positions of the similar features can vary somewhat [21]. A pooling layer operates on blocks of the feature map and combines the feature activations [18]. 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 [14]. The pooling operation works as follows: a window slides across the input feature map with a specific stride [18], 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). It is highly beneficial to include pooling layers for relieving the computational load.

C. Fully Connected Layer

After convolutions and pooling layers, CNN has FCL layers. The output of the convolutions and pooling layers are fed in one or more fully connected layers [25]. 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. 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 [18]. It is given by:

$$y = f(W^T x + b) \qquad (1)$$

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 [18]. The last FCL is used to predict the class label [25]. 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.

D. 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 [18]. The activation functions used in deep learning are differentiable in order to allow the backpropagation optimization [18]. The activation functions are applied to convolutional layers and FCL. The most popular nonlinear function is the ReLU [21], since it helps in overcoming the vanishing gradient problem and allows the network to converge very quickly, since it learns much faster in networks with many layers. ReLU is defined by ReLU(z) = max(0,z).

Other activation functions are commonly used, such as $tanh(z)$ and $sigmoid$.

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:

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

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

E. 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 [18].

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). 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:

$$\theta_t = \theta_{t-1} - \eta \frac{\partial L}{\partial \theta}, \tag{3}$$

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 [11].

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 [21]. 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 [21]. 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, Adam, Adaptive Delta (AdaDelta). [18]. 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. This algorithm is computationally efficient with little memory requirements [19].

F. Popular CNN architectures

The development of popular CNN architectures for classification is often linked with the ImageNet Large Scale Visual Recognition Challenge. The most popular architectures are those that participated in the ImageNet challenges.

AlexNet [20] 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.

VGG ranked second in the ImageNet challenge in 2014, showing that it is possible to train deeper networks to achieve better results. In [26] 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, with very small filters.

GoogleNet [27] won ImageNet challenge in 2014. This architecture uses a new structure called inception module. In this module, instead of choosing one size for the filters in each layer, it uses different size filters, and then a concatenation of the feature maps from each filter into one big feature map is performed.

With deeper networks, a degradation problem has been exposed: with the increase of network depth, accuracy gets saturated and then degrades rapidly. Therefore, adding more layers to a previous trained deep model leads to a decrease of the training accuracy [15]. There is a vanishing gradient problem. Therefore, some architectures, such as ResNet and DenseNet, present techniques to improve the information flow between layers in deep networks. In ResNet [15], the traditional convolution blocks were replaced by residual connections. In DenseNet [17], all layers are connected directly with each other.

In 2017 a convolutional neural network architecture based entirely on depthwise separable convolution layers - Xception - was proposed [8]. It is a stronger version of the Inception architecture, which stands for "Extreme Inception" [8]. 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.

III. Methodology

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. 1. Before being fed in the different models to perform the classification, the image and the metadata are pre-processed.

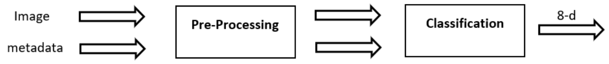
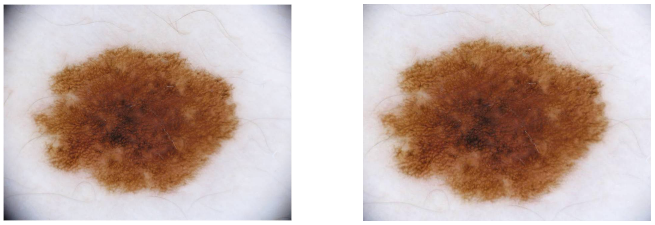


Fig. 1: The main blocks of the proposed system.

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. Two different CNN 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.

