



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

SKIN LESION DETECTION FROM DERMOSCOPIC IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

A Degree Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona
Universitat Politècnica de Catalunya
by
Adrià Romero López

In partial fulfillment
of the requirements for the degree in
Audiovisual Systems Engineering

Advisor: Oge Marques
Co-Advisor: Xavier Giró-i-Nieto

Barcelona, January 2017

Abstract

The recent emergence of machine learning and deep learning methods for medical image analysis has enabled the development of intelligent medical imaging-based diagnosis systems that can assist physicians in making better decisions about a patient's health. In particular, skin imaging is a field where these new methods can be applied with a high rate of success.

This thesis focuses on the problem of automatic skin lesion detection, particularly on melanoma detection, by applying semantic segmentation and classification from dermoscopic images using a deep learning based approach. For the first problem, a U-Net convolutional neural network architecture is applied for an accurate extraction of the lesion region. For the second problem, the current model performs a binary classification (benign versus malignant) that can be used for early melanoma detection. The model is general enough to be extended to multi-class skin lesion classification. The proposed solution is built around the VGG-Net ConvNet architecture and uses the transfer learning paradigm. Finally, this work performs a comparative evaluation of classification alone (using the entire image) against a combination of the two approaches (segmentation followed by classification) in order to assess which of them achieves better classification results.

Experimental results for the classification task are encouraging: on the ISIC Archive dataset, the proposed method achieves an accuracy in the top three of the best previously published results. The experimental results of the segmentation evaluations demonstrate that the proposed method can outperform other state-of-the-art models.

Keywords: Medical Image Analysis, Deep Learning, Medical Decision Support Systems, Convolutional Neural Networks, Transfer Learning, Machine Learning, Melanoma, Dermoscopy, Skin Lesions, Skin Cancer.

Resum

La recent arribada de tècniques d'aprenentatge automàtic (conegeudes en anglès com a *machine learning*) i d'aprenentatge profund (*deep learning*) aplicat a l'anàlisi d'imatges mèdiques ha facilitat l'aparició de sistemes de diagnòstic intel·ligents que poden ajudar als metges i facultatius a fer decisions més encertades sobre la salut del pacient. En concret, en el camp del tractament d'imatges de pell, l'aplicació d'aquests nous mètodes tenen un alta probabilitat d'èxit.

Aquest treball es centra en el problema de detecció automàtica de lesions de pell, en particular, en la detecció de melanoma a partir d'una segmentació semàntica de la lesió i una posterior classificació en forma d'imatges dermoscopiques fent servir un model d'aprenentatge profund. Per afrontar el primer problema, la xarxa neuronal convolucionada anomenada U-Net és aplicada amb la intenció d'obtenir una acurada extracció de la regió pertinente a la lesió. Per al segon problema, el model proposat aplica una classificació binària (benigne versus maligne) que pot ser útil per a la detecció primerenca de melanoma. El model permet una generalització per a qualsevol classificació de lesions de pell. La solució proposada fa servir l'arquitectura de la xarxa convolucionada VGG-Net i investiga el paradigma de la transferència d'aprenentatge. Finalment, el projecte realitza una avaluació comparativa del cas senzill de classificació (sobre la imatge original) amb la combinació de les dues idees (segmentació seguida d'una classificació) amb la intenció d'esbrinar quina d'elles aconsegueix el millor resultat en la classificació.

Els resultats dels experiments en la tasca de classificació són encoratjadors: en el conjunt de dades (*dataset*) de l'ISIC Archive, el mètode proposat aconsegueix una mètrica d'exactitud (anomenada *accuracy*) d'entre les tres millors de tots els resultats publicats. Per altre banda, els resultats de la segmentació demostren que el model proposat pot superar altres propostes de l'estat de l'art (anglicisme de l'expressió *state of the art*).

Paraules clau: Anàlisi d'Imatges Mèdiques, Aprendentatge Profund, Sistemes de Suport de Decisió Mèdica, Xarxes Neuronals Convolucionals, Transferència d'Aprendentatge, Aprendentatge Automàtic, Dermatoscòpia, Lesions de Pell, Càncer de Pell.

Resumen

La recién llegada de técnicas de aprendizaje automático (conocidas en inglés como *machine learning*) y de aprendizaje profundo (*deep learning*) aplicados al análisis de imágenes médicas ha facilitado la aparición de sistemas de diagnóstico inteligentes que pueden dar soporte a médicos y profesionales de la salud a tomar decisiones más acertadas sobre la salud del paciente. En concreto, en el campo del tratamiento de imágenes de piel, la aplicación de estos nuevos métodos tienen una alta probabilidad de éxito.

Este trabajo se centra en el problema de detección automática de lesiones de piel, en concreto, en la detección de melanoma a partir de una segmentación semántica de la lesión seguida de una clasificación en forma de imágenes dermoscópicas utilizando un modelo de aprendizaje profundo. Con el fin de dar solución al primer problema, la red neuronal convolucionales llamada U-Net es aplicada con la intención de obtener una exacta extracción de la región pertinente a lesión. Para el segundo problema, el modelo propuesto aplica una clasificación binaria (benigno versus maligno) que puede ser útil para una detección temprana de melanoma. El modelo permite una generalización para cualquier clasificación de lesiones de piel. La solución propuesta utiliza la arquitectura de la red convolucional VGG-Net e investiga el paradigma de la transferencia de aprendizaje. Finalmente, el proyecto realiza una evaluación comparativa del caso sencillo de clasificación (sobre las imágenes originales) sobre una combinación de las dos ideas (segmentación seguida de una clasificación) con la intención de averiguar cuál de ellas logra un mejor resultado en la clasificación.

Los resultados de los experimentos en la tarea de la clasificación son alentadores: en el conjunto de datos (*dataset*) del ISIC Archive, el método propuesto consigue una métrica de exactitud (conocida como *accuracy*) de entre las tres mejores de todos los resultados a día de hoy publicados. Por otra banda, los resultados de la segmentación demuestran que el modelo propuesto puede llegar a superar otras propuestas del estado del arte (anglicismo de la expresión *state of the art*).

Palabras clave: Análisis de Imágenes Médicas, Aprendizaje profundo, Sistemas de Soporte de Decisión Médica, Redes Neuronales Convolucionales, Transferencia de Aprendizaje, Aprendizaje Automático, Dermoscopia, Lesiones de Piel, Cáncer de Piel.

Acknowledgements

To my family: Des de petit que m'heu vist obrint i tancant etapes a la meva vida. Aquest treball, com tota última prova, representa el final d'un altre. Una etapa que vaig decidir emprendre per voluntat pròpia, però que sempre he estat orgullós i content de compartir-la amb vosaltres. Vau ser vosaltres els que em vau dotar de les millors eines que podria haver imaginat: voluntat, sacrifici i capacitat de ser sempre una bona persona. Diuen que al tancar una porta, s'obren nombroses finestres petites: doncs espero que, independentment quina d'elles acabi triant, sempre us pugui trobar-hi al darrere. Gràcies Mama, Papa i Víctor. Gràcies tiets i cosins. Gracias yayo, yaya y Maribel.

To Oge Marques: I have always believed that everyone meets key people during the course of their life who, by providing a vote of confidence, enable wonderful things to happen - things that are going to change our life completely. I wholeheartedly believe that you have been one of these people for me. For the deep humanity that defines your being, for trusting me since the beginning, for letting me live the American Dream at FAU, for the patience you have shown with me and for keeping an eye on me during my stay: muito obrigado, amigo.

To Xavier Giró: Per brindar-me una oportunitat vital que sense cap mena de dubte m'ha obert la porta al món de la recerca. Per la dedicació que poses als alumnes, la bona feina i professionalitat que has desplegat a l'escola. Gràcies.

To the MIDDLE research group (Oge, Jack, Janet and Adam): For the excellent job done and for the valuable suggestions and ideas when we were in need of them. My sincere thanks for giving me the opportunity to work with all of you.

To Jack Burdick: For your willingness to help me, for your patience and understanding while we worked together. I wish you the best in whatever path you choose, my friend.

To Jiménez, Víctor, Fullana, Chema and Ferran: No em puc referir a cada un de vosaltres sense fer menció al grup sencer. Gràcies pels bons moments junts, per l'amistat que va néixer farà ja temps i perdurarà per molt de temps. Jiménez, por todas las risas compartidas contigo; Víctor, por la teva genuïtat; Fullana, pel teu carisma; Chema, por tu inmenso corazón; Ferran, per la felicitat que sempre transmets.

To Dan: Brindo por el momento de encontrarnos en un laboratorio de Fundamentos de Física y Electrónica, y por el camino que desde ese día hemos vivido juntos. Un camino enrevesado y lleno de duros obstáculos que fue forjando una bonita y sincera amistad hasta el día de hoy, en que admito que te echaré en falta. Sé que eres consciente que una parte muy grande del hecho de que yo haya llegado hasta aquí ha sido gracias a ti. No me equivoco en predecir que esta simbiosis nos seguirá acompañando muchos años más. Vă mulțumesc foarte mult, prietene.

To Albert Gil and the GPI (Image Processing Group) from UPC for the technical advice provided during my project.

And finally, to Andrea Calafell for creating and sharing the L^AT_EX template used in this work.

Revision history and approval record

Revision	Date	Purpose
0	16/12/2016	Document creation
1	13/01/2017	Document revision
2	14/01/2017	Document revision
3	15/01/2017	Document approbation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Adrià Romero López	adria.romero@alu-etsetb.upc.edu
Oge Marques	omarques@fau.edu
Xavier Giró i Nieto	xavier.giro@upc.edu

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date		Date		Date	
Name	Adrià Romero López	Name	Oge Marques	Name	Xavier Giró i Nieto
Position	Project Author	Position	Project Supervisor	Position	Project Co-Supervisor

Contents

1 Introduction	11
1.1 Background of the Problem	11
1.2 Statement of Purpose	12
1.3 Requirements and specifications	13
1.3.1 ISBI Challenge	13
1.3.2 Framework used	14
1.3.3 GPU power	14
1.4 Methods and Procedures	14
1.4.1 Skin Lesion Segmentation	14
1.4.2 Skin Lesion Classification	15
1.5 Work Plan	15
1.5.1 Work Packages	16
1.5.2 Gantt diagram	16
1.5.3 Deviations and Incidents	17
2 State of the art	18
2.1 Artificial Neural Networks	18
2.2 Convolutional Neural Networks	19
2.2.1 ConvNets for Image Classification of Skin Lesions	20
2.2.2 ConvNets for Medical Image Segmentation	21
2.3 Transfer Learning	22
3 Methodology	23
3.1 Objective	23
3.2 Datasets	24
3.3 Neural Network Architecture	25

3.3.1	U-Net	25
3.3.2	VGG-16	26
3.4	Preprocessing	26
3.5	Data Augmentation	27
3.6	Training methods	28
3.6.1	Definition of hyper-parameters	28
3.6.2	Segmentation	29
3.6.3	Classification	29
4	Results	31
4.1	Experimental setup	31
4.2	Metrics	31
4.3	Results	32
4.3.1	Training the U-Net for Skin Lesion Segmentation	32
4.3.2	Fine-tuning the VGG-16 for Skin Lesion Classification	33
5	Budget	37
6	Conclusions	38
A	BioMed 2017 paper	40
B	SIIM 2017 abstract	47
C	Segmentation Evaluation	52
D	Segmentation examples	63

List of Figures

1.1	Sample images created from the ISIC Archive dataset[1]	12
1.2	Binary masks examples	15
1.3	Overall architecture proposed for this project	15
1.4	Gantt diagram	16
2.1	Mathematical model of the biological neuron	19
2.2	A neural network with six inputs, two hidden layers and one single output	19
2.3	LeNet-5 [2] CNN architecture for handwritten digits recognition	20
2.4	U-Net, Convolutional Networks for Biomedical Image Segmentation results: (a) HeLa cells raw images recorded with DIC (differential interference contrast) microscopy, (b) ground truth segmentation, (c) generated segmentation mask, (d) pixel-wise map related to the loss weight	21
2.5	Semantic segmentation networks [3] using deep deconvolutional layers	21
3.1	Image summary: (a) Original image, (b) Original binary mask, (c) Automatically generated binary mask, (d) Perfectly segmented image, (e) Automatically segmented image	24
3.2	U-Net Architecture (example for 32×32 pixels in the lowest resolution)	25
3.3	VGG-16 Architecture	27
3.4	Data augmented preview	28
3.5	Training methodology [4]	28
4.1	Examples of satisfactory segmentation results	33
4.2	Examples of poor segmentation results	33
4.3	Confusion Matrices	34
4.4	Unaltered lesion classification: Precision & Recall and ROC curve	35
4.5	Perfectly segmented lesion classification: Precision & Recall and ROC curve	35
4.6	Automatically segmented lesion classification: Precision & Recall and ROC curve	35



4.7 Examples of True and False predictions done by the automatically segmented lesion model	36
---	----

List of Tables

3.1 ISIC Dataset [1] distribution	24
4.1 Segmentation results benchmarking against the Challenge Top 3	32
4.2 Classification results comparison	34
5.1 Budget of the project	37

Chapter 1

Introduction

1.1 Background of the Problem

Skin is the largest organ of the body. Its importance comes from the way it protects the internal body tissues from the external environment, i.e., skin keeps the body temperature at a constant level, protects our body from undesirable sun radiation such as ultraviolet (UV) light exposure, prevents infections and allows the production of vitamin D, essential for many body functions [5].

