IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS

The final published version is available with DOI: 10.1109/JBHI.2021.3113609

A Cloud Approach for Melanoma Detection based on Deep Learning Networks

Luigi Di Biasi, Alessia Auriemma Citarella,
Michele Risi, *Member, IEEE*, Genoveffa Tortora, *Senior Member, IEEE*

***Abstract*— In the era of digitized images, the goal is to extract information from them and create new knowledge thanks to Computer Vision techniques, Machine Learning and Deep Learning. This enables the use of images for early diagnosis and subsequent treatment of a wide range of diseases. In the dermatological field, deep neural networks are used to distinguish between melanoma and non-melanoma images. In this paper, we have underlined two essential points of melanoma detection research. The first aspect considered is how even a simple modification of the parameters in the dataset determines a change of the accuracy of classifiers. In this case, we investigated the Transfer Learning issues. Following the results of this first analysis, we suggest that continuous training-test iterations are needed to provide robust prediction models. The second point is the need to have a more flexible system architecture that can handle changes in the training datasets. In this context, we proposed the development and implementation of a hybrid architecture based on Cloud, Fog and Edge Computing to provide a Melanoma Detection service based on clinical and dermoscopic images. At the same time, this architecture must deal with the amount of data to be analyzed by reducing the running time of the continuous retrain. This fact has been highlighted with experiments carried out on a single machine and different distribution systems, showing how a distributed approach guarantees output achievement in a much more sufficient time.**

***Index Terms*— Skin Cancer, Melanoma Classification, Deep Learning, Distributed Computing, Smart Healthcare**

I. Introduction

Melanoma is an aggressive form of skin cancer that originates from melanocytes, the cells of the epidermis responsible for production of melanin pigment. Although this kind of tumor represents a minority of cutaneous malignancies, it is the main cause of mortality [1]. Melanoma of the skin has increased dramatically over the last 30 years, but trends vary by age group. Between 2007 and 2016, the rate for people under 50 years old fell by 1.2% while the rate for those 50 and over increased by 2.2% each year. According to the American Cancer Society, 100350 new cases and 6850 deaths in both sexes were estimated in 2020, only in the United States[1].

It is well known in the literature that early diagnosis of melanoma remains an open challenge [2]. The attestation of a correct diagnosis also depends on the physician's ability related to his degree of experience in discriminating between different types of skin lesions. The final word on a bad diagnosis is still demanded to the biopsy.

The importance of an early diagnosis of melanoma has become increasingly evident, especially in subjects who have a high risk of developing cancer, which allows an increase in the cure rate. Generally, in clinical practice, a first visual inspection by a dermatologist is used to diagnose melanoma, often with the assistance of polarized light magnification dermoscopy [3].

Technology can dramatically change the conception of medicine and, at the same time, it plays a critical role in advanced diagnostics systems in making decisions intrinsic to patient care [4]. However, the fundamental aspects of the decision-making process must not be lost sight of: the answer to the clinical question is inseparable from medical research and, therefore, from the doctors' experience. Only in a collaborative context, between technological and medical actors, the quality of the final result is guaranteed.

Many types of computer software have been developed, in the last years, with the aims to help dermatologists to better (and fast) understand if a skin lesion is, is not or could become a melanoma [1]. Nowadays, many proposals exist for a computer-aided system for dermatologists [5] [6], but despite the claims related to the great performance of AI in surpassing clinicians' performance, there are many other challenges still open. Most of this software lies on computer vision-related techniques like border detection, symmetry/asymmetry analysis, color analysis and dimension detection [5]. Some software also uses other types of information like Electronic Health Records (EHR) to improve the prediction accuracy. Overall, current melanoma detection systems must take into account the complexity of the images to be processed, which might cause issues such as irregular fuzzy boundaries of lesions, noise and artifact presence, low contrast, or poor image lighting [7].

Luigi Di Biasi is with the Computer Science Department of University of Salerno, Fisciano, SA 84084 IT (e-mail: ldibiasi@unisa.it).

Alessia Auriemma Citarella is with the Computer Science Department of University of Salerno, Fisciano, SA 84084 IT (e-mail: aauriemmacitarella@unisa.it).

Michele Risi is with the Computer Science Department of University of Salerno, Fisciano, SA 84084 IT (e-mail: mrisi@unisa.it).

Genoveffa Tortora is with the Computer Science Department of University of Salerno, Fisciano, SA 84084 IT (e-mail: tortora@unisa.it).

"This work was supported in part by Softmatter Lab - DIFARMA"

[1] https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2020/cancer-facts-and-figures-2020.pdf

The way these systems are built can be summarized as follows. At first, an image dataset containing melanoma and non-melanoma images is downloaded or created. Dermoscopic, clinical and histological images are available: the first ones are more detailed but the need for a dermoscopy could limit the size of the dataset; the second ones are less detailed but it is more readily available; the third ones are related to the highest images resolution. Then, one or more Computer Vision and image processing techniques are applied to extract features from the images. These features will be used as training inputs.

In order to train a chosen model, it is possible to use neural networks, SVMs, custom predictors, classifiers or other kinds of Machine/Deep Learning techniques. Generally, one or more techniques could be used together to improve model accuracy. Finally, a validation step and a test step are performed in order to measure the model performances.

Unfortunately, at least two big drawbacks might be found in the previous workflow. The first one is related to the storage space and the heavy computation needed to train a complex model on large datasets to achieve good performance. As an example, the pure segmentation method based on k-means clustering could require a Running Time (RT) of $O(2^n)$ to converge [8]. The time complexity of the simplest neural network (using no optimization) training is strictly related to the RT of the matrix multiplication algorithm, that is, O(N^d) where d is the dimension of the matrix (in square matrix case). The RT increases when weight optimization models are introduced. As a brief example, if we consider a one-forward pass, we have to introduce another iteration to the training, which means our d term becomes d+1; introducing a more sophisticated weight computation approach means that our RT could explode very soon. More approximation and optimization techniques exist to cut the training time, but it is clear that, when input grows, we need more computation power and more optimized hardware (like GPUs) to have the results in human time. The second drawback is related to the required efforts to keep one or more models updated. Once a model is trained and deployed, there are no simple ways to update and improve its performances without repeating the training step. It is notable what happened in ISIC 2019 Challenge: the ISIC 2018 winning algorithm performance drop from 88.5% to 63.6% in the ISIC 2019 only due to the introduction of more categories and images [5]. Too near to a flipping coin decision base technique. The authors in [5] suggested three reasons related to this fact:

- deep learning mostly depends on the quality of the image and the training dataset structure (balanced/unbalanced);
- intra-class dissimilarities and inter-class similarities can affect system performances;
- deep learning requires that the input undergo a data augmentation process to learn how to discriminate the same object from different points of view (stretched, rotated, illuminated, and so forth).

Often, the three datasets (training, validation, test) are fixed before network training. This fact might be hidden that a slight change in the subsets could impact prediction accuracy. This fact suggests that Transfer Learning, the utilization of a pre-trained network for other problems, is still not reliable.