A. Pre-processing

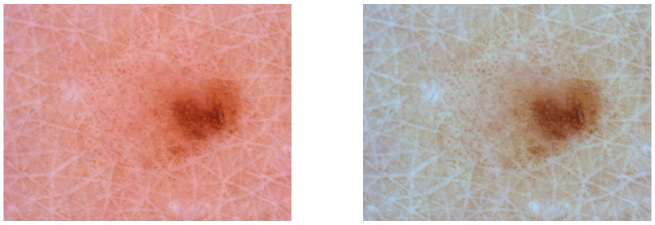
The ISIC dataset comes from different medical centers: HAM10000 [28], BCN 20000, [10], and MSK [9], and was acquired using different equipments. For this reason, the size, the color and the aspect ratio of the images are different. To overcome these differences, pre-processing operations were performed. As far as metadata is concerned, since the metadata contains categorical features, one-hot encoding technique was applied. In relation to images, 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. An example is illustrated in fig. 2.



(a) Original (b) Cropped

Fig. 2: Example of crop transformation in dermoscopic images.

If a system operates with multisource images, 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 [5]. 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 [5]. This method estimates the color of the illuminant, acquired using an unknown light source, and transforms the image, based on this value, to identical colors under a canonical light source. An example is outlined in fig. 3.



(a) Before Normalization (b) After Normalization

Fig. 3: Example of color normalization with color constancy algorithm - Shade of Gray.

As far as metadata is concerned, it 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. The final feature vector has a size 28. In some of the examples, one or more type of metadata may be missing. Thus, all of the features associated with that data will be zero.

B. 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.

Classification using only dermoscopic Images

The diagnostic with images is performed using a CNN. The image is first pre-processed, 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, 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, that will be fed into a FCL with 8 neurons, which performs the decision. This overall scheme is outlined in fig. 4, where it is assumed that the image is already pre-processed.

The configuration of the Convolutional and Pooling Layers block depend on the architecture used. ResNet and Xception were chosen to be used as CNN Model block. 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 7×7 channel). CNN Model block has an image as input and outputs a vector of size 2048. In the next subsections, CNN Model will be shown in the block diagrams. After this block, there is an 8-way fully-connected layer (because there are 8 classes/lesions)

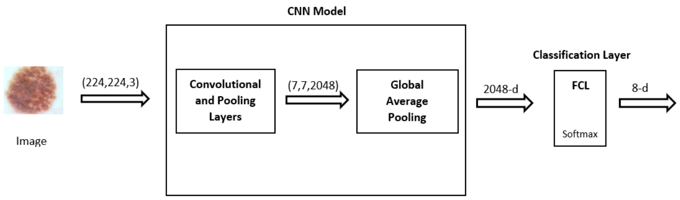


Fig. 4: The model used to classification with dermoscopic images, where a CNN Model block and a Classification layer modules are defined, to be used in the next examples. The Convolutional and Pooling Layers block depends on the model used.

with $Softmax$ as the activation function, that performs the classification. This is called Classification Layer block and will appear in the next subsections.

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 FCL (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 III-A.

Classification with both dermoscopic images and metadata

In order to classify lesions using images and metadata, 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: This method comprises the CNN Model block (the same as the depicted in fig. 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. 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 [13]. 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).

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 dermoscopic images. 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 FCL's 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 [22]. 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. 5.

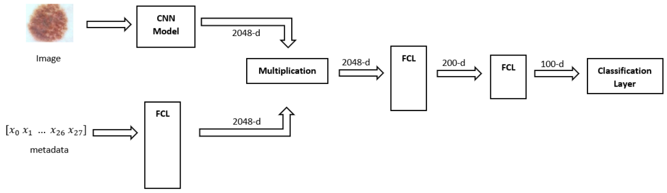


Fig. 5: Method 2: A fusion of metadata and the CNN image model. Fusing architectures by multiplying the outputs at feature level - early fusion.

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 FCL's 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: This 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.

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. Thus, 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.

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) 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. 6.

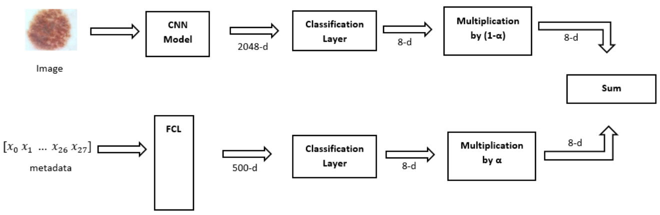


Fig. 6: Method 5: A fusion of metadata and the CNN image network by combining the classifiers. Nevertheless, all the model is re-trained.