In the past few years the number of skin cancer cases has been going up and studies announce that the frequency of melanoma doubles every 20 years [6]. Skin cancer, the most predominant type of cancer, is produced when skin cells begin to grow out of control. There are 3 main types of skin cancers: basal cell skin cancers (basal cell carcinomas), squamous cell skin cancers (squamous cell carcinomas) and melanomas. Generally, skin cancers that are not melanomas are commonly grouped as non-melanoma skin cancers. [7]

This project is focused on melanoma detection, which is a fatal form of skin cancer often undiagnosed or misdiagnosed as a benign skin lesion [8]. There are an estimated 76,380 new cases of melanoma and an estimated 6,750 deaths each year in the United States [9]. Melanoma can be detected by simple visual examination since it occurs on the skin surface, but an early detection is imperative: the lives of melanoma patients depend on accurate and early diagnosis.

This poses additional challenges to the task of distinguishing among skin lesions, especially between benign or malignant tumors, due to the large imbalance in the number of samples of each class of tumors. These aspects must be kept in mind when designing (semi-)automatic skin lesions classifiers systems whose performance should be at least comparable to traditional detection methods. That is the reason to distinguish among skin lesions, especially between benign or malignant cancer, has allowed the emergence of automatic skin lesions classifiers systems to compete against traditional detection methods.

Most current methods in the field of skin lesion classification rely on hand-crafted features, such as *ABCDE rule* (the acronym stands for Asymmetry, Border, Color, Dermoscopic structure and Evolving) [10], *3-point checklist* [11], *7-point checklist* [12], *Menzies method* [13] and *CASH* (Color, Architecture, Symmetry, and Homogeneity) [14]. Physicians often rely on personal experience and evaluate each patient's lesions on a case-by-case basis by taking into account the patient's local lesion patterns in comparison to that of the entire body [15]. Without any type of computer-based assistance, the clinical diagnosis accuracy for melanoma detection is reported to be between 65 and 80% [16]. Use of dermoscopic images, pictures taken skin by a skin surface microscopy [17], improves diagnostic accuracy of skin lesions by 49% [18]. However, the visual differences between melanoma and benign skin lesions can be very subtle (Figure 1.1) , making it difficult to distinguish the two cases, even for trained medical experts.

For the reasons described above, an intelligent medical imaging-based skin lesion diagnosis system can be a welcome tool to assist a physician in classifying skin lesions.

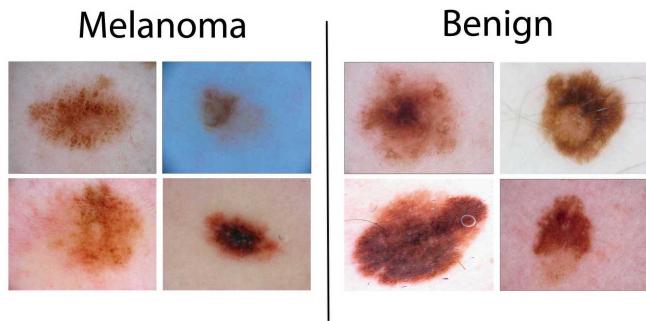


Figure 1.1: Sample images created from the ISIC Archive dataset[1]

1.2 Statement of Purpose

The emergence of a machine learning paradigm known as deep learning and recent advances in computational power have enabled the development of intelligent medical image analysis systems that can display remarkable performance in comparison to hand-crafted features. Artificial neural networks have produced promising results when classifying skin lesions [19][20][21].

The purpose of this work is to assess the performance of *Convolutional Neural Networks* (also known as ConvNets or CNNs) [22] in building models for the diagnosis of skin lesions. To achieve this goal, this work is divided into three parts:

- The development of an algorithm for automated prediction of *skin lesion segmentations* from dermoscopic images in the form of binary masks.
- The design of a *method for automatic classification* of skin lesions from dermoscopic images.
- The evaluation of *skin lesion segmentation impact* on the accuracy of the designed classifier.

The hypothesis of this work states that previous segmentation of an image containing a skin lesion (i.e., isolating the lesion from the background) improves the accuracy and sensitivity of a Deep Learning model approach for a 2-class classifier for early melanoma detection based on skin lesion dermoscopic images. This assumption considers that segmentation removes nonessential information, such as hair or non-target surrounding lesions, and helps in accurately classifying the lesion. However, segmentation may lose information that could be contextually relevant to the CNN. Segmenting images using state of the art techniques provides promising but not perfect results and has yet to be proven to be robust across different datasets.

This project was carried out during the Fall semester 2016 at the Department of Computer & Electrical Engineering and Computer Science (CEECS) [23] of the Florida Atlantic University (FAU). Additionally, it was supervised by the Barcelona School of Telecommunications Engineering (ETSETB) of the Polytechnic University of Catalonia (UPC). Furthermore, this project contributed to the Medical Imaging Diagnosis using Deep LEarning (MIDDLE) research group from FAU.

1.3 Requirements and specifications

This thesis was one of initial works of the MIDDLE research group at the Florida Atlantic University, so an overall goal for it was setting the grounds for further research projects in the field of deep learning for medical image analysis. In particular, it was required to create a dynamic environment tool easy to adapt and update by future members of the group. The basic tasks the required software package were the following:

- Train and evaluate a pre-existent convolutional neural network to *properly segment the skin lesion*.
- Train and evaluate a pre-existent convolutional neural network to *accurately classify the skin lesion*.
- Combine the results obtained in the two first requirements in order to improve the classification performance task.

The performance and results of the developed tools were also required to be assessed and published in three international scientific forums: a challenge at the ISBI Symposium, a publication at the BioMed 2017 Conference and an abstract at the SIIM 2017 Annual Meeting.

1.3.1 ISBI Challenge

The existence of the IEEE International Symposium on Biomedical Imaging (ISBI) [24] 2016 challenge for Skin Lesion Analysis Towards Melanoma Detection [25] has provided this project with a useful dataset, the ISIC Archive dataset [1], which contains dermoscopic images paired with their corresponding lesion binary masks, and the possibility of benchmark the results obtained against other participants.

The organisation of the challenge provided the possibility of benchmark the results obtained against the top-ranked participants and evaluate how well our model performs. Moreover, the organization provides a guideline of how to develop the entire project by dividing the work into three parts: (I) Lesion Segmentation [26], (II) Lesion Dermoscopic Feature Extraction [27], (III) Lesion Classification [28], in particular, the organization proposed a subsection involving a previous segmentation of the lesion in order to study if this technique improves or not the classification task [29].

This project has followed the specifications and evaluation metrics proposed by the ISBI 2016 Challenge, but the development of this thesis was posterior to the event. Given the satisfactory results obtained over the 2016 dataset, the MIDDLE group has expressed its intention to participate to the coming ISBI 2017 Challenge [30]. The new edition of the challenge provides participants with a larger skin image dataset, new dermoscopic features for extraction, additional classification categories and new evaluation metrics.

1.3.2 Framework used

Python [31] was the main programming language chosen to develop the project. In some specific tasks Matlab [32] was also used.

Keras [33] is a deep learning framework that was initially released on March 2015 [34]. Keras provides a layer of abstraction on top of TensorFlow [35] or Theano [36], which is used as the main neural network framework. Keras is compatible with Python 2.7 and 3.5. It allows for: (1) modularity: users can define their neural networks following a sequence which is a linear stack of layers; (2) minimalism: functions included in the library allow the user to create and modify network layers easily; and (3) extensibility: daily updates provide solutions to ongoing challenges faced by deep learning researchers. Moreover, Keras works on a Python environment, which gives users the freedom to use additional Python dependencies, including SciPy [37] and PIL [38]. For all these reasons, Keras was used to implement the neural network architectures on this project.

1.3.3 GPU power

In addition to Keras, CUDA libraries [39] for parallel computing and programming model developed by NVIDIA were required to utilize the GPUs (Graphics Processing Unit). NVidia GeForce GTX TITAN X [40] GPUs were used to train and evaluate the implementation. The GPU power supply was provided by the *Image Processing Group* (GPI) from UPC.

1.4 Methods and Procedures

The baseline of the project follows the guidelines proposed by the ISBI Challenge (see Section 1.3.1), with few modifications that can be found explained below. The challenge consisted of three tasks: lesion segmentation, dermoscopic feature extraction, and disease classification. This project does not address the second task since it is not related to deep learning techniques. This project uses the same evaluation criteria as the ones used in the challenge.

First and foremost, all input images had to be preprocessed to be compatible with the format and size expected by the input layer of the proposed architecture.

1.4.1 Skin Lesion Segmentation

Typically, segmentation is used as a preprocessing method in the classification process to remove potentially non-relevant information from the classification process [41]. Through segmentation, those pixel values that do not directly represent the lesion, are discarded so that only the lesion values are considered by the network.

As a first contribution, we explored the use of Convolutional Neural Networks (ConvNets) for a fast and precise segmentation of medical images. The automated predictions of the skin lesion from the dermoscopic images were generated in the form of binary masks (Figure 1.2) (pixel values of 255 were considered inside the lesion area, while pixel values of 0 were considered to be outside).

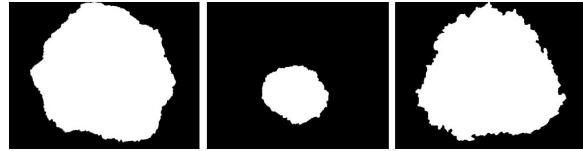


Figure 1.2: Binary masks examples

1.4.2 Skin Lesion Classification

The image classification task was addressed with a pretrained ConvNet to perform a 2-class classification as either malignant or benign skin tumor (Figure 1.3). To properly test the project's hypothesis, it was necessary to compare the performance metrics of classification performed on:

- Raw dermoscopic skin images through the ISIC dataset [1] original images.
- Manually segmented dermoscopic skin images taking advantage of the binary masks provided by the ISIC dataset [1].
- Segmented dermoscopic skin images obtained by an automatic algorithm, as previously introduced in Section 1.4.1.

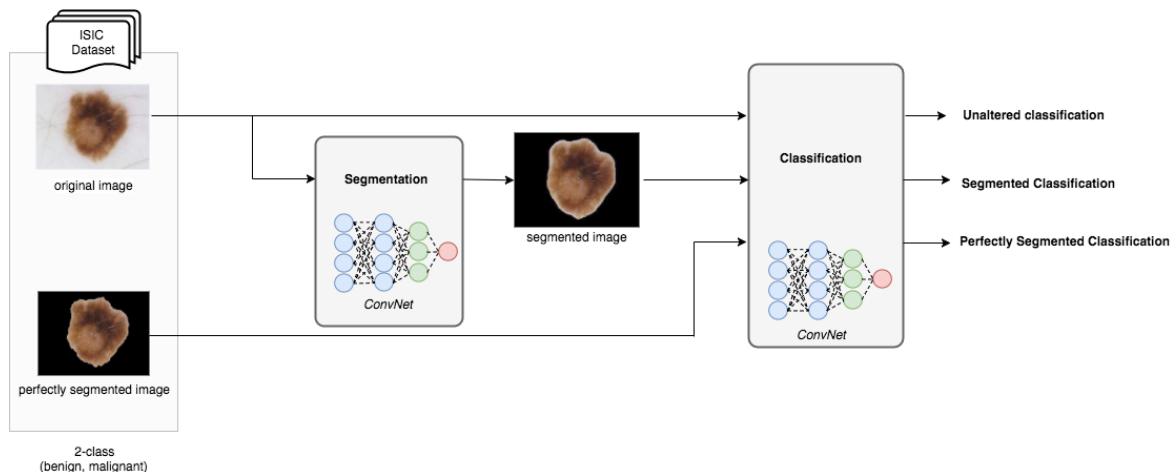


Figure 1.3: Overall architecture proposed for this project

The main goal of the project was comparing the three approaches and assess their relative benefits and limitations. All three subtasks were compared by fixing all the rest of condition and parameters of the experiments.