Generally, Transfer Learning is a practice that allows using pre-trained networks and reduces training times when using previously acquired knowledge to solve a similar problem. In this way, we focus only on training the last layers engaged in the classification process. From a Machine Learning perspective, this transfer assumes that the original data and target data are in the same feature space and have the same distribution [9].

If a new image dataset, more accurate and more refined, is created, there might be the need to go back to the laboratory, train the model again, test it and deploy it again. Also, if the new datasets are large, it is necessary to use High-Performance Computing (HPC) to compute the result in good time.

Due to the ISIC 2019 results reported in [5], in this work, we have assumed that the training step must be reiterated many times (we chose to reiterate it 100 times) by randomly rebuilding the three datasets for each training iteration in order to show how accuracy changes following the dataset changes. This assumption will demonstrate the importance of using a distributed computing approach for heavier computational operations, the same needed for continuous retraining.

The main contributions of this work include the following:

- the proposal of a hybrid architecture for the classification of melanoma. We show how a simple dataset modification can impact the classifier performance and that a distributed and cooperative system is needed to enable deploying a melanoma classifier usable into the real world;
- a novel perspective of evaluating the quality of melanoma categorization by comparing clinical images with and without segmentation.

The paper is organized as follows. In Section II, we show a general overview of the distributed architectures proposed in the context of Internet of Medical Things (IoMT) and the main methods of melanoma classification. Section III describes the methods with which the experiment was conducted, with particular reference to the scenarios of the use of a single machine and of a distributed approach, respectively. Furthermore, the impact of the dataset structure on the accuracy of the models (Section III-C) and the obtained results are highlighted in Section III-D. Finally, Section IV and Section IV-A summarize the conclusions of the paper with a view to future research that should allow the switch from the current Machine Learning approach to a fully Deep Learning approach that would not need any.

II. Background and related works

Over the past few years, the importance of having tools available to assist dermatologists in diagnosing melanoma has grown considerably.

Many methods for automatic detection and classification of melanoma have been developed, ranging from decision trees [10], to Support Vector Machines (SVM) [11], to logistic regression [12] and Bayesian classifiers [13].

Convolutional Neural Networks (CNNs) play an essential role in the image-based detection of a pathology [14]. Their

use is widely demonstrated in the detection [15], segmentation and subsequent classification of melanocytic lesions [16].

The study of Haenssle *et al.* [17] reports a comparison between the performance of dermatologists and a marked approved convolutional neural network in order to detect skin lesion. Specifically, dermatologists were classified according to their level of experience in dermoscopy (beginner, less than two years of experience; skilled, between 2 and 5 years of experience; expert, more than five years of experience) and could access two levels of information: dermoscopic image (level I) and dermoscopic image with clinical close-up image and textual information (level II). The aim was to indicate the outcome of the diagnosis (malignant, benign or premalignant lesion) and their management decision (treatment/excision, no action, follow-up examination). When dermatologists observed only the first level of information, they reached a mean sensitivity and a lesion classification specificity of 83.8% and 77.6%, respectively. With the addition of the secondary level of information, sensitivity increased to 90.6% while specificity reached 82.4%. In conclusion, the study showed that in the presence of a wider spectrum of diagnoses, dermatologists and convolutional neural networks have achieved similar results. In addition, dermatologists have shown their ability to integrate information from multiple sources in order to make an appropriate diagnosis. This evidence justifies the explosion of research on melanoma detection with the help of Machine/Deep Learning methods. In order to support the classification of melanoma, a variety of neural networks have been tested, some of which are given below.

In 2014, Simoyan [18] highlighted how the architecture of the deeper Visual Geometry Group model (VGG), based on the learning of models with a greater number of image descriptors used as inputs (such as color, symmetry, contour, etc), can guarantee a better efficacy in melanoma detection. Furthermore, depending on the blocks and the used filter, the VGGs can be applied to the search field in question. The most common models are VGG 11, 16 and 19, which differ from each other in the number of convolutional layers: 8, 13 and 16, respectively.

In [19] the use of AlexNet, a convolutional neural network with 8 layers, is presented: the first five layers were convolutional layers, some of them followed by max-pooling layers, and the last three were fully connected layers. It used the non-saturating ReLU activation function, which showed improved training performance over *tanh* and *sigmoid*. Authors reported that by using Transfer Learning and the data augmentation method, testing and verifying it on the three MED-NODE, Derm (IS-Quest) and ISIC datasets, the achieved accuracy of the network with these parameters was 96.86%, 97.70%, and 95.91%, respectively.

GoogleNet [20] is a convolutional neural network composed of about 100 building blocks of various types, in particular convolutions, average pooling, max pooling and contacts, arranged on 27 layers in deep. This network relies on the primary Inception architecture, which already appeared in 2015 as a computationally efficient network even in limited computational resources. GoogleNet executions on Cloud TPU are available via Google Cloud Platforms.

Google InceptionV3 [20] relies on the evolution of the Inception Architecture. It is a widely-used image recognition model that has been shown to reach greater than 75% accuracy on the ImageNet datase [21]. Google InceptionV3 is composed of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concerts, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.

Esteva *et al.* [1] have evaluated the use of the Google InceptionV3 network in skin cancer classification against the expertise of 21 dermatologists, demonstrating how the network was able to efficiently compare with specialists in this task.

At the same time, with the advancement of technologies, there has been an exponential increase in intelligent devices connected to the Internet capable of generating massive volumes of data. This parallelism may also be evident in the dermatological field, with the prospect of using simple devices such as smartphones to capture clinical images as well as as sensors for remote skin anomaly screening [14]. This last point highlights both the promise of improving melanoma screening by assisting in the real-time mapping of nevi, as well as the requirement for a more flexible and faster system architecture for image processing, storage, and management.

In 2017, the first architectures composed of Edge, Fog and Cloud resources appeared in the context of the IoMT and which support anticipatory learning [22]. Most of the management and analysis approaches present in the literature regarding IoMT data are based on Cloud computing. The fundamental problems that remain unsolved in this sector are related to improving individual user data security, resolution in the exchange of medical images, data archiving, and the ability to improve diagnosis response times by decentralizing computing power for Machine and Deep learning techniques on network nodes, which are used as microdata center mesh network [23].

The architecture proposed by a recent article allows modeling solutions for the classification of lung and skin diseases but limiting itself to testing the system in terms of IoMT data security, presenting the possibility of offering flexibility in the adoption and combination of artificial intelligence techniques [24].

There is still limited information on hybrid architectures of three layers which allow specific computational operations for melanoma images and build a database processed in real-time starting with access to a more user-friendly instrumentation, such as a smartphone.

III. METHODS

In order to show how the two drawbacks could impact classifier performances and the design of an AI-based detection systems, we performed two experiments.

This work did not focus on the features extraction step (like segmentation, border analisys and others) thanks to the high availability of works in this research field (e.g., [25] [26] [27] [28]). Consequently, the simplest Otsu segmentation and Gaussian filter are used into segmentation/preprocessing step.

For the performed experiments, we considered the following melanoma detection process workflow, using only the most computationally expensive step:

1) a pre-processing phase with the removal of all artifacts from the images;
2) a segmentation phase of the lesion in order to separate the melanoma from its background;
3) a post-processing phase to further improve the image's quality;
4) a phase of extraction of clinical characteristics for the recognition of melanoma based on dermatological guidelines and Computer Vision techniques;
5) classification phase with a validation step and a test step performed to measure the model performances.