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.

C. Training issues

Since the trained models have a large number of parameters, there may be an overfitting problem. Moreover, 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.

Data Augmentation: 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. Thus, 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.

Regarding metadata, data augmentation is necessary, since not all images contain metadata. If a certain piece of metadata is missing, all features of that type will be zero. 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. However, it may encode the gender feature as a missing value, and, in this case, the one-hot encoding input vector will have the entries $Fem.$ and $Male$ equal to zero.

Class weights in Loss Function: 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. As such, the less frequent classes have a higher weight in relation to the others. Thus, 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.

Dropout: In order to handle overfitting problems, Dropout is applied to all FCL, since it is in these layers that exist more weights. In this technique, it sets to zero a subset of hidden neuron randomly chosen with probability p = 0.1.

Transfer Learning: For all CNN architectures pre-trained models were used. It consists of taking features learned on a problem and leveraging them on a new problem [2]. In other words, the initial weights used in our CNN model were obtained from models trained for the classification of the ImageNet dataset.

IV. Experiments and Results

This chapter starts by 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.

A. Dataset

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 [1], the ISIC 2019 dataset comes from different hospital sources: HAM10000 [28], BCN 20000 [10], and MSK [9].

The original training dataset is divided into the training set (80%) and the validation set (20%). Table I summarizes the number of images and metadata records for each of the training, validation, and test sets, split by all the eight different classes.

TABLE I: The total number of samples in training, validation and test sets, and the number of samples per class.

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.

B. Evaluation Metrics

In order to compare the results, the main metrics used were the Sensibility (SE) and the Balanced Accuracy ($BACC$). 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:

$$SE_i = \frac{TP_i}{TP_i + FN_i}, \tag{4}$$

where TP_i is the True Positive of the class i (it is predicted class i and it is true), and FN_i is the False Negative of the class i (it is predicted negative, and it is false - it belongs to class i).

Regarding $BACC$, since the dataset is unbalanced, instead of using the weighted accuracy, this metric is used. Thus, the same importance is given to all classes, independently of the number of examples. $BACC$ is the average of the SE obtained for each class. In this case, it is given by:

$$BACC = \frac{\sum_{i=0}^{7} SE_i}{8}. \tag{5}$$

C. 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 with Adam Optimizer algorithm.
- 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 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.

Classification with dermoscopic images only

The experiments without metadata were performed using ResNet 101 and Xception architectures. 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. Table II represents the results obtained with ResNet and Xception architectures.

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.

TABLE II: $BACC$ in the experiments without metadata.

Architecture	Validation Set [%]	Test Set [%]
ResNet	74.62	52.22
Xception	75.56	50.52

Classification with metadata only

In this case, 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.

In the validation set, the $BACC$ obtained is 34.41%. 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.

Classification with images and metadata

Experiments with different fusion methods were carried out. For each method, ResNet 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. Regarding Method 5, the best value of the hyperparameter α was 0.2. Table III shows the comparison between the methods that combine images with metadata and the methods without metadata.

TABLE III: comparison between the methods that combine images with metadata, and the methods without metadata, based on $BACC$.

	ResNet		Xception	
Method	Val. Set [%]	Test Set [%]	Val. Set [%]	Test Set [%]
No metadata	74.62	52.22	75.56	50.52
Method 1	78.18	55.07	78.18	55.50
Method 2	77.76	56.01	79.65	54.79
Method 3	78.00	54.30	78.49	53.39
Method 4	77.79	54.13	78.22	54.84
Method 5	78.03	54.85	78.97	51.26

All the methods that combine images with metadata lead to improvements in the $BACC$ scores, both for the validation and test sets. 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.

In order to see the improvements brought by each method, which fuses images and metadata, in relation to the network without metadata, the SE of each lesion was analyzed. In this analysis, 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. This analysis was performed on the validation and the testing set, for both ResNet and Xception architectures. As an example, table IV represents the results obtained for ResNet in the validation set.