1.5 Work Plan

This project followed the work packages detailed in this section, with the exception of some minor modifications and deviations described in Section 1.5.3.

1.5.1 Work Packages

- WP 1: Project documentation
- WP 2: Research the state of the art
- WP 3: Dataset selection
- WP 4: Software and Hardware selection and configuration
- WP 5: Experimentation
- WP 6: Results
- WP 7: Documents delivery & presentation
- WP 8: Conference participation (BioMed 2017 & SIIM 2017)

1.5.2 Gantt diagram

The Gantt diagram of Figure 1.4 illustrates the work breakdown structure of the project.

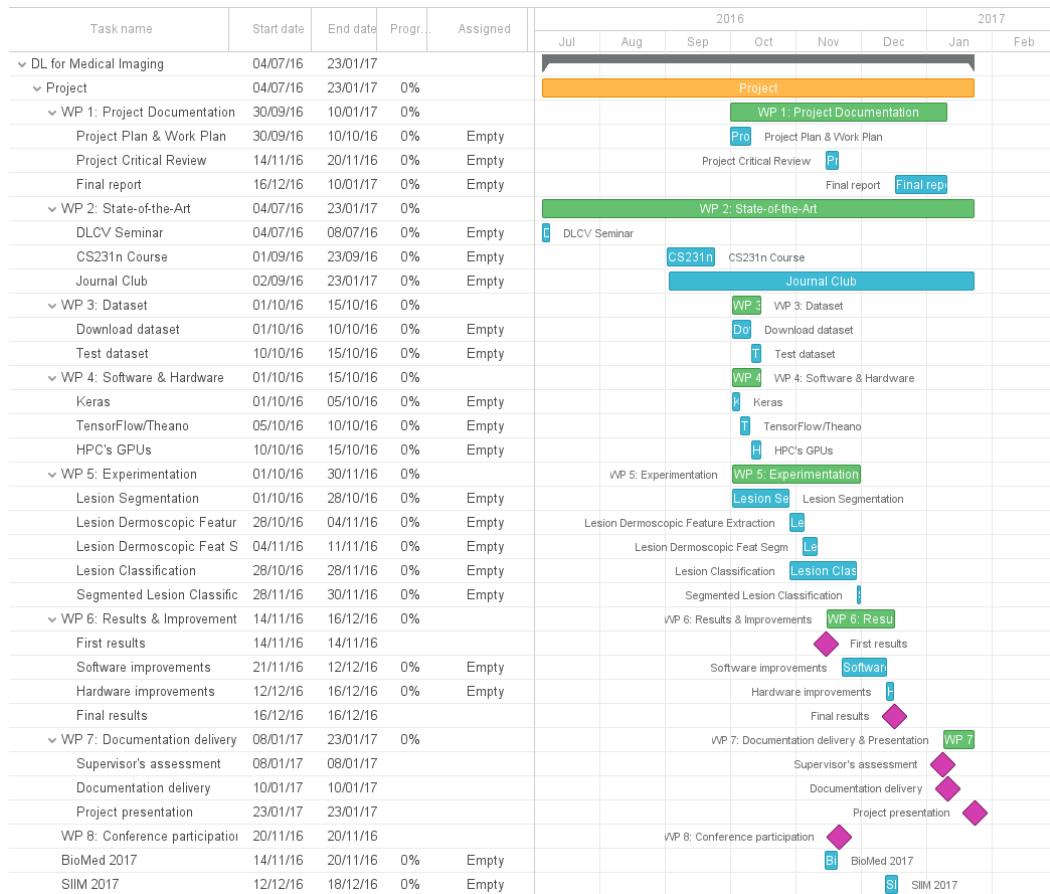


Figure 1.4: Gantt diagram



1.5.3 Deviations and Incidents

The only incidence the project suffered was related to the GPU power supply by the hosting university. Despite the availability of GPUs, the documentation was insufficient and the staff was not capable of assisting with technical questions in a timely manner, which caused delays that could have potentially jeopardize the entire project schedule. The problem was solved by using the computation service from the Image Processing Group at UPC instead of the machines at FAU. This way, while the project was developed in Florida, the actual computation was run in Barcelona.

Chapter 2

State of the art

During the past decades many approaches have been proposed to automatically generate image representations that can provide support to tasks like image classification, object detection, recognition or semantic segmentation. Most of them have relied on hand-engineered low-level descriptors. But since the publication of AlexNet in 2012, state-of-the-art methods in computer vision mostly rely on learning representations using deep convolutional neural networks.

Most current methods in the field of melanoma classification still rely on hand-crafted features (described in Section 1.1). Typically, after the feature extraction based on these descriptions, machine learning methods such as k-nearest neighbors (kNN), Artificial Neural Networks (ANNs), logistic regression, decision trees and support vector machines (SVMs) have been explored to solve the classification task with moderate success[42].

Examples of related work using hand-crafted features and popular classifiers include:

- Codella et al. [21] use hand-crafted feature extraction techniques including color histogram, edge histogram, and a multi-scale variant of color local binary patterns (LBP).
- Barata et al.[43] explore two different methods for the detection of melanoma in dermoscopy images based on global and local features. They conclude that color features perform much better than texture features alone.

More recently, the emergence of a machine learning paradigm known as *deep learning* has enabled the development of medical image analysis systems that can display remarkable accuracy, to the point of raising concerns about the future of the human radiologist [44][45].

The next sections provide an overview of the deep learning techniques for image classification. The reader is referred to the *Deep Learning for Computer Vision 2016 Summer seminar* [46] by UPC TelecomBCN and the very well known *CS231n: Convolutional Neural Networks for Visual Recognition* course [47] by Stanford University for a more detailed explanation

2.1 Artificial Neural Networks

Artificial Neural Networks (known as ANNs) are a category of machine learning algorithms whose design has been inspired by the neurophysiological workings of the human brain.

ANNs use a mathematical model (Figure 2.1) of a neuron, in which the input nerve impulse (x_i) is multiplied by a learnable matrix of *weights* (w_i), that represent the synaptic strengths of neurons. The second parameter that the model can learn is called the *bias* term (b), that is directly added to the elementwise multiplication of previous matrices. The mathematical model of the neuron will *fire* the output signal ($x_i w_i$) according to an activation function (f), which introduces a non-linearity to the equation. Considering the multiple impulses a neuron can receive as an input, the output of the mathematical model can be expressed as Equation 2.1.

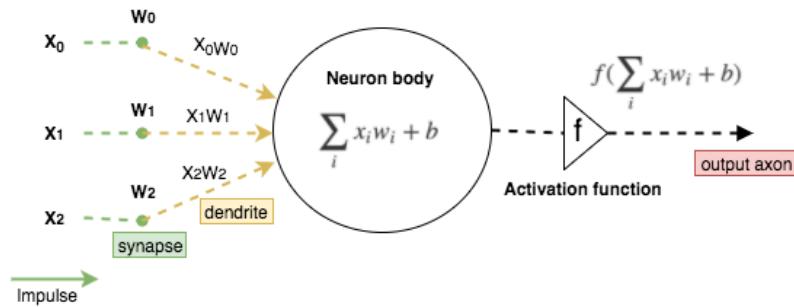


Figure 2.1: Mathematical model of the biological neuron

$$f\left(\sum_i x_i w_i + b\right) \quad (2.1)$$

The most common activation functions used in ConvNet architectures are the *sigmoid function*, with expression: $\sigma(x) = 1/(1 + e^{-x})$) and the *ReLU* (Rectified Linear Unit): $f(x) = \max(0, x)$ that applies a threshold at zero.

Neural Network contain millions of neurons organized into three kinds of layers (Figure 2.2): (i) the input layer (neurons in this layer do not have inputs themselves) (ii) hidden layers, connected to the input layer, and (iii) the output layer, connected to the hidden layers. The most common layer organization is the Fully-Connected layer, where each neuron is fully paired with adjacent neurons.

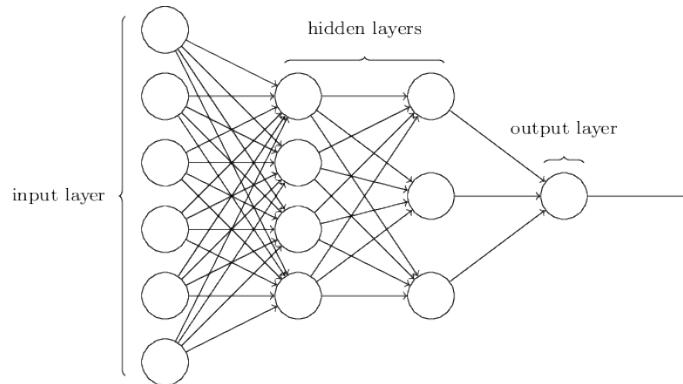


Figure 2.2: A neural network with six inputs, two hidden layers and one single output

2.2 Convolutional Neural Networks

Convolutional Neural Networks (also known as CNNs or ConvNets) maintain a strong relationship with Artificial Neural Networks: they are also inspired in the behavior of biological systems through artificial neurons with learnable weights and biases. The layered architecture that Neural Networks performs based on matrix multiplications enables its application for image classification tasks. For this reason, ConvNets architectures assume that the input are images that have to

be transformed into an output holding the class score predicted. The loss function is used to measure how well the predicted scores agrees with the ground truth labels in the input data. Most common loss functions are the Multiclass Support Vector Machine (SVM) (Equation 2.2) and the  (Equation 2.3).

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta) \quad (2.2)$$

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \quad (2.3)$$

ConvNets work similarly to Neural Networks: each neuron receive an input, a dot product (Hadamard product or elementwise multiplication) between each input and its associated weight is performed, followed with a non-linearity. The most common hierarchical distribution of ConvNets layers contains:

- Input layer, containing the raw pixel values from input images.
- Convolutional layers, the core block of ConvNets, computes a locally dot product (2D in the case of images) between the weights and a certain tiny region of the input volume.
- Non-linear layers, most of the times using a ReLU activation function which applies an elementwise activation by thresholding at zero.
- Pooling layers that apply a spatial downsampling along the output volume.
- Fully Connected layers that compute the class scores,

A CNN structure (Figure 2.3) is made up of repetitive patterns (which explains the expression *deep learning*) of Convolutional, ReLU and Pooling layers (considered *hidden layers*) and finally the fully-connected layers. The resulting volume structure is called feature map (in the case of images, it has a two dimension volume). The learning process (also referred to network training) where weights are optimized is achieved through backpropagation [2], a technique to efficiently compute gradients for its weights with respect to the loss function.

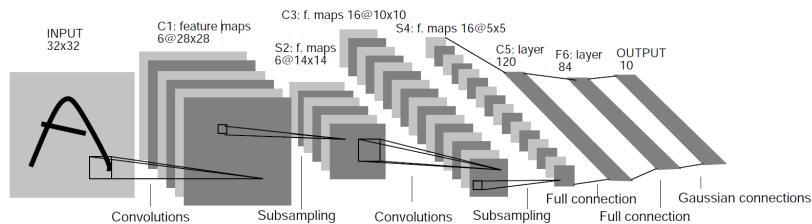


Figure 2.3: LeNet-5 [2] CNN architecture for handwritten digits recognition

2.2.1 ConvNets for Image Classification of Skin Lesions

Convolutional neural networks have produced promising results when classifying skin lesions. Examples of related work using deep learning include:

- The work of Kawahara et al.[19] explores the idea of using a pretrained ConvNet as a feature extractor to distinguish among 10 classes of non-dermoscopic skin images.
- Liao [20] describes an attempt to construct a universal skin disease classification by applying transfer learning 2.3 on a deep CNN and fine-tuned its weights.
- Codella et al. [21] report state-of-the-art performance results using ConvNets to extract image descriptors by using a pre-trained model from the *Imagenet Large Scale Visual Recognition Challenge (ILSVRC) 2012* [48]. They also investigate a more recent network structure called Deep Residual Network (DRN) [49].

2.2.2 ConvNets for Medical Image Segmentation

ConvNets are typically used on image classification tasks: in the case of *supervised learning*, the input are images sorted by class, and the output to a certain image is a single class label. However, in the biomedical image processing field, a localization is often also required in addition to a global scale label, i.e. the network assigns a class label to each pixel. This is the main idea of a semantic segmentation model [50] using ConvNets (Figure 2.4).