To this aim, we propose a hybrid architecture divided into:

- *Cloud layer*: in which high-performance computing techniques and data storage are carried out. When new images become available, new system training could be performed in the Cloud, periodically. The validation and test steps are also completed in the Cloud. The Cloud layer will deliver a new network to the edges when a new classifier is available.
- *Fog layer*: it consists of network-distributed server systems which receive data from the Edge layer, pre-process, filter, and upload it to the Cloud. At this level, mid-weight computational methods can also be conducted. Finally, for devices with insufficient processing capacity, the Fog layer may use the same algorithm as the Edge layer.
- *Edge layer*: it consists of all intelligent devices (i.e., Edge Devices) of the IoT architecture. At this level, the data are processed by the Edge Device (smartphone). Into the Edge, the latest classifier is executed in order to analyze the images.

A. The MED-NODE dataset

The dataset used is the same used in realizing the MED-NODE computer-assisted system for melanoma diagnosis [29]. In the following, we refer to it with the name of the MED-NODE dataset. It consists of 170 clinical images (70 melanoma and 100 nevi images) from the digital image archive of the Department of Dermatology of the University Medical Center Groningen (UMCG). These are clinical images obtained with Nikon D3 or Nikon D1x body and a Nikkor 2.8/105 mm microlens, whose average distance between the lens and the lesion is about 33 cm in 95% of the images in the dataset. Dermatologists have verified each image to label them correctly. The images come from different patients of Caucasian origin, have been made anonymous and already pre-processed. Hair removal has already been done previously with the Dullrazor software [30].

B. Transfer Learning Reliability Evaluation

In the first experiment, our goal is to show how the datasets structure modifications could cause a drop in the overall system performance. In particular, we aim to demonstrate that Transfer Learning is not fully reliable. In order to validate our hypothesis, we performed continuous retraining of three classifiers, varying the dataset's structure slightly. In this scenario, we identify the training, validation, and test set compositions with datasets structure term.

Due to our assumption regarding the Transfer Learning realibility [5] we have generated four new datasets (MDS) as described following.

1) MD1 - contains MED-NODE original images;
2) MD2 - contains MED-NODE images segmented with the Otsu method;
3) MD3 - contains MED-NODE images and augmented images without segmentation;
4) MD4 - contains MED-NODE images and augmented images segmented with the Otsu method.

The basic hypothesis is that the source and destination domain data may differ in terms of the marginal distribution, considering the four datasets generated but that the reference labels are always the same. The Otsu method was used for the segmentation process, which can minimize intra-class variance [31].

Otsu segmentation was performed using the following code:

```
[input_image,map] = imread(F);
bw_input = rgb2gray(input_image);
[T, EM] = graythresh(bw_input);
BW = imbinarize(bw_input, T);
mask_otsu = BW;
mask_otsu= ~mask_otsu;
new_image = input_image * mask_otsu;
```

An example of the Otsu's output is depicted in Figure 1.

Since the dataset used is made up of only 170 total images (70 malignant and 100 benign), we opted for the data augmentation technique to obtain additional variations through artificial transformations of the images [32]. The choice of the considered transformations was originated from a careful inspection of the characteristics of the test set. Data augmentation was performed by using Matlab `imageDataAugmenter` object, with the following configuration:

```
'RandRotation', [-180, 180], ...
'RandScale', [1, 10], ...
'RandXTranslation', [-180, 180], ...
'RandYTranslation', [-180, 180]
```

Fig. 1. Image with the application of the Otsu segmentation.

We used this experiment to collect classifier performance data and evaluate the effort (expressed in time) needed to execute the retrain without using the distributed approach. The system training pipeline is reported in Figure 3. A single Intel Scientific Workstation with 16 core, 16GB RAM, and one GPU GTX980 was used in this setup.

We downloaded the Google InceptionV3 [20], GoogleNet [20] and AlexNet [19], whose architectures are publicly available. These networks were pre-trained to work with many image classes: We used these networks for the high scores reported for accuracy. We adapted the final layers to classify between 2 classes instead of 1000. This step involves replacing the SoftLayer and ClassificationLayer from the original network with a new layer with two output classes (i.e., melanoma/non-melanoma).

We simulated the continuous retraining by repeating the training step 700 times for each dataset D in MDS. For each iteration, the dataset was split using the following ratios: 0.5 training set, 0.3 for validation set and 0.2 for the test set. We used the splitEachLabel function with randomized option active. Then, for each iteration, the training set contained 50% of the melanoma images and 50% of non-melanoma images, picked randomly from the initial image set. The validation and test sets contained used the same technique with different ratios.

Before each network training, a Gaussian filter application was performed to reduce noise: the imgaussfilt function was used with a dynamic sigma value between 1 and 7 [33]. The Figures 2(a) and 2(b) show the two Gaussian filter application results with different sigma values. We performed 100 training iterations for each sigma value.

(a) Sigma value set to one.

(b) Sigma value set to seven.

Fig. 2. Gaussian filter with sigma value set to one (a), and to seven (b).

The training options were the following, with a 30 epochs, a N initial learning rate of 10^{-4} and the *stochastic gradient descent with momentum* (SGDM) [34]:

```
('sgdm', ...
'MaxEpochs', 30, ...
'MiniBatchSize', 12, ...
'Shuffle', 'every-epoch', ...
'InitialLearnRate', 0.0001, ...
'Verbose', true, ...
'ValidationData', imdsValidation, ...
'ValidationFrequency', 1, ...
'VerboseFrequency', 1, ...
'Plots', 'training-progress',
    'ExecutionEnvironment', 'gpu')
```

This experiment involved the execution of 8400 training steps (each with 30 epochs) on a single workstation environment. The obtained results suggest that the mean accuracy of every model is rarely near to the best accuracy. On the contrary, we observed the same behavior reported for the ISIC 2019 challenge. The details are reported and discussed in Section III-D.

C. Impact of the three-layers architecture

In the second experiment, we collected the architecture's performance. We explicitly design the architecture to allow automatic classifier retraining and deploying to show that a distributed and cooperative system is needed to deploy a melanoma classifier robust against Transfer Learning issues. In particular, due to the high reiteration needed to detect the best classifier if data structure changes happen. In this setup, the user can query the system and take part in dataset and model improvements thanks to the Fog layer, which stores and delivers each image to the Cloud. In this scenario, the data scientists must only annotate new images with a label (i.e., melanoma/non-melanoma).