TABLE IV: Comparison of metrics between all methods that combine images with metadata, and the model with only images as input, in the validation set with the ResNet architecture.

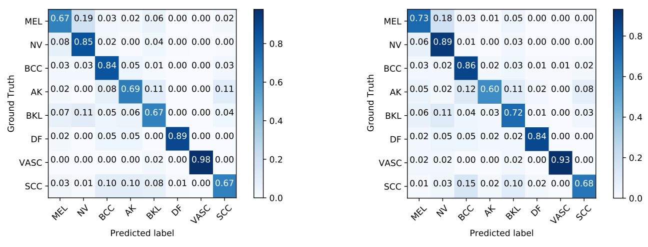
	SE								
Model	MEL	NV	BCC	AK	BKL	DF	VAS	SCC	$BACC$
No metadata	68.43	78.28	81.40	65.09	66.11	72.72	97.72	67.21	74.62
Method 1	67.17	84.66	84.34	68.64	67.04	88.64	97.73	67.21	78.18
Improvements	-1.26	+6.38	+2.94	+3.55	+0.93	+15.92	+0.01	0.00	+3.56
Method 2	68.78	87.32	83.26	72.19	70.00	84.09	90.91	65.57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

Regarding ResNet architecture, the bigger SE improvements happen in DF and NV lesions. In the method 3, DF improved 22.73% (it has reached a SE = 95.45%). The class SCC got worse with the introduction of metadata. As $VASC$ has a great SE in the classification with images only in the validation set, just method 1 led to minimal improvements in the SE of this class. All the other methods made it worse.

In the case of Xception (the respective table is not presented here), 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 test set, AK and $VASC$ do.

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 biggest 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 depend on the method used to incorporate the metadata.

Another important observation was drawn based on the confusion matrices of all the methods. As an example, fig. 7 represents the confusion matrices of method 1, for ResNet and Xception, in the validation set.



(a) ResNet

(b) Xception

Fig. 7: Confusion matrices to method 1 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.

V. Effect of each type of metadata feature

In order to study the influence of each combination of metadata features, 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 only gender is taken into account, the size will be 2. The remaining training conditions were the same as those used with all features. Table V summarizes the $BACC$ obtained when method 2 was trained with all the combinations of features, for ResNet architecture.

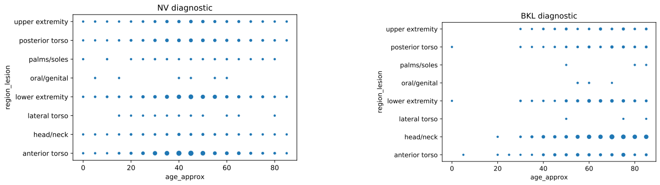
TABLE V: $BACC$ of the Model 2 with all metadata combinations as input, for ResNet architecture.

Features	Val. Set [%]	Test Set [%]
No metadata	74.62	52.22
Age	76.28	53.46
Gender	74.71	51.71
Region	76.15	51.21
Age + Gender	77.03	51.82
Age + Region	76.46	53.67
Gender + Region	76.10	51.40
Age + Gender + Region	77.76	56.01

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 (Xception's results are not depicted here). The combination that led to the best result, in the validation and test sets, is the one that combines all the metadata features: age, gender and body region. This was observed for both CNN architectures.

The other goal of this section is to analyze the metadata features, and their relationships with each lesion, in order to better understand the potential influence of the patient's information on the classification of skin lesions. The improvements obtained when metadata is taken into consideration may be related to some hypotheses defined. The networks may take advantage of some relationships in metadata to improve the distinction of some classes. The hypotheses were defined based on the validation set. As an example, in fig. 8 the markers represent the two-dimensional distribution of occurrences according to the patient's age and the body region of the lesion, for NV and BKL.