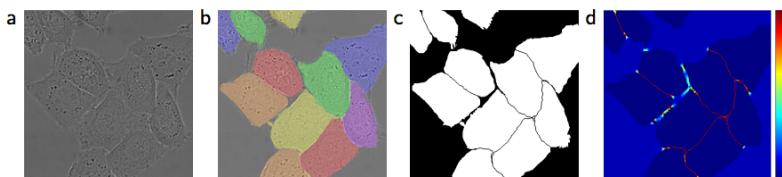


Figure 2.4: U-Net, Convolutional Networks for Biomedical Image Segmentation results: (a) HeLa cells raw images recorded with DIC (differential interference contrast) microscopy, (b) ground truth segmentation, (c) generated segmentation mask, (d) pixel-wise map related to the loss weight

Novel proposals for semantic segmentation [3] using convolutional neural networks introduce the idea of deconvolutional networks on the top of common ConvNets. This backward strided convolution is convenient to generate a segmentation map of the input through a sequence of deconvolution operations. These layers (shown in Figure 2.5) compute a dense pixel-wise class probability map by consecutive operations of unpooling, deconvolution, and rectification.

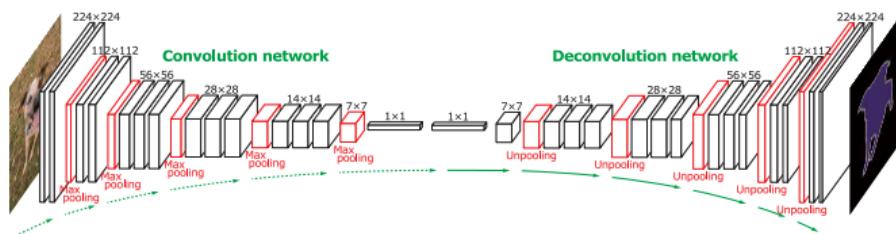


Figure 2.5: Semantic segmentation networks [3] using deep deconvolutional layers

2.3 Transfer Learning

Training an entire convolutional neural network in the medical imaging field is not always possible, due to the fact that datasets are often not large enough. Alternatively, random initialization of weights is replaced by a pretrained network on large datasets, i.e. ImageNet [48], that contains 1.2 million images labeled with 1,000 classes. This technique is known as Transfer Learning [51] and it is very common in machine learning scenarios. The main objective is the improvement of learning in a new task by transferring knowledge from a related task that has already been learned.

In practice, Transfer Learning from a pretrained ConvNet is typically used in these two different ways [52]:

- **Fixed Feature Extraction.** Use the output of one intermediate layer to train a machine learning classifier.
- **Fine-tuning.** Replace the last fully connected layer of the network with a new classification task and use the training data from the new domain to update the weights of some layers. In general, the earlier layers of the ConvNets are not updated and only the deeper ones are updated to the new task.

These two techniques are explored in this work and can be better understood in Chapter 3 - Methodology, Section 3.6 .

Chapter 3

Methodology

3.1 Objective

This work develops a study in the Deep Learning field about the impact and tradeoffs of removing skin image background by applying a semantic segmentation for a subsequent classification of the disease. By isolating and comparing results from both unaltered and segmented skin lesion images, this work aims to better understand the impact on performance results when segmenting an image. Specifically, it hopes to better understand whether the values outside the lesion are detrimental to lesion classification, or are instead beneficial to lesion classification by providing contextual information relevant to each lesion.

In order to achieve the project goal, two successful and well-known Convolutional Neural Networks architectures in the image semantic segmentation and image classification tasks have been adopted. The order of execution will be:

1. **Skin lesion segmentation.** The first task will perform an automatic prediction of lesion segmentation from dermoscopic images taking the form of binary masks.
2. **Skin lesion classification.** The second task will perform the automatic classification as either melanoma or non-melanoma. In order to find the best classification model, this task will be divided into three subtasks according to different type of input skin images:
 - (a) **Unaltered lesion classification.** The basic model will perform the classification over the original skin RGB images contained in the ISIC dataset.
 - (b) **Perfectly segmented lesion classification.** The perfectly segmented image will be generated by performing a bit-wise *and* operation on the original images and its corresponding original binary mask contained in the ISIC dataset. The second model will perform the classification over the perfectly segmented images.
 - (c) **Segmented lesion classification.** The automatically segmented image will be generated by performing a bit-wise *and* operation on the original images and its corresponding binary mask generated automatically during the first task 1. The third, and most complex model will perform the classification over the automatically segmented images.

A summary representation of input images to be classified is shown in Figure 3.1.

A proper evaluation under the same conditions will be performed over the three classification models and will confirm or refuse the hypothesis of this work.

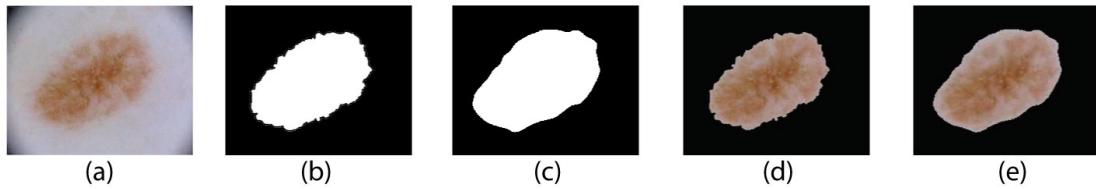


Figure 3.1: Image summary: (a) Original image, (b) Original binary mask, (c) Automatically generated binary mask, (d) Perfectly segmented image, (e) Automatically segmented image

3.2 Datasets

The ISBI 2016 Challenge[25] dataset[1] for *Skin Lesion Analysis towards melanoma detection* was used for this work.

The dataset is publicly available and contains 1279 RGB images that are pre-partitioned into 900 training images and 379 validation images. All images are labeled as either benign or malignant and include a binary image mask to label the lesion within the image.

Class		
	Benign	Malignant
Training subset	727	173
Validation subset	304	75
	Total images	
	900	
	379	

Table 3.1: ISIC Dataset [1] distribution

In order to solve the different problems addressed by this project, four variations of the ISIC dataset were used:

- *Segmentation dataset*. The original ISIC dataset split into training and validation subsets (note that a division between classes is not needed for the segmentation problem).
- *Unaltered lesion classification dataset*. The original ISIC dataset excluding the binary masks.
- *Perfectly segmented lesion classification*. This dataset contains the perfectly segmented images (see Section 3.1 for a better understanding).
- *Automatically segmented lesion classification dataset*. This dataset contains the automatically segmented images (see Section 3.1 for a better understanding).

Due to the unbalanced nature (see Table 3.1) of the dataset (few malignant class samples compared to the benign class samples), the training and validation subsets were previously weighted by the *class weight* function provided by Keras framework. The weighting was done in relationship to the dataset partition: every instance of benign class as 4.2 instances of malignant class, since benign class represents approximately the 80% of training and validation subsets, while malignant class is the 20%.

3.3 Neural Network Architecture

Nowadays, there are several convolutional neural networks architectures that achieved successful results on benchmark challenges such as the *Imagenet Large Scale Visual Recognition Challenge* (ILSVRC) [48], and as a consequence became very popular in the *deep learning* community.

In particular, the *U-Net* was in charge of the semantic segmentation task, while the *VGG-16* was chosen for the classification task among popular architectures such as the *AlexNet* [22] and the *GoogLeNet* [53].

3.3.1 U-Net

The *U-Net* [54] is a ConvNet architecture specifically designed by O. Ronneberger et al. from the University of Freiburg to solve Biomedical Image Segmentation problems. It was successfully rated for winning the ISBI cell tracking challenge [55] 2015.

The network architecture is illustrated in Figure 3.2. As explained in Section 2.2.2 (ConvNets for Image Segmentation), the network merges a convolutional network architecture (contracting path on the left side) with a deconvolutional architecture (expansive path on the right side) to obtain the semantic segmentation. The convolutional network is composed of a repetitive pattern of two 3×3 convolutions operations, followed by a ReLU layer and a downsampling process through a 2×2 maxpooling operation with stride 2. On the other hand, the deconvolutional architecture includes a upsampling operation of the feature map obtained during the contracting path, followed by a 2×2 deconvolution that fractions the feature map channels into 2. A posteriori concatenation of the resulting feature map and the obtained during the contracting path is needed, followed by a 3×3 convolutions and a ReLU layer. The entire network is 23 convolutional layers deep, where the last layer is used to map each component feature vector related to the number of classes.

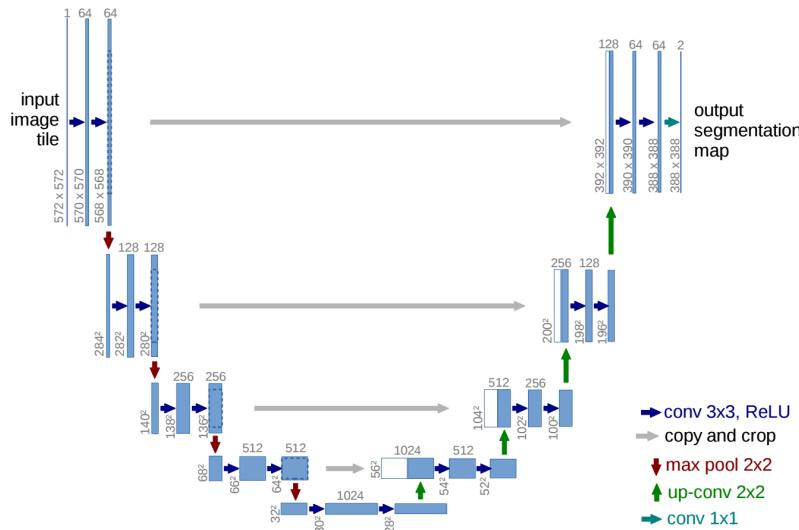


Figure 3.2: U-Net Architecture (example for 32×32 pixels in the lowest resolution)

The architecture accepts RGB input images and their corresponding binary masks. The training is done by learning weights with the stochastic gradient descent implemented in Keras [56].

3.3.2 VGG-16

VGGNet is a very well-documented and commonly used architecture for convolutional neural networks [57]. This ConvNet became popular by achieving excellent performance on the ImageNet [48] dataset. It comes in several variations of which the two best-performing (with 16 and 19 weight layers) have been made publicly available. In this work, the VGG16 architecture (Figure 3.3) was selected, since it has been shown to generalize well to other datasets.

The input layer of the network expects a 224×224 pixel RGB image. The input image is passed through five convolutional blocks. Small convolutional filters with a receptive field of 3×3 are used. Each convolutional block includes a 2D convolution layer operation (the number of filters changes between blocks). All hidden layers are equipped with a ReLU as the activation function layer (nonlinearity operation) and include spatial pooling through use of a max-pooling layer. The network is concluded with a classifier block consisting of three Fully-Connected (FC) layers.

The original VGG16 must be modified to suit our needs, as follows:

- The final fully-connected output layer performs a binary classification (benign-malignant), not 1000 classes.
- The activation function in the modified layer has been changed from Softmax to Sigmoidal.

3.4 Preprocessing

This project takes advantage of ConvNets properties regarding input preprocessing: few previous processing techniques are needed. Although some basic preprocessing forms are performed:

- **Mean subtraction.** In order to center the cloud of RGB values from input data around zero along every dimension of the image, a mean subtraction is applied across the image features.
- **Image normalization.** By dividing each RGB dimension of input images by its standard deviation, a normalization is obtained from its original 0 and 255 pixel values to 1 and 0 normalized values. This preprocessing technique will avoid further issues caused by poor contrast images.
- **Image cropping & resizing.** Input images are preprocessed to be accepted by the architecture through cropping the image to the same aspect ratio as needed and resizing the original image to 64×80 pixels for the U-Net and 224×224 pixels for the VGG-16.