We simulated a three-layers architecture in which the training and retraining steps are performed in the Cloud layer (see Figure 4). This setup was built with the GRIMD framework [35] which has allowed us to distribute each iteration upon multiple instances. First, we have deployed the GRIMD instances on Amazon AWS. Then, the training, retraining, validation, test and performance comparison steps were moved into Cloud layer. The underlying idea is that every time a new model is ready, it is deployed into Fog if and only if its accuracy outperforms the previous one. The synchronization between each layer is managed by the Layer Agents, which we implemented as a simple CROND instance. More in detail, we isolated the Cloud layer that is fully independent of the classification problem. Then, the Agent layer was configured to deploy a new trainer classifier to the Fog layer if and only if the new classifier median accuracy outperforms the previous one. Then, the training of the classifiers is isolated from the execution for prediction purposes. Finally, the classification

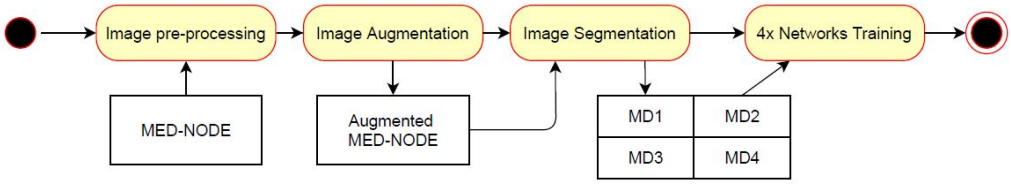


Fig. 3. The sequential pipeline used in experiment one for performing the continuous retraining.

and prediction steps were moved to Fog layer, where the web server and trained models are also stored. In this scenario, every end-user uses an app that communicates with the Fog layer.

This second experiment involves the same computations as the first one, but it allowed GRIMD to scale up to 128GB and multiple GPU, using Ec2 instance from type t2 (micro-instances: t2.micro) and m5 (balanced computation instances: m5a.2xlarge), to type c6 (optimized computation instances: c6g.16xlarge). Due to the presence of the t-instances and c-instances (computationally optimized instance), we configured the training session as follows:

```
('sgdm', ...
'MaxEpochs', 30, ...
'MiniBatchSize', 12, ...
'Shuffle', 'every-epoch', ...
'InitialLearnRate', 0.0001, ...i
'Verbose', false, ...
'ValidationData', imdsValidation, ...
'ValidationFrequency', 1, ...
'VerboseFrequency', 1, ...
'Plots', 'none',
    'ExecutionEnvironment', 'auto')
```

The execution environment parameter set to auto allowed Matlab to automatically detect and use CPU (and GPU if available) on the different machine instances. The obtained results are described in the following section.

D. Results

In order to estimate the performance of the three networks, we adopted the RMSE metric as default accuracy (denoted with ACC). The RMSE formula is reported in Equation 1: it indicates how far the observed data values differ from the estimated values [36].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{1}$$

where n is the size of the dataset, y_i represents the i_{th} observation, that is the value assumed in the real dataset and $\hat{y_i}$ the value predicted by the network on y_i. In addition, sensitivity (TPR), specificity (TNR), precision (PPV), false discovery rate (FDR), false-negative rate (FNR) and false-positive rate (FPR) were calculated and have been shown graphically in Figures 5(a)-5(c) with Otsu segmentation and in Figures 6(a)-6(c) without Otsu. The chosen metrics are described in the equations below (Equations 2-7):

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

$$TNR = \frac{TN}{TN + FP} \tag{3}$$

$$PPV = \frac{TP}{TP + FP} \tag{4}$$

$$FDR = \frac{FP}{FP + TP} \tag{5}$$

$$FNR = \frac{FN}{FN + TP} \tag{6}$$

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

where TP, TN, FP are the numbers of true positives and true negatives correctly predicted and FP and FN are the numbers of false positives and false negatives erroneously predicted, respectively. We took into account also the standard deviation (SD) to determine, on average, how much the accuracy measures differ from each other, according to the formula in Equation 8.

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{8}$$

where n is the size of the dataset and $\overline{x}$ is $\frac{1}{n}\sum_{i=i}^{n} x_i$ the arithmetic mean of x.

In Figures 7(a)-7(c) are reported the SD values for all three networks.

In Table I and Table II we have reported the results obtained on the MED-NODE dataset by performing the analysis with and without Otsu segmentation. In bold we have highlighted the highest values reached by the networks in the calculations of the average, maximum, minimum and standard deviation values of the ACC. The best result for the average ACC is obtained for the AlexNet network without applying data augmentation and with and without the applied segmentation (highlighted in red).

We evaluated the performance of the networks analyzing their behaviors for each dataset. The results, reported in Table III, suggest that GoogleNet is the most robust network with a mean drop of -19.60% in prediction accuracy. Also, the results seem to confirm what happened in the ISIC 2019

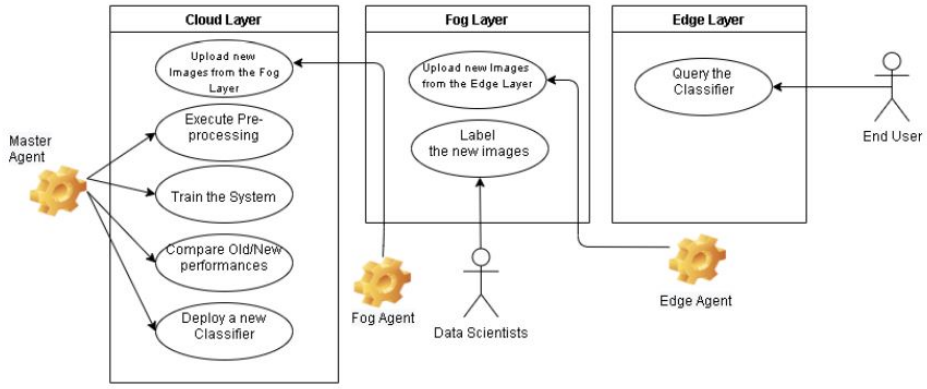


Fig. 4. The setup of the second experiment simulates a three layers architecture.

WITH OTSU SEGMENTATION					
Net	**Data Augmentation**	**ACC (min)**	**ACC (max)**	**ACC (mean)**	**ACC (sd)**
AlexNet	None	**0.65**	**0.94**	**0.78**	0.06
	Yes	0.44	0.91	0.68	0.08
Google InceptionV3	None	**0.56**	**0.94**	**0.76**	0.07
	Yes	0.32	0.74	0.53	0.09
GoogleNet	None	**0.60**	**0.91**	**0.75**	0.07
	Yes	0.32	0.74	0.55	0.09

TABLE I
PERFORMANCES ON MED-NODE DATASET FOR ACCS WITH OTSU SEGMENTATION AND WITH AND WITHOUT DATA AUGMENTATION.

WITHOUT OTSU SEGMENTATION					
Net	**Data Augmentation**	**ACC (min)**	**ACC (max)**	**ACC (mean)**	**ACC (sd)**
AlexNet	None	0.68	**1**	**0.89**	0.05
	Yes	**0.76**	0.97	0.87	0.05
Google InceptionV3	None	**0.56**	**0.94**	**0.74**	0.07
	Yes	0.32	0.71	0.55	0.07
GoogleNet	None	**0.65**	**0.94**	**0.80**	0.06
	Yes	0.30	0.76	0.55	0.09

TABLE II
PERFORMANCES ON MED-NODE DATASET FOR ACCS WITHOUT OTSU SEGMENTATION AND WITH AND WITHOUT DATA AUGMENTATION.