As can be seen, these two lesions appear more frequently in specific body regions. NV is more frequent in the anterior torso and BKL in head/neck. Taking into account not only the region of the lesion but also the patient's age, it may be



(a) NV

(b) BKL

Fig. 8: 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.

observed that the preferences for some regions of the body can be restricted to some age ranges. In NV the most frequent region is the anterior torso between 30 and 55 years old, and the lower extremity between 30 and 55 years old. In BKL the prevalence to head/neck is higher for ages over 55 years old. Table VI shows the SE of these lesions in the validation set, without metadata and with the combination of metadata Age and Region of the body, for ResNet architecture.

TABLE VI: SE of the lesions NV and BKL in the validation set, without metadata and with the combination Age and Region of the body, for ResNet architecture.

Lesion	No metadata	Age + Region
NV	78.28	84.70
BKL	66.11	72.41

As the SE of these two classes have improved considerably in the case of the ResNet architecture when just Age and the region of the body were considered, this seems to support the hypotheses. The ResNet network might be learned these relationships, and taken into account in the classification, leading to significant improvements on the SE of these lesions.

VI. Web Site Application

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. To created 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.

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 (Method 1 with Xception was chosen because of the memory). After receiving an image and the patient's information, the server feeds the input into the model, performs the prediction, and returns the output of the $Softmax$ and the diagnostic to the client, that displays the result. A complete example of how the website application works is available on: https://www.youtube.com/watch?v=cwCnXPRWa1o

VII. 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 test 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 test 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 test set. It led to an improvement of 3.14% and 3.79% in the validation and the test 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 most significant 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 some hypotheses proposed. These hypotheses say that some combinations of metadata may be helpful to improve the SE of specific 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 body region (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.

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. A complete example of how the website application works is available on:
https://www.youtube.com/watch?v=cwCnXPRWa1o

A. 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.

- 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 that allows automated retraining, in order for the dermatologists add samples, and the system automatically retrains the model.

References

[1] Training Data — ISIC 2019.

[2] Developer guides, 2020.

[3] Home - The Skin Cancer Foundation - Skin Cancer Facts & Statistics. 2020.

[4] Ultraviolet (UV) Radiation and Skin Cancer, 2020.

[5] C. Barata, M. E. Celebi, and J. S. Marques. Improving Dermoscopy Image Classification Using Color Constancy. *IEEE Journal of Biomedical and Health Informatics*, 19(3):1146–1152, 5 2015.

[6] 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*, 23:1096–1109, 2018.

[7] 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.

[8] F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

[9] 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*, pages 168–172. IEEE Computer Society, 5 2018.

[10] 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.

[11] J. M. Ede and R. Beanland. Adaptive Learning Rate Clipping Stabilizes Learning. *Machine Learning: Science and Technology*, 1(015011), 6 2019.

[12] 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*, 542(7639):115–118, 2017.

[13] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer. Skin Lesion Classification Using Ensembles of Multi-Resolution EfficientNets with Meta Data. *MethodsX*, 7(100864), 2019.

[14] S. Goes. Introduction to Convolutional Neural Networks. In *Learning the Scale of Image Features in Convolutional Neural Networks*, chapter 3, pages 23–40. 2017.

[15] 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)*, pages 770–778, 2016.

[16] N. Heller, E. Bussman, A. Shah, J. Dean, and N. Papanikolopoulos. Computer Aided Diagnosis of Skin Lesions from Morphological Features. Technical report, 2018.

[17] 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*, pages 4700–4708, 2017.

[18] 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*, 8(1):1–207, 2 2018.

[19] 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.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Neural Information Processing Systems*, pages 1097–1105, 2012.

[21] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. In *Nature*, volume 521, pages 436–444. Nature Publishing Group, 5 2015.

[22] 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)*, pages 1996–2000, 2020.

[23] S. Nofallah and W. Wu. Transfer Learning for Automatic Disease Diagnosis with Dermoscopic Images. Technical report, 2018.

[24] 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.

[25] W. Rawat and Z. Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9):2352–2449, 2017.

[26] 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.

[27] 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*, pages 1–9, 2015.

[28] 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*, 5(180161), 3 2018.

[29] 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*, 153(1):23–29, 1 2017.