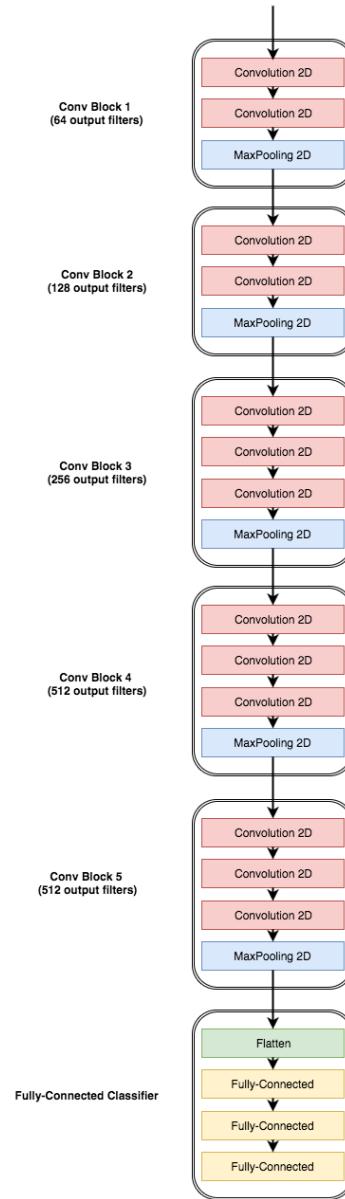


Figure 3.3: VGG-16 Architecture

3.5 Data Augmentation

In order to make the most of our few training examples and increase the accuracy of the model, the ISIC dataset was augmented via a number of random transformations. The selected data augmentation techniques were: size re-scaling, rotations of 40 degrees, horizontal shift, image zooming, and horizontal flipping. Furthermore, it is expected that data augmentation should also help prevent *overfitting* (a common problem in machine learning related to small datasets, when the model, exposed to too few examples, learns patterns that do not generalize to new data) and, for this reason, improving the model's ability to generalize.

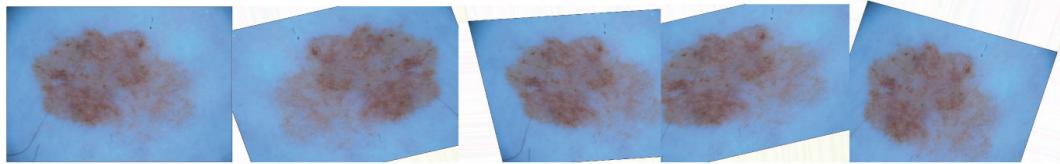


Figure 3.4: Data augmented preview

3.6 Training methods

The same training methodology is proposed to face the two tasks of this project. The only difference between the segmentation and classification tasks are the models used. As previously commented in Section 3.3, the U-Net architecture is the architecture proposed for the segmentation task, while the VGG-16 is associated with a classification model. Following the diagram of Figure 3.5, the training data is trained through the learning algorithm defined by each model, which applies the *stochastic gradient descent* (SGD). Once the model has learned the weights, a prediction algorithm classifies the validation data according to the training. A final model evaluation is performed by comparing the predictions with the ground truth data.

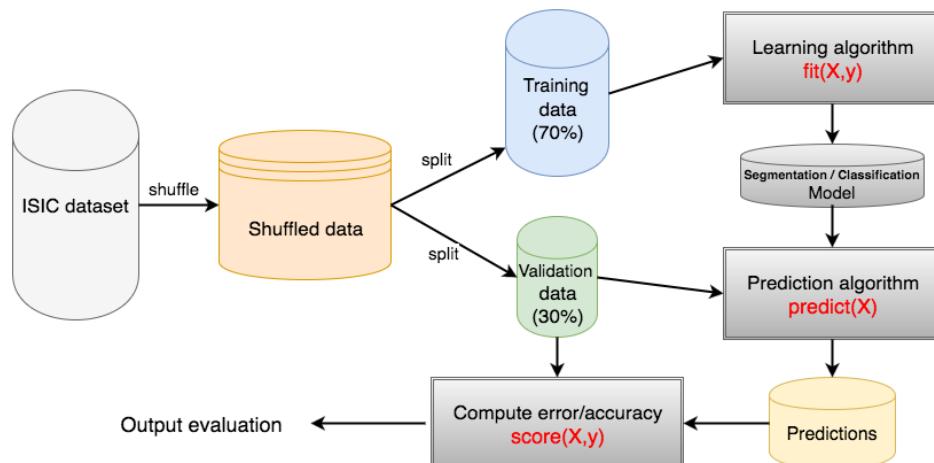


Figure 3.5: Training methodology [4]

A detailed explanation for each task is done in Sections 3.6.2 and 3.6.3.

3.6.1 Definition of hyper-parameters

During training, some parameters must be considered to be altered in order to get the best performance of the network proposed regarding the problem to be solved. Typical ConvNets parameters are the following:

- **Batch size.** The *batch size* is attributed to the number of training images in one forward or backward pass. It is important to highlight that the higher the batch size, the more memory will be needed.
- **Iterations.** The number of *iterations* is the number forward or backward of passes: each pass using a batch size number of images.

- **Epoch.** The number of *epochs* measures how many times every image has been seen during training (i.e. one epoch means that every image has been seen once). It can be also understood as a one forward pass and one backward pass of all the training examples. It is numerically computed as:

$$epochs = \frac{\text{batch size} * \text{iterations}}{\text{training images}} \quad (3.1)$$

- **Loss function.** Loss function (also called cost function) evaluates the penalty between the prediction and the ground truth label in every batch.
- **Learning rate.** The learning rate parameter defines the step size for which the weights of a model are updated regarding the stochastic gradient descent.
- **Decay.** The weight decay is an additional weight update parameter that induces the weights to exponentially decay to zero once the update process is over.
- **Optimizer.** Keras framework provides optimizers [58] in order to find the most optimal set of hyperparameters for the model. Some optimizer examples are the SGD, RMSprop and Adam.

3.6.2 Segmentation

The overview of the segmentation task follows the main structure described in Figure 3.5. Training and Validation data from the *Segmentation dataset* is processed with a Python script in order to load the images and masks and convert them into NumPy binary format files (.npy) that will allow a faster loading during the learning and prediction algorithm. Data is previously organized into *train*, *train_masks* and *test* folders.

The proposed model is the U-Net network (Section 3.3.1) trained from scratch, which means that weights are randomly initialized and optimized by backpropagation. The network output is a Numpy array containing the 379 test binary masks, which can be converted into JPG images of 64×80 pixels dimension. This means a posterior post-processing technique will be needed to enlarge the images.

The loss function chosen for the training was a *dice coefficient*.

3.6.3 Classification

The main issue of the classification task is to avoiding overfitting caused by the small number of images of skin lesion in most dermatology datasets. In order to solve this problem, the objective of the proposed model is to firstly extract features from images with the original VGG-16 and secondly load those extracted representations on a fine-tuned VGG-16 architecture.

Due to the reduced size of the ISIC dataset, the suggested approach initializes the model with weights from the VGG-16 trained on a larger dataset (such as ImageNet), a process known as *transfer learning*. The underlying assumption behind transfer learning is that the pre-trained model has already learned features that might be useful for the classification task at hand. This corresponds, in practice, to using selected layer(s) of the pre-trained ConvNet as a fixed feature

extractor, which can be achieved by freezing all the convolutional blocks and only training the fully connected layers with the new dataset.

Another common transfer learning technique consists of not only retraining the classifier on the top of the network with the new dataset, but also applying a fine-tuning of the network by training only the higher-level portion of the convolutional layers and continuing the backpropagation. In this work, it was proposed to freeze the lower level layers of the network because they contain more generic features of the dataset. The interest in training only the top layers of the network comes from the ability to perform extraction of more specific features. For this reason, the first four convolutional layers in the original VGG-16 architecture are initialized with weights from the ImageNet dataset. The fifth, and final, convolutional block is initialized with weights saved during the features extraction.

Due to the reduced nature of the ISIC dataset, it was observed that the VGG presented *overfitting* caused by the few number of images. In order to prevent it, the VGG-16 was chosen in front of the upgraded VGG-19 architecture by reducing the network capacity. Note that the number of parameters to be computed is reduced from 144M parameters (VGG-19) to 138M parameters (VGG-16) [57]. This overfitting issue can be also cut down by adding *dropout*, a network regularization technique that reduces co-adaptation between neurons by increasing iterations for convergence. The last prevention that was applied is data augmentation, previously commented in Section 3.5.

Since the nature of the ISIC dataset is not balanced (approximately 80% corresponds to the benign class, while the rest is referred to the malignant), the network has to be sure that the same number of images for each class are being trained. Instead of manually balancing the dataset through *downsampling*, that it would further reduce the small dimension of the dataset, a weighted loss function [59] (equation 3.2) was proposed in favor of balancing the minor class.

$$H(p, q) = \sum_x \alpha(x) p(x) \log(q(x)), \quad \alpha(x) = \begin{cases} \rho & x = \text{major class} \\ 1 & \text{otherwise} \end{cases} \quad (3.2)$$

where p and q are the probability distributions of the ground truth the ρ value weights the loss regarding the major class instances during training. ρ is defined as $1 - \text{frequency apparence}(\text{minor class})$.

Finally, the process described in this section were repeated for each classification task considered in this project: (1) *Unaltered Classification dataset*, (2) *Perfectly Segmented Classification dataset*, and (3) *Automatically Segmented Classification dataset*.

Chapter 4

Results

This chapter shows and analyzes the results obtained following the guidelines proposed in Chapter 3.

4.1 Experimental setup

The following experiments were performed:

- Experiment 1. Train the semantic segmentation model on the *Segmentation dataset*. Obtain an evaluation of the model and the binary masks as a result.
- Experiment 2. Train the classification model on the *Unaltered lesion classification dataset*. Obtain the evaluation of the model and the predictions done by the model.
- Experiment 3. Train the classification model on the *Perfectly segmented lesion classification dataset*. Obtain the evaluation of the model and the predictions done by the model.
- Experiment 4. Train the classification model on the *Automatically segmented lesion classification dataset*. Obtain the evaluation of the model and the predictions done by the model.

The segmentation model was trained with a 32 batch size for 500 epochs (chosen based on examining the behavior of accuracy/loss plots vs. number of epochs), since just a 13 seconds were needed to train each epoch on NVidia GeForce GTX TITAN X [60] GPU. Weights were randomly initialized and updated by Adam optimizer [61] with a 10^{-5} learning rate.

The hyper-parameters chosen for the classification process were 50 epochs (chosen based on examining the behavior of accuracy/loss plots vs. number of epochs), where each epoch took 35 seconds on NVidia GeForce GTX TITAN X [60] GPU. The fine-tuning process was done with a very small learning rate (10^{-4}) and the SGD optimizer to minimize the loss function.

4.2 Metrics

Main metrics used for the evaluation of this work are the following:

- **Accuracy**, which computes the number of correct predictions divided by the total number of samples.

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{number of samples}} \quad (4.1)$$

- **Dice coefficient.** This metric is computed by comparing the pixel-wise agreement between the groundtruth (Y) and its corresponding predicted segmentation (X). Specially, this metric is just used to evaluation the segmentation model performance.

$$\text{Dice coefficient} = \frac{2 * |X \cap Y|}{|X| + |Y|} \quad (4.2)$$

- **Jaccard index**, also known as the *Jaccard similarity coefficient*, compares predictions (A) with the groundtruth (B) to see which samples are shared and which are distinct. The higher the index, the more similar the two subsets.

$$\text{Jaccard index}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4.3)$$

- **Sensitivity**, also know as *recall*, is computed as the fraction of true positives that are correctly identified.

$$\text{Sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}} \quad (4.4)$$

- **Precision**, which is computed as the fraction of retrieved instances that are relevant.

$$\text{Precision} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false positives}} \quad (4.5)$$

- **Specificity**, computed as the fraction of true negatives that are correctly identified.

$$\text{Specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}} \quad (4.6)$$

4.3 Results

4.3.1 Training the U-Net for Skin Lesion Segmentation

Segmentation results are based on an average computation over each individual testing masks comparison between the automatically created binary mask and its groundtruth mask. Note the comparison is easy to be done since the images are binary (0 or 1 values). The entire evaluation table for each validation image is included in the Appendix C.

Participant	Rank	Accuracy	Dice	Jaccard	Sensitivity	Specificity
MIDDLE group	(1st)	0.9176	0.8689	0.9176	0.9301	0.9544
Urko Sanchez [62]	1st	0.953	0.910	0.843	0.910	0.965
Lequan Yu [63]	2nd	0.949	0.897	0.829	0.911	0.957
Mahmudur Rahman [64]	3rd	0.952	0.895	0.822	0.880	0.969

Table 4.1: Segmentation results benchmarking against the Challenge Top 3

Participants from the ISBI 2016 Challenge [25] were ranked and awards granted based only on the Jaccard index (Classification in Figure 4.3.1). The winner of the Challenge, the participant Urko Sanchez, obtained a the best Jaccard index of 0.843 over the 28 participants on the

segmentation task, which means the model proposed in this work achieves a better result with an index of 0.9176.