Net	**Measure**	**MD1**	**MD2**	**MD3**	**MD4**	**Mean Drop**
AlexNet	Best	0.97	0.91	0.97	0.89	
	Average	0.81	0.72	0.81	0.73	
	Drop	-19.75	-26.38	-19.75	-21.91	-21.95
Google InceptionV3	Best	0.91	0.88	0.90	0.89	
	Average	0.75	0.72	0.75	0.74	
	Drop	-21.33	-22.22	-20.0	-20.27	-20.96
GoogleNet	Best	0.94	0.93	0.91	0.89	
	Average	0.81	0.77	0.75	0.74	
	Drop	-16.04	-20.77	-21.33	-20.27	**-19.60**

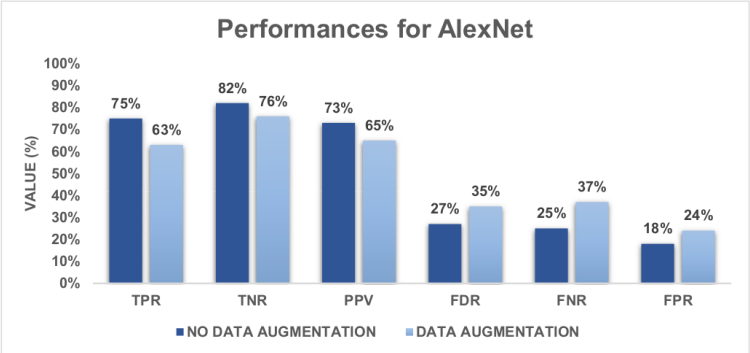
TABLE III
PERFORMANCE DROP AFTER 100 TRAINING STEPS (RELATED TO TRAINING AND VALIDATION STEPS).

challenge, where the ISIC 2018 winners drop up to 28% in performances.

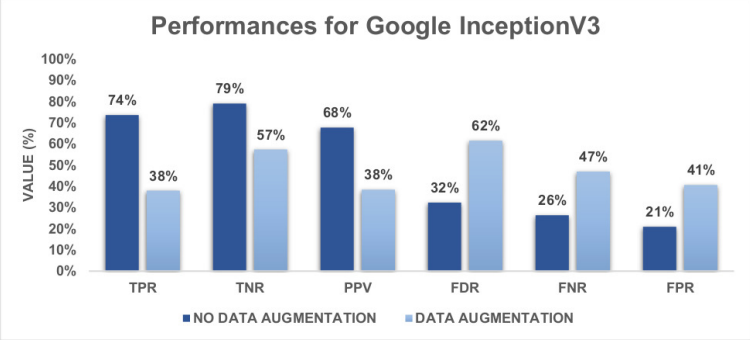
In Table IV, the clock time for both experiments is shown instead of the RT of the computation. In this case, we used clock time instead of RT since we were not interested in the (trivial) fact that the presence of a distributed architecture can speed up the training of networks, but we wanted to estimate the amount of time saved by data scientists under the two conditions of the experiments:

- in the condition of experiment 1, simulating the standard case which requires laboratory work to update the datasets, perform training, validation and deployment;
- in the condition of experience two provides for an architecture that takes care of all aspects except the annotation of the images (in any case delegated to expert dermatologists).

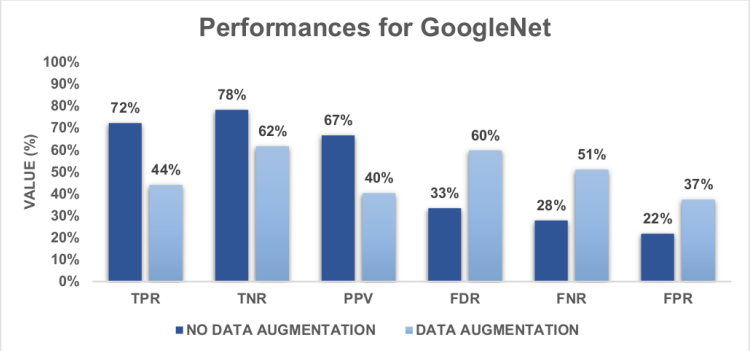
We collected the time effort needed to maintain a classifier to the top performance. In order to reach a good performance



(a) Performances with Otsu segmentation and with and without data augmentation for AlexNet.



(b) Performances without Otsu segmentation and with and without data augmentation for Google InceptionV3.



(c) Performances with Otsu segmentation and with and without data augmentation for GoogleNet.

Fig. 5. Performances for all used networks by applying Otsu segmentation.

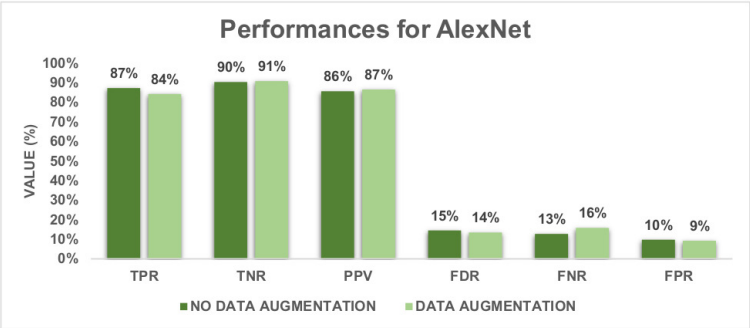
Environment	GoogleNet	Google InceptionV3	AlexNet
Single	82710	115200	19724
GRIMD(t2)	55140	94348	13327
GRIMD(m5)	20677	37105	6872
GRIMD(c6)	7519	17710	3171

TABLE IV
CLOCK TIME (IN SECONDS) MEASURED FOR BOTH THE EXPERIMENTS

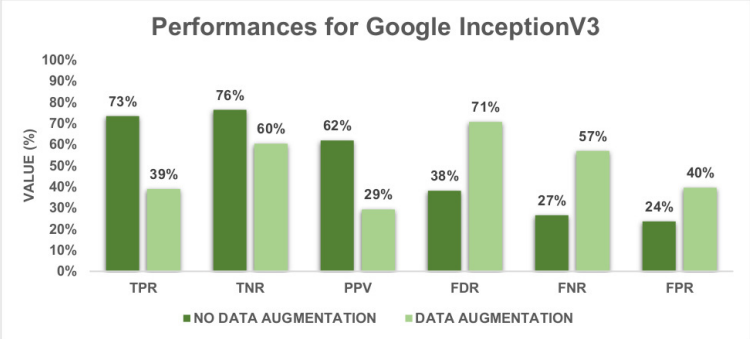
for MED-NODE (only 170 images) datasets, we spent up to 82000 seconds for each retraining. The last result suggests that a three layers architecture can enable Melanoma Detector's design, improving itself by improving the switch from Machine Learning to Deep Learning (unsupervised learning). This step should improve the decoupling between data scientists and model training. Also, using user-generated images can speed up new model deployment.