Excellent results were also obtained for the sensitivity and specificity metric. In the case of sensitivity, the model approach would obtain the best value in the Challenge; while for the specificity metric would be placed in the top 3 of the leaderboard. Regarding the accuracy metric, the model achieved an accuracy value of 91.76%, which would place the proposed approach in the top 15 in the Challenge.

Figure 4.1 and 4.2 show three examples of what are considered satisfactory results versus not-so-satisfactory results. For each example, the original skin image (left side) is compared to the segmentation proposed (right side) by the deep-learning based approach. In the first case, it can be seen how hairs did not cause much trouble. The second case shows how a low contrast in the image and irregularities can cause a bad segmentation. More examples are provided in Appendix D.

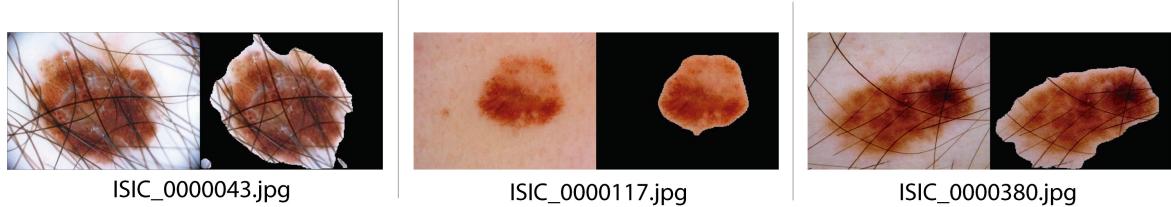


Figure 4.1: Examples of satisfactory segmentation results

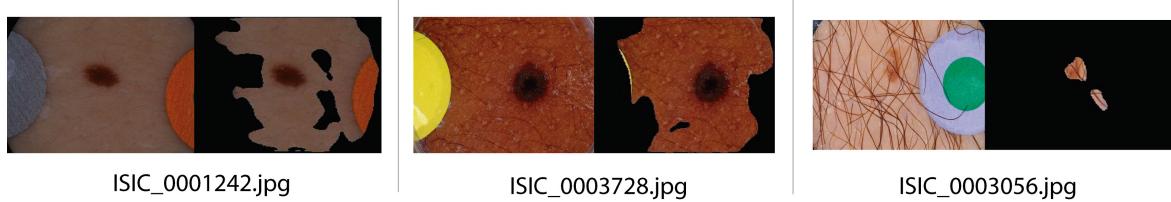


Figure 4.2: Examples of poor segmentation results

4.3.2 Fine-tuning the VGG-16 for Skin Lesion Classification

Results for this task demonstrate that a good performance can be obtained by fine-tuning the VGG-16 model originally trained for image classification on the Imagenet dataset [48]. This fact implies the proposed model can generalize well to a wide variety of classification problems [65] even including situations where the class to be classified is not present in the Imagenet dataset.

In the context of medical images, sensitivity refers to the percent of true positives (malignant lesions) that are correctly identified whereas specificity measures how many samples predicted as benign (negative) are actually so. Results (shown in Table 4.3.2, with best values are highlighted in **bold**) explains that segmentation improves sensitivity results (89.18%) in comparison to unsegmented images classification (82.43% for the unaltered images) and even for the case of perfectly segmented (86.48%), which is a good indicator of the quality of the predictions made by the segmentation model. Moreover, for the sake of comparison, the sensitivity of the best

proposed model is far superior to the values reported by all other contestants in the ISBI 2016 challenge[24]. Precision metric shows the same situation as sensitivity, achieving the best value on the automatically segmented classifier with 96.81%.

Nevertheless, best accuracy and loss value are obtained with the unaltered lesion classifier (84.69% and 0.4723), followed by the perfectly segmented classifier (83.90% and 0.4958).

	Accuracy	Loss	Sensitivity	Precision
Unaltered lesion classification	0.8469	0.4723	0.8243	0.9523
Perfectly segmented lesion classification	0.8390	0.4958	0.8648	0.9621
Automatically segmented lesion class.	0.8174	0.5144	0.8918	0.9681

Table 4.2: Classification results comparison

Confusion matrices, usually called *matching matrices*, depicted in Figure 4.3, show the performance of each classification model on the validation set. The most decisive quadrant to be considered in the evaluation of each model is the top-right quadrant. It is related to the number of true melanomas classified as benign cases. Note this is the wrong case scenario, where the model misses a real case of melanoma that could end up in a dangerous and potentially fatal situation for the patient. With reference to this False Negative (FN) value, automatic segmentation model (whose confusion matrix appears in the right side of the Figure) showed the best performance of all with just 8 FN cases over the 379 total samples, followed by the perfectly segmentation classifier (10 FN cases) and the unaltered classifier (13 FN cases). This outcome reaffirms one assumption from the hypothesis of this work, which subscribed that a previous segmentation of the skin lesion improves the sensitivity of the model. The assumption related to the accuracy improvement by segmenting the lesion has been finally rejected.

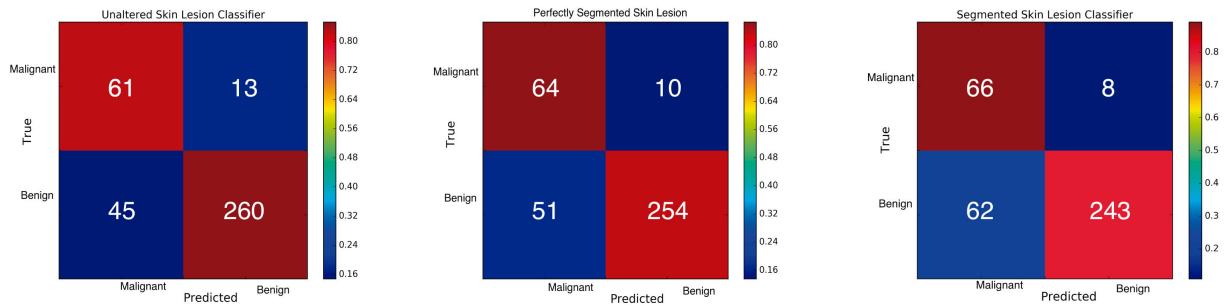


Figure 4.3: Confusion Matrices

Finally, precision & recall and ROC curves (Figures 4.4, 4.5 and 4.6) have been attached at the end of this section to evaluate the performance of the binary classification. Similar behavior is observed for the three different cases: the third case maintains the best result from Table 4.3.2.

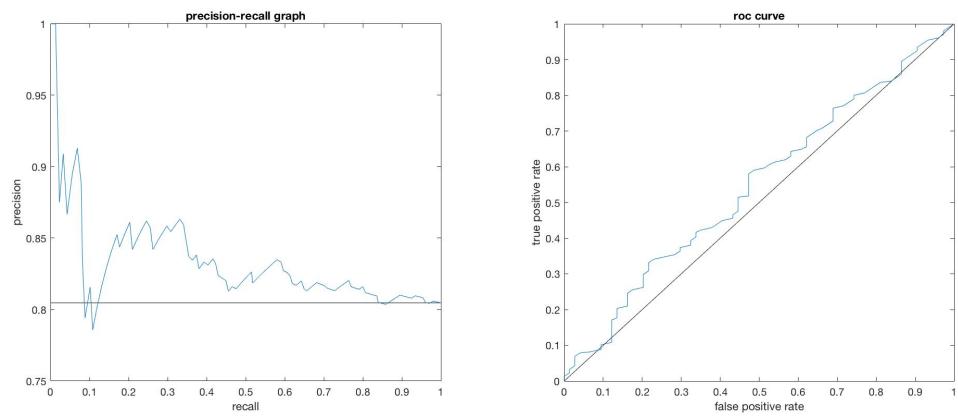


Figure 4.4: Unaltered lesion classification: Precision & Recall and ROC curve

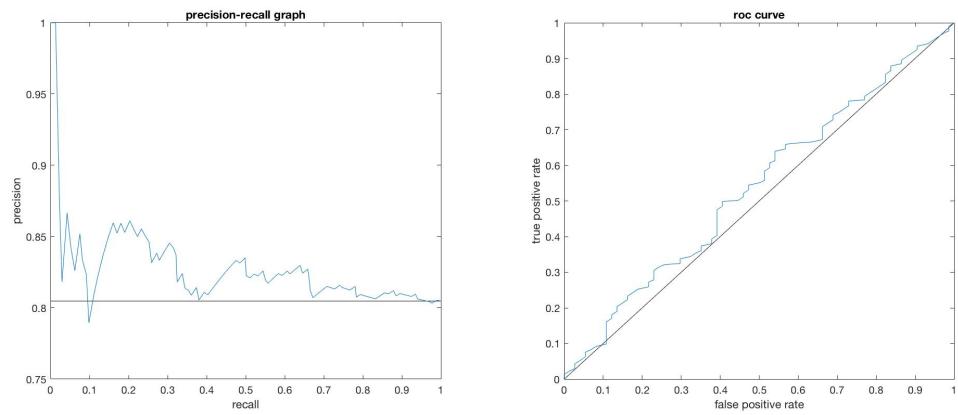


Figure 4.5: Perfectly segmented lesion classification: Precision & Recall and ROC curve

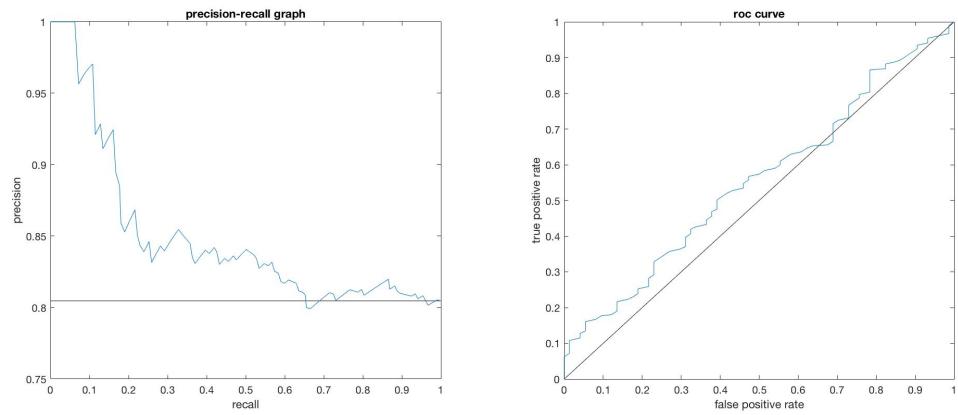


Figure 4.6: Automatically segmented lesion classification: Precision & Recall and ROC curve

Figure 4.7 shows representative examples of prediction errors made by the automatic segmentation classifier (false negatives and false positives, respectively) and examples of correct prediction results (benign and malignant, respectively).

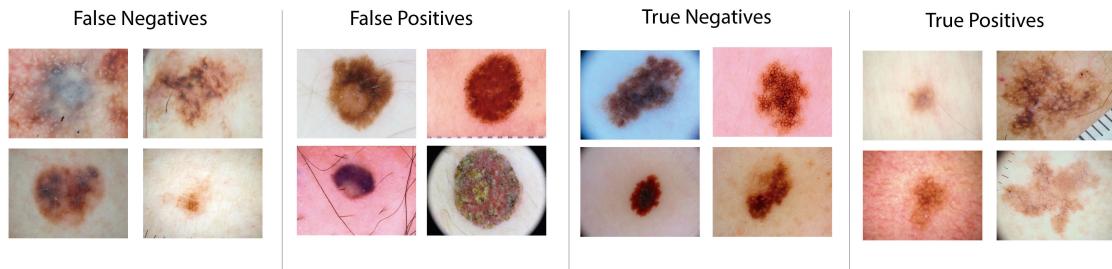


Figure 4.7: Examples of True and False predictions done by the automatically segmented lesion model

Chapter 5

Budget

Despite this project has been developed using the resources provided by Image Processing Group of UPC in the form of GPU power, monthly costs of deploying this solution has been computed based on the Amazon Web Services (AWS) [66] calculator: \$1,873.30 (1,768.52 €) using the *g2.8xlarge* instance [67].

The major costs of the project comes from the salary of the researches and their dedication on it. The positions considered for the budget are referred to the MIDDLE research group: me, the author of this work, I consider myself as a full-time dedication junior engineer, Jack Burdick is a part-time junior engineer, while the two advisors (Dr. Oge Marques and Dr. Xavier Giró-i-Nieto) of this work are considered senior engineer.

The total duration of the project was considered of 25 weeks, as illustrated in the Gantt diagram of Figure 1.4.

	Amount	Wage/hour	Dedication	Total
Junior engineer full-time	1	8.00 €/h	48 h/week	9,600 €
Junior engineer part-time	1	8.00 €/h	30 h/week	6,000 €
Senior engineer	2	20.00 €/h	3 h/week	1,500 €
GPU costs	1	\$12.482 /h	6 h/week	1,768.52 €
			Total	18,869 €

Table 5.1: Budget of the project

Chapter 6

Conclusions

This project has proposed a *Deep Learning* solution for assisting dermatologists during the diagnosis of skin lesions. More specifically, it has investigated how a previous semantic segmentation of the skin lesion improves the performance of a fine-tuned convolutional neural network model approach for a 2-class classifier for early melanoma detection.

The proposed segmentation model trains from scratch the U-Net, a ConvNet for Biomedical Image Segmentation, in order to obtain binary masks that will produce the automatically segmented skin image. The proposed approach achieves promising results, most notably, a Jaccard value of 91.76%, which is significantly higher than the current state of the art on the ISBI 2016 Challenge.

The classification model accepts input skin lesion images labeled as benign or malignant, and it predicts whether a previously unseen image of a skin lesion is either benign or malignant. When classifying (i) the original ISIC dataset, (ii) the perfectly segmented ISIC dataset and (iii) the automatically segmented ISIC dataset by fine-tuning the VGG-16 network, segmentation did not significantly improve classification performance when strictly measuring accuracy. Indeed, validation performance even suffered when segmentation was applied. However, accuracy is not the full story and may not be the best measure of performance in a medical setting. Sensitivity is often considered a more important metric in the medical setting. In these situations, where early diagnosis is of great importance, it is better to raise a false positive than to create a false negative – the project would rather be overly pessimistic than optimistic in its prediction. Considering sensitivity, automatically segmenting an image prior to classification produces better results than segmenting the image perfectly nor without segmenting an image.

Using state of the art methods to classify images, the automatically segmented images achieved the best results with 89.18% sensitivity and 81.74% accuracy, the perfectly segmented images obtained 86.48% sensitivity and 83.90%, while the the accuracy improves to 84.69% and the sensitivity decreases 82.43% when using unsegmented images. The precision is recorded at 96.81% when using automatically segmented images and decreases to 95.23% when using unsegmented images. The comparison of the three approaches concludes that segmentation is needed, but not necessarily perfect (avoiding manual and time-consuming segmentation tasks done by physicians). It can be explained as the perfect segmentation may lose information of the lesion borders (kept intact when automatically segmenting) that could be contextually relevant to the CNN.

In future work, tweaks in regularization [68] (stronger L2 weight penalty, higher dropout (>0.5)) could also be applied. Moreover, weights could be saved and loaded from the architecture being trained with Dermnet [69], a larger and skin related dataset, rather than Imagenet, a general dataset, that would also help lessen the risk of overfitting. Further work could also explore the performance of Residual Neural Networks [70], that have recently performed excellent results on image classification tasks by proposing substantially deeper and easier to optimize networks.

The MIDDLE research group participated in the upcoming 13th IASTED [71] International Conference on Biomedical Engineering [72], co-sponsored by the IEEE Engineering in Medicine



and Biology Society [73], in Innsbruck (Austria), February 20-21, 2017, with a long paper [74] (of which I am the main author), attached in the Appendix A, and a tutorial on "Medical image analysis using deep learning" by my supervisor.

The MIDDLE research group has also submitted an abstract (in which I am a co-author), attached in the Appendix B, to the SIIM (Society for Imaging Informatics in Medicine) [75] 2017 Annual Meeting [76], which is currently under review.

The implementation of this project is publicly available on: https://github.com/iamrosmarin/BSc_Thesis_Skin_Lesion_Detection



Appendix A

BioMed 2017 paper

Appendix A includes the *BioMed 2017* paper.

SKIN LESION CLASSIFICATION FROM DERMOSCOPIC IMAGES USING DEEP LEARNING TECHNIQUES

Adria Romero Lopez, Xavier Giro-i-Nieto

Universitat Politecnica de Catalunya
Barcelona, Catalunya, Spain

{adria.romero@alu-etsetb., xavier.giro@ }upc.edu

Jack Burdick, Oge Marques

Florida Atlantic University
Boca Raton, FL, USA

{jburdick2015, omarques}@fau.edu

ABSTRACT

The recent emergence of deep learning methods for medical image analysis has enabled the development of intelligent medical imaging-based diagnosis systems that can assist the human expert in making better decisions about a patient's health. In this paper we focus on the problem of skin lesion classification, particularly early melanoma detection, and present a deep-learning based approach to solve the problem of classifying a dermoscopic image containing a skin lesion as malignant or benign. The proposed solution is built around the VGGNet convolutional neural network architecture and uses the transfer learning paradigm. Experimental results are encouraging: on the ISIC Archive dataset, the proposed method achieves a sensitivity value of 78.66%, which is significantly higher than the current state of the art on that dataset.

KEY WORDS

Medical Image Analysis, Deep Learning, Medical Decision Support Systems, Convolutional Neural Networks, Machine Learning, Skin Lesions

1 Introduction

Melanoma is a fatal form of skin cancer which is often undiagnosed or misdiagnosed as a benign skin lesion. There are an estimated 76,380 new cases of melanoma and an estimated 6,750 deaths each year in the United States [1]. Early detection is imperative: the lives of melanoma patients depend on accurate and early diagnosis. Physicians often rely on personal experience and evaluate each patient's lesions on a case-by-case basis by taking into account the patient's local lesion patterns in comparison to that of the entire body [2].

Without computer-based assistance, the clinical diagnosis accuracy for melanoma detection is reported to be between 65 and 80% [3]. Use of dermoscopic images improves diagnostic accuracy of skin lesions by 49% [4]. However, the visual differences between melanoma and benign skin lesions can be very subtle (Figure 1), making it difficult to distinguish the two cases, even for trained medical experts.

For the reasons described above, an intelligent medical imaging-based skin lesion diagnosis system can be a welcome tool to assist a physician in classifying skin le-

sions. In this work, we are interested in a specific two-class classification problem, namely: determine whether a dermoscopic image containing a skin lesion contains a melanoma or a benign lesion.

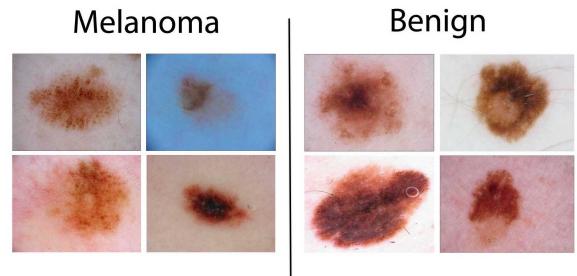


Figure 1: Sample images created from the ISIC Archive dataset [5]

In this paper, a novel method for skin lesion classification using deep learning is proposed, implemented, and successfully benchmarked against a publicly available skin lesion dermoscopic image dataset (the ISIC Archive dataset [5]). It uses an existing convolutional neural network (CNN) architecture – VGGNet (Very Deep Convolutional Network for Large-Scale Visual Recognition) developed by the Visual Geometry Group of the University of Oxford [6] in three different ways: (i) training the CNN from scratch; (ii) using the transfer learning paradigm to leverage features from a VGGNet pre-trained on a larger dataset (ImageNet [7]); and (iii) keeping the transfer learning paradigm and fine-tuning the CNNs architecture. This paper is structured as follows: Section 2 reviews related work and associated datasets and challenges; Section 3 described the proposed solution (and its variants) and associated methods and tools; Section 4 presents the results of experiments and discusses their significance; finally, Section 5 offers concluding remarks and directions for future work.

2 Background

In this section we provide a summary of relevant recent work in this field, as well as associated datasets and challenges.

2.1 Related work

Most current methods in the field of melanoma classification rely on hand-crafted features, such as: lesion type (primary morphology), lesion configuration (secondary morphology), color, distribution, shape, texture, and border irregularity measures [8]. After feature extraction, machine learning methods such as k-nearest neighbors (kNN), Artificial Neural Networks (ANNs), logistic regression, decision trees and support vector machines (SVMs) can be used to perform the classification task with moderate success [9]. Examples of related work using hand-crafted features and popular classifiers include:

- Codella et al. [10] utilize hand-coded feature extraction techniques including color histogram, edge histogram, and a multi-scale variant of color local binary patterns (LBP).
- The approach proposed by Barata et al. [11] utilizes two different methods for the detection of melanoma in dermoscopy images based on global and local features. The global method uses segmentation and wavelets, Laplacian pyramids or linear filters followed by a gradient histogram are used to extract features such as texture, shape, and color from the entire lesion. After that, a binary classifier is trained from the data. The second method of local features uses a Bag of Features (BoF) classifier for image processing tasks (i.e. object recognition). Barata et al. conclude that color features perform much better than texture features alone.

More recently, the emergence of a machine learning paradigm known as deep learning has enabled the development of medical image analysis systems that can display remarkable accuracy, to the point of raising concerns about the future of the human radiologist [12][13]. Convolutional neural networks have produced promising results when classifying skin lesions. Examples of related work using deep learning include:

- The work of Kawahara et al. [14] explores the idea of using a pretrained ConvNet as a feature extractor rather than training a CNN from scratch. Furthermore, it demonstrates the use filters from a CNN pretrained on natural images generalize to classifying 10 classes of non-dermoscopic skin images.
- Liao's [15] work attempted to construct a universal skin disease classification by applying transfer learning on a deep CNN and fine-tuned its weights by continuing the backpropagation.
- In Codella et al. [10], the authors report new state-of-the-art performance using ConvNets to extract image descriptors by using a pre-trained model from the Image Large Scale Visual Recognition Challenge (ILSVRC) 2012 dataset [7]. They also investigate the

most recent network structure to win the ImageNet recognition challenge called Deep Residual Network (DRN) [16].

2.2 Datasets and challenges

There are relatively few datasets in the general field of dermatology and even fewer datasets of skin lesion images. Moreover, most of these datasets are too small and/or not publicly available, which provides an additional obstacle to performing reproducible research in the area. Examples of dermatology-related image datasets used in recent research include:

- Dermofit Image Library [17] is a dataset that contains 1,300 high quality skin lesion images collected across 10 different classes.
- Dermnet [18] is a skin disease atlas with website support that contains over 23,000 skin images separated into 23 classes.

In the beginning of 2016, the International Symposium on Biomedical Imaging (ISBI) [19] released a challenge dataset for Skin lesion analysis towards melanoma detection. Photos in this dataset were obtained from the ISIC (International Skin Imaging Collaboration) [5].

3 Proposed solution

In this section we describe the selected convolutional network (ConvNet) architecture and discuss associated design choices and implementation aspects.

3.1 ConvNet architecture

VGGNet is a well documented and commonly used architecture for convolutional neural networks [6]. This ConvNet became popular by achieving excellent performance on the ImageNet [7] dataset. It comes in several variations of which the two best-performing (with 16 and 19 weight layers) have been made publicly available. In this work, the VGG16 architecture (Figure 2) was selected, since it has been shown to generalize well to other datasets. The input layer of the network expects a 224×224 pixel RGB image. The input image is passed through five convolutional blocks. Small convolutional filters with a receptive field of 3×3 are used. Each convolutional block includes a 2D convolution layer operation (the number of filters changes between blocks). All hidden layers are equipped with a ReLU (Rectified Linear Unit) as the activation function layer (nonlinearity operation) and include spatial pooling through use of a max-pooling layer. The network is concluded with a classifier block consisting of three Fully-Connected (FC) layers.

3.2 Design considerations

The original VGG16 must be modified to suit our needs, as follows:

- The final fully-connected output layer must perform a binary classification (benign vs. malignant), not 1000 classes.
- The activation function in the modified layer is modified from a softmax to sigmoidal.