IV. CONCLUSIONS

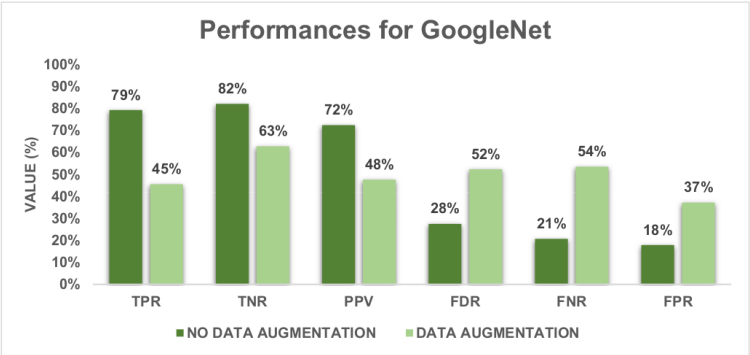
The results presented in this work suggest that the Transfer Learning approach, broadly adopted nowadays, might not be



(a) Performances without Otsu segmentation and with and without data augmentation for AlexNet.



(b) Performances without Otsu segmentation and with and without data augmentation for Google InceptionV3.



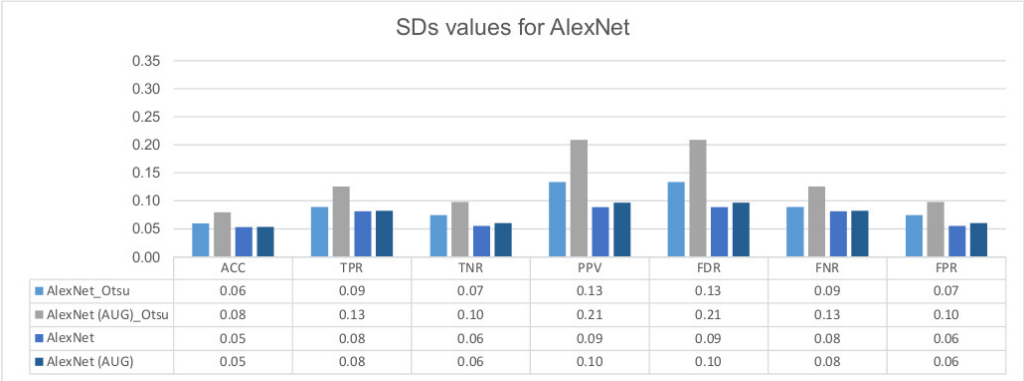
(c) Performances without Otsu segmentation and with and without data augmentation for GoogleNet.

Fig. 6. Performances for all used networks without Otsu segmentation.

reliable, despite the high performance reported in the literature. In particular, the first experiment's results show that a classifier performance could drop dramatically even for small changes to the original training dataset. These results agree with what recently happened in [5]. Our results also suggest that AlexNet is the most robust network regarding the Transfer Learning issue. Furthermore, all the used CNN networks have reached a better average ACC without segmentation and data augmentation.

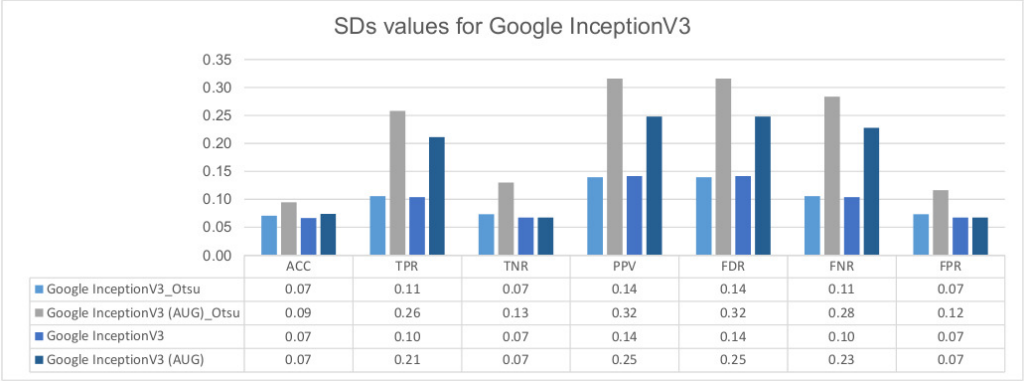
Due of the large number of training reiterations required to find the best classifier, continuous retraining is required to minimize performance drop. As a result of these findings, we performed the second experiment allowing continuous retraining thanks to a Cloud/Fog/Edge architecture. We were able to perform the continuous retraining step needed to maintain the classifier robust saving up to 76% of computational time. Consequently, we may infer that envisioning a distributed architecture could provide several benefits to the end user by allowing:

- the collection and aggregation of data "on the network" in order to support the early diagnosis of melanoma, enriching the image databases of new knowledge;
- processing critical data locally, at the network's Edge, with local data storage, resulting in reduced data processing latency, real-time response, lower bandwidth, and faster data access;
- widespread distribution of resources and computing services through a large number and mobility of Fog nodes and interoperability.



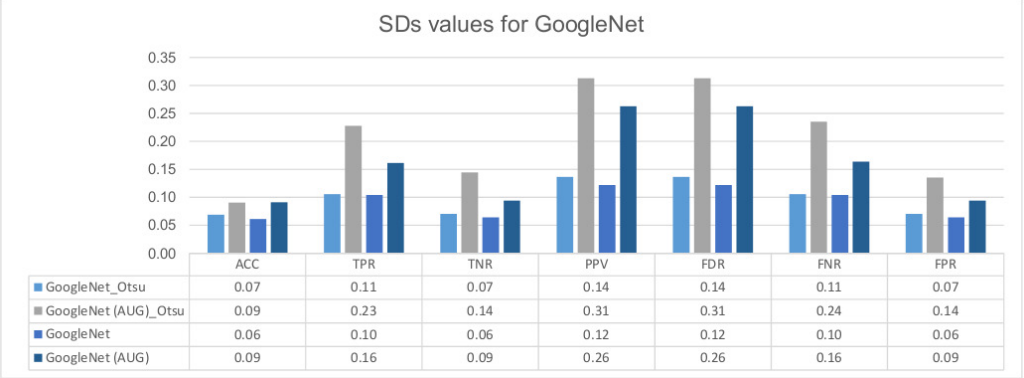
	ACC	TPR	TNR	PPV	FDR	FNR	FPR
AlexNet_Otsu	0.06	0.09	0.07	0.13	0.13	0.09	0.07
AlexNet (AUG)_Otsu	0.08	0.13	0.10	0.21	0.21	0.13	0.10
AlexNet	0.05	0.08	0.06	0.09	0.09	0.08	0.06
AlexNet (AUG)	0.05	0.08	0.06	0.10	0.10	0.08	0.06

(a) SDs values for AlexNet.



	ACC	TPR	TNR	PPV	FDR	FNR	FPR
Google InceptionV3_Otsu	0.07	0.11	0.07	0.14	0.14	0.11	0.07
Google InceptionV3 (AUG)_Otsu	0.09	0.26	0.13	0.32	0.32	0.28	0.12
Google InceptionV3	0.07	0.10	0.07	0.14	0.14	0.10	0.07
Google InceptionV3 (AUG)	0.07	0.21	0.07	0.25	0.25	0.23	0.07

(b) SDs values for Google InceptionV3.



	ACC	TPR	TNR	PPV	FDR	FNR	FPR
GoogleNet_Otsu	0.07	0.11	0.07	0.14	0.14	0.11	0.07
GoogleNet (AUG)_Otsu	0.09	0.23	0.14	0.31	0.31	0.24	0.14
GoogleNet	0.06	0.10	0.06	0.12	0.12	0.10	0.06
GoogleNet (AUG)	0.09	0.16	0.09	0.26	0.26	0.16	0.09

(c) SDs values calculated for GoogleNet.

Fig. 7. Several SDs values computed for all networks.

This type of architecture implementation responds to a novel demand and data management methods which are more advantageous than traditional methods, in the pre-processing and classification of melanoma images. In particular, it addresses the issue of sending images to a central data server or Cloud service for processing. Furthermore, by decentralizing them, the capacities and, as a result, the calculation times can be improved.

Figure 8 shows the general operation of the proposed hybrid architecture. Within the Cloud, data buckets are maintained and systems training is performed. The orchestrator takes care of the distribution of the optimized services after each formation in the Fog area, where services are executed. Local calculations on IoMT devices (e.g., smartphones) are performed in the Edge area. HiC-Otsu is a software component of the Fog system on the IoMT device, which performs a preliminary analysis of the loaded data. QoS moderator annotates content to improve system performance. The generic user exploits the output of services, but by loading data he contributes to the growth of the knowledge base of the system.

A. Future work

According to our results, as future work we plan to design more robust neural network models to better learn from the images (and generalize from them). The results we obtained show that CNN networks reported better performance when no

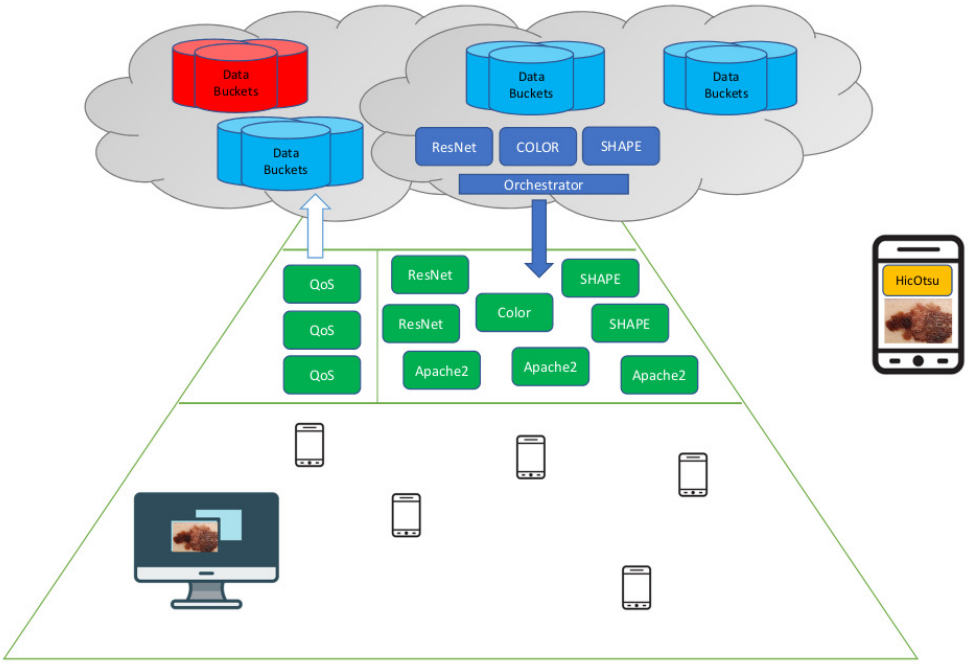


Fig. 8. General operation of the three layers architecture for melanoma detection.

segmentation was used. This finding could suggest that skin near lesions may contain important information to take into account during training. Experiment one should be performed again, analyzing other pre-processing techniques.

We were no able to estimate the minimum training step needed to reach good robustness. Future works should investigate this aspect to allow optimization in training time.

The second experiment helps to evaluate how a distributed environment might aid in time savings (both RT and clock time). A more in-depth investigation of the distributed architecture could be performed to understand if more complex architecture can improve performance.

REFERENCES

[1] 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*, vol. 542, no. 7639, pp. 115–118, 2017.

[2] N. R. Abbasi, H. M. Shaw, D. S. Rigel, R. J. Friedman, W. H. McCarthy, I. Osman, A. W. Kopf, and D. Polsky, "Early diagnosis of cutaneous melanoma: revisiting the abcd criteria," *JAMA Dermatology*, vol. 292, no. 22, pp. 2771–2776, 2004.

[3] S. Chatterjee, D. Dey, S. Munshi, and S. Gorai, "Dermatological expert system implementing the abcd rule of dermoscopy for skin disease identification," *Expert Systems with Applications*, vol. 167, p. 114204, 2021.

[4] K. Møllersen, H. Kirchesch, M. Zortea, T. R. Schopf, K. Hindberg, and F. Godtliebsen, "Computer-aided decision support for melanoma detection applied on melanocytic and nonmelanocytic skin lesions: a comparison of two systems based on automatic analysis of dermoscopic images," *BioMed research international*, vol. 2015, 2015.

[5] M. Goyal, T. Knackstedt, S. Yan, and S. Hassanpour, "Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities," *Computers in Biology and Medicine*, vol. 127, p. 104065, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482520303966

[6] V. Shah, P. Autee, and P. Sonawane, "Detection of melanoma from skin lesion images using deep learning techniques," in *Proceedings of the International Conference on Data Science and Engineering (ICDSE)*, 2020, pp. 1–8.

[7] A. Adegun and S. Viriri, "Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state-of-the-art," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 811–841, 2021.

[8] M. Garey, D. Johnson, and H. Witsenhausen, "The complexity of the generalized lloyd-max problem (corresp.)," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 255–256, 1982.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[10] Y. Zhou and Z. Song, "Binary decision trees for melanoma diagnosis," in *Proceedings of the International Workshop on Multiple Classifier Systems*. Springer, 2013, pp. 374–385.

[11] S. Gilmore, R. Hofmann-Wellenhof, and H. P. Soyer, "A support vector machine for decision support in melanoma recognition," *Experimental Dermatology*, vol. 19, no. 9, pp. 830–835, 2010.

[12] A. Tenenhaus, A. Nkengne, J.-F. Horn, C. Serruys, A. Giron, and B. Fertil, "Detection of melanoma from dermoscopic images of naevi acquired under uncontrolled conditions," *Skin Research and Technology*, vol. 16, no. 1, pp. 85–97, 2010.

[13] D. Ruiz, V. Berenguer, A. Soriano, and B. Sánchez, "A decision support system for the diagnosis of melanoma: A comparative approach," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15 217–15 223, 2011.

[14] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, and Q. Sun, "Deep learning for image-based cancer detection and diagnosis- a survey," *Pattern Recognition*, vol. 83, pp. 134–149, 2018.

[15] E. Nasr-Esfahani, S. Samavi, N. Karimi, S. M. R. Soroushmehr, M. H. Jafari, K. Ward, and K. Najarian, "Melanoma detection by analysis of clinical images using convolutional neural network," in *Proceedings of the 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 1373–1376.

[16] X. Zhang, "Melanoma segmentation based on deep learning," *Computer Assisted Surgery*, vol. 22, no. sup1, pp. 267–277, 2017.

[17] H. A. Haenssle, C. Fink, F. Toberer, J. Winkler, W. Stolz, T. Deinlein, R. Hofmann-Wellenhof, A. Lallas, S. Emmert, T. Buhl *et al.*, "Man against machine reloaded: performance of a market-approved convolutional neural network in classifying a broad spectrum of skin lesions in comparison with 96 dermatologists working under less artificial conditions," *Annals of Oncology*, vol. 31, no. 1, pp. 137–143, 2020.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] K. M. Hosny, M. A. Kassem, and M. M. Foaud, "Classification of skin lesions using transfer learning and augmentation with alex-net," *PloS one*, vol. 14, no. 5, p. e0217293, 2019.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedimngs of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.
[22] H. Cao, M. Wachowicz, C. Renso, and E. Carlini, "An edge-fog-cloud platform for anticipatory learning process designed for internet of mobile things," *arXiv preprint arXiv:1711.09745*, 2017.
[23] L. Rajabion, A. A. Shaltooki, M. Taghikhah, A. Ghasemi, and A. Badfar, "Healthcare big data processing mechanisms: the role of cloud computing," *International Journal of Information Management*, vol. 49, pp. 271–289, 2019.
[24] D. Połap, G. Srivastava, and M. Woźniak, "Multi-agent architecture for internet of medical things," in *Proceedings of the International Conference on Artificial Intelligence and Soft Computing*. Springer, 2020, pp. 49–58.
[25] D. S. Rigel, R. J. Friedman, A. W. Kopf, and D. Polsky, "ABCDE—an evolving concept in the early detection of melanoma," *Archives of Dermatology*, vol. 141, no. 8, pp. 1032–1034, 2005.
[26] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh, "Seven-point checklist and skin lesion classification using multitask multimodal neural nets," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 538–546, 2018.
[27] M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, and R. H. Moss, "A methodological approach to the classification of dermoscopy images," *Computerized Medical Imaging and Graphics*, vol. 31, no. 6, pp. 362–373, 2007.
[28] T. Sreelatha, M. Subramanyam, and M. G. Prasad, "Shape and color feature based melanoma diagnosis using dermoscopic images," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2020.
[29] I. Giotis, N. Molders, S. Land, M. Biehl, M. F. Jonkman, and N. Petkov, "Med-node: A computer-assisted melanoma diagnosis system using non-dermoscopic images," *Expert Systems with Applications*, vol. 42, no. 19, pp. 6578–6585, 2015.
[30] T. Lee, V. Ng, R. Gallagher, A. Coldman, and D. McLean, "Dullrazor®: A software approach to hair removal from images," *Computers in Biology and Medicine*, vol. 27, no. 6, pp. 533–543, 1997.
[31] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
[32] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
[33] M. Basu, "Gaussian-based edge-detection methods-a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 3, pp. 252–260, 2002.
[34] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate bayesian inference," *arXiv preprint arXiv:1704.04289*, 2017.
[35] S. Piotto, L. Di Biasi, S. Concilio, A. Castiglione, and G. Cattaneo, "Grimd: distributed computing for chemists and biologists," *Bioinformation*, vol. 10, no. 1, p. 43, 2014.
[36] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.

Luigi Di Biasi LuigLuigii Di Biasi is a Ph.D. student in Computer Science @ the University of Salerno. He studied Computer Science at the University of Salerno in 2013. From 2014 to 2016, he was a research fellow for PRIN 2010-2011 Data-Centric Genomic Computing (GenData 2020)). He collaborates with the GRIMD developing (grid for molecular dynamics), which permits the parallel and distributed computation of many kinds of scientific software, to the YADA developing (Yet Another Docking Approach). This tool improves and refines the predictions made by VINA Autodock and the implementation of ProtComp alignment-free tools. He is the CIO of Softmining SRL, a spin-off of the University of Salerno.

Alessia Auriemma Citarella graduated cum laude in Biology at the University of Salerno in July 2017. After graduation, she was the winner of a scholarship from the University of Salerno with research activities within the project Methodologies and Visual Analytics and Data Warehousing techniques applied to life sciences. She is currently a Ph.D. student in Computer Science at the same University and works on projects concerning Big Data Analysis. In particular, her research interests are oriented towards detecting melanoma through Machine Learning and Deep Learning techniques, with the help of physical skills in the detection phase of characteristic patterns of this pathology. Her field of application concerns Pattern Recognition, the analysis of protein and genomic Big Data, Information Visualization and modeling by Game Theory of systems responsible for the epidemiological trend of biomedical data.

Michele Risi is an Associate Professor of Computer Science at the Department of Computer Science of the University of Salerno, Italy. He was a research fellow at Department of Computer Science, 2005-2011, 2015-2016; Department of Civil Engineering, 2011-2013; Department of Management & Information Technology, 2013-2015, of the University of Salerno.

He served in the program and organized international workshops, Ph.D. schools, and conferences: ICPC, Inf. Vis., SEKE, AVI, ICSEA, DET, International Conference on Smart Cities, Systems, Devices and Technologies, and International Summer School on Software Engineering (2008-2017). In 2016, he served as Program Chair of the International Workshop on Distance Education Technologies, and from 2017 to 2020 as Organizing & Liaison Committee of the Inf. Vis. Evaluation Symposium for the International Conference on Information Visualization. He was a reviewer of high-quality international journals in the software engineering, data science, and computer science fields, e.g., Journal of Systems and Software, Journal of Data and Information Quality, Empirical Software Engineering, Multimedia Tools and Applications Software, Practice and Experience, Science of Computer Programming, Journal of Software: Evolution and Process, IEEE Transactions on Human-Machine Systems, IET Software, Soft Computing.

He is a member of the Editorial Board of the Scientific Programming Journal, Software Networking journal, Journal of Engineering & Technology for Automobile Security, and the Review Board of the International Journal of Distance Education Technologies.

Genoveffa Tortora is a full professor of Computer Science (DI) since 1990, holder of the courses of Data Bases (Bachelor's Degree in Computer Science) and Data Bases II (Master's Degree in Computer Science) and IoT Data Analytics (Internet of Things Curriculum of the Master's Degree in Computer Science). She is the Context-Aware Intelligent Systems Laboratory (CAIS Lab) director at the Department of Informatics. She is the author or co-author of over 290 articles published in journals and proceedings of international conferences. Member of the editorial board of high-quality international scientific journals. Member of the Steering Committee of the IEEE Symposium on Visual Languages. Program Chair and member of the Program Committee of several international conferences. Senior Member of the IEEE Computer Society, member of the ACM Special Interest Group on Computer-Human Interaction. Member of the EATCS (European Association of Theoretical Computer Science) Member of the IAPR (International Association of Pattern Recognition). Reviewer for several international scientific journals. Member of the editorial board of various international journals. Since 2002 he has been in the MIUR Register of Experts and, as project reviewer, has held numerous positions by the MIUR and MISE ministries and by Regions.