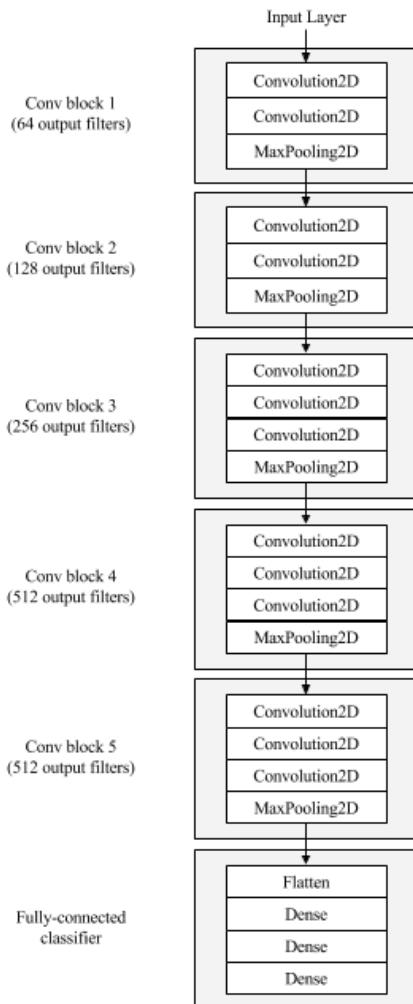


Figure 2: Original VGG16 architecture (adapted from [6])

3.2.1 Preprocessing

Input images must be preprocessed by: (i) normalizing the pixel values to a $[0,1]$ range; (ii) cropping the image to square aspect ratio (if necessary); and (iii) resizing the image to the expected size of 224×224 pixels.

3.2.2 Data augmentation

In order to make the most of our few training examples and increase the accuracy of the model, we augmented the data via a number of random transformations. The selected data augmentation techniques were: size re-scaling, rotations of 40° , horizontal shift, image zooming, and horizontal flipping. Furthermore, it is expected that data augmentation should also help prevent overfitting (a common problem with small datasets, when the model, exposed to too few examples, learns patterns that do not generalize to new data) and, for this reason, improving the models ability to generalize.

3.3 One problem, three possible solutions

The modified VGG16 ConvNet can be used in three different ways: (i) training the ConvNet from scratch; (ii) using the transfer learning paradigm to leverage features from a pre-trained VGG16 on a larger dataset; and (iii) keeping the transfer learning paradigm and fine-tuning the ConvNets architecture. These variants (named *Method 1*, *Method 2*, and *Method 3*, respectively) are described next.

3.3.1 Method 1 - Training from scratch

The architecture is initialized with random weights and trained for a number of epochs. After each epoch, the model learns features from data and computes weights through backpropagation. This method is unlikely to produce the most accurate results if the dataset is not significantly large. However, it still can serve as a baseline for comparison against the two other methods.

3.3.2 Method 2 - ConvNet as feature extractor

Due to the relatively small number of images of skin lesion in most dermatology datasets, this method initializes the model with weights from the VGG16 trained on a larger dataset (such as ImageNet [7]), a process known as *transfer learning*. The underlying assumption behind transfer learning is that the pre-trained model has already learned features that might be useful for the classification task at hand. This corresponds, in practice, to using selected layer(s) of the pre-trained ConvNet as a fixed feature extractor, which can be achieved by freezing all the convolutional blocks and only training the fully connected layers with the new dataset.

3.3.3 Method 3 - Fine-tuning the ConvNet

Another common transfer learning technique consists of not only retraining the classifier on the top of the network with the new dataset, but also applying a fine-tuning of the network by training only the higher-level portion of the convolutional layers and continuing the backpropagation. In this work, we propose to freeze the lower level layers

of the network because they contain more generic features of the dataset. We are interested in training only the top layers of the network due to their ability to perform extraction of more specific features. In this method, the first four convolutional layers in the final architecture are initialized with weights from the ImageNet dataset. The fifth, and final, convolutional block is initialized with weights saved and loaded from the corresponding convolutional layer in Method 1.

3.3.4 Implementation aspects

Keras [20], a deep learning framework for Python, was utilized to implement the neural network architecture. Keras provides a layer of abstraction on top of Theano [21], which is used as the main neural network framework. Keras allows for: (1) modularity: users can create their network following a sequence which is a linear stack of layers; (2) minimalism: functions included in the library allow the user to create and modify network layers easily; and (3) extensibility: daily updates provide solutions to ongoing challenges faced by deep learning researchers. Moreover, Keras works on a Python environment, which gives users the freedom to use additional Python dependencies, including SciPy [22] and PIL [23].

In addition to Keras, CUDA libraries [24] were required to drive the NVidia GeForce GTX TITAN X GPUs (Graphics Processing Units) used to train and evaluate the implementation [25].

4 Experiments and Results

This section discusses the results of experiments using the proposed methods and the selected implementation.

4.1 Dataset

The ISBI 2016 Challenge dataset for Skin Lesion Analysis towards melanoma detection (described in Section 2.2) was used for our experiments.

The dataset contains a representative mix of images labeled as benign or malignant, pre-partitioned into sets of 900 training images and 379 test images [5].

4.2 Parameters

All methods were implemented in Keras. The optimizing function is RMSProp [26]. The loss function is described in [27]. A value of 0.5 is used for a dropout optimization in the fully connected layers. A batch size of 16 images is selected due to the small size of our dataset.

The dataset is balanced through undersampling. Listed alphabetically, the first 173 images from each class in the training dataset were selected and the first 75 images

in each class from the testing dataset were selected. In total, the final dataset was composed of 346 training images and 150 testing images.

For data augmentation we used the following variations: size re-scaling, rotations (angles), horizontal shift, zooming (factor), and horizontal flipping.

4.3 Results

The model evaluation is performed using the same training and testing partition used in the ISIC dataset.

The metrics used are:

- *loss*, defined as the quantification of the agreement between the predicted images and the groundtruth labels;
- *sensitivity*, the fraction of true positives that are correctly identified;
- *precision*, the fraction of retrieved instances that are relevant;
- *specificity*, the fraction of true negatives that are correctly identified; and
- *accuracy*, the number of correct predictions divided by the total number of predictions.

The number of epochs for each method (chosen based on examining the behavior of accuracy/loss plots vs. number of epochs) was: 20 epochs for Method 1, 50 epochs for Method 2, and 20 epochs for Method 3.

Training and testing results for each method are shown in Tables 1 and 2, respectively. Best values are highlighted (in **bold**).

Table 1: Model evaluation: training dataset

	Loss	Accuracy	Sensitivity	Precision
M1	0.5637	71.87%	0.7087	0.6990
M2	0.1203	95.95%	0.9621	0.9560
M3	0.4891	76.88%	0.6903	0.8259

Table 2: Model evaluation: test dataset

	Loss	Accuracy	Sensitivity	Precision
M1	0.6743	66.00%	0.5799	0.6777
M2	1.0306	68.67%	0.3311	0.4958
M3	0.4337	81.33%	0.7866	0.7974

Figure 3 shows representative examples of prediction errors made by the classifier (false positives and false negatives, respectively). For contrast, Figure 4 shows examples of correct prediction results (malignant and benign, respectively).

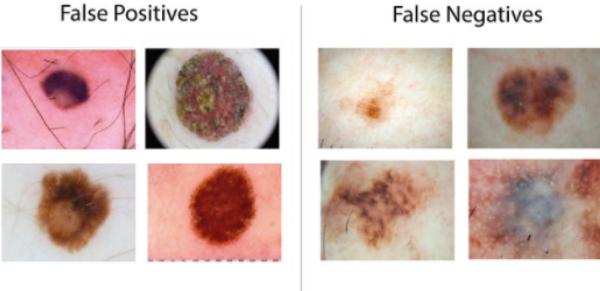


Figure 3: Examples of False Positives and False Negatives

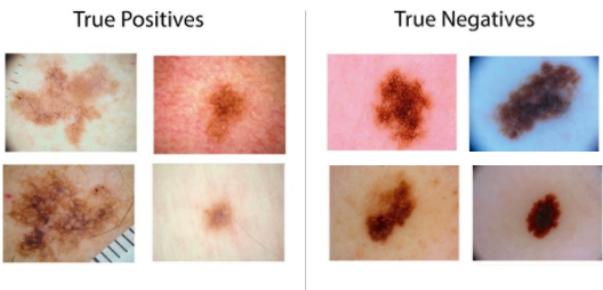


Figure 4: Examples of True Positives and True Negatives

4.4 Discussion

The results obtained using the first method are acceptable and the accuracy value is substantially above chance. Moreover, the minor difference between training and test sets suggests that the model neither overfits nor underfits.

Using the results for Method 1 as a baseline, it was not surprising to notice that Method 3 produced superior results for loss, accuracy, sensitivity, and precision in the test data. The performance of Method 2 – with exceptionally good numbers for the training set and the worst combination of loss, sensitivity, and accuracy values for the test set – presents a classical example of overfitting.

Transfer learning offers the benefit of faster epoch processing times since the layers are frozen and loaded from a previously trained network. Though decreasing processing time is preferred, the trade off is a result of the learned features potentially being unrelated to intended classification task. This observation could explain, to some extent, the inferior results of Method 2, since the ImageNet dataset is trained on 15 million labeled high resolution images from 22,000 different categories.

<i>True</i>	Malignant	59	16
	Benign	12	63
		Malignant	Benign
			<i>Predicted</i>

Figure 5: Confusion Matrix for Method 3

Method 3 (whose confusion matrix appears in Figure 5) showed the best performance of all, due to reduced dependency on the ImageNet initialization weights. When using the dataset partitioned identically to the ISBI 2016 Challenge [19], it achieved an accuracy value of 81.33%, which would place the proposed approach in the top three in that challenge. Most importantly, when considering sensitivity, we achieve 78.66%, a result that is significantly better than the one reported by the current leader (50.7%). Our precision value, 79.74%, is also superior to the the current best result, of 63.7%.

In the context of medical images, *sensitivity* refers to the percent of true positives (malignant lesions) that are correctly identified whereas *specificity* measures how many samples predicted as benign (negative) are actually so. Our results for Method 3 (78.66% for sensitivity and 84.00% for specificity) are good indicators of the quality of the predictions made by the proposed model.

5 Conclusion

We propose a solution for assisting dermatologists during the diagnosis of skin lesions. More specifically, we have designed and implemented a two-class classifier that takes skin lesion images labeled as benign or malignant as an input, builds a model using deep convolutional neural networks, and uses this model to predict whether a (previously unseen) image of a skin lesion is either benign or malignant. The proposed approach achieves promising results – most notably, a sensitivity value of 78.66% and a precision of 79.74% – which are significantly higher than the current state of the art on this dataset (50.7% and 63.7%, respectively).

Avenues for future work include: (i) using a larger dataset to help lessen the risk of overfitting; (ii) performing additional regularization tweaks and fine-tuning of hyperparameters; and (iii) training the architecture with Dermnet – a skin related dataset – rather than Imagenet, a general dataset.

Acknowledgements

The authors gratefully acknowledge funding from NSF Award No. 1464537, Industry/University Cooperative Research Center, Phase II under NSF 13-542. We are also thankful to the 33 corporations that are the members of the Center for their active participation and funding.

We also thank Janet Weinthal (FAU) for her insights and assistance during the preparation of this manuscript.

References

- [1] R. L. Siegel, K. D. Miller, and A. Jemal, “Cancer statistics, 2016,” *CA: a cancer journal for clinicians*, vol. 66, no. 1, pp. 7–30, 2016.
- [2] J. Gachon, P. Beaulieu, J. F. Sei, J. Gouvernet, J. P. Claudel, M. Lemaitre, M. A. Richard, and J. J. Grob, “First prospective study of the recognition process of melanoma in dermatological practice,” *Archives of dermatology*, vol. 141, no. 4, pp. 434–438, 2005.
- [3] G. Argenziano and H. P. Soyer, “Dermoscopy of pigmented skin lesions—a valuable tool for early diagnosis of melanoma,” *The Lancet Oncology*, vol. 2, no. 7, pp. 443–449, 2001.
- [4] H. Kittler, H. Pehamberger, K. Wolff, and M. Binder, “Diagnostic accuracy of dermoscopy,” *The lancet oncology*, vol. 3, no. 3, pp. 159–165, 2002.
- [5] “International Skin Imaging Collaboration: Melanoma Project Website,” <https://isic-archive.com/>.
- [6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [8] E. H. Page, “Description of skin lesions,” <https://goo.gl/m9ybFp>.
- [9] S. Dreiseitl, L. Ohno-Machado, H. Kittler, S. Vintherbo, H. Billhardt, and M. Binder, “A comparison of machine learning methods for the diagnosis of pigmented skin lesions,” *Journal of biomedical informatics*, vol. 34, no. 1, pp. 28–36, 2001.
- [10] N. Codella, Q.-B. Nguyen, S. Pankanti, D. Gutman, B. Helba, A. Halpern, and J. R. Smith, “Deep learning ensembles for melanoma recognition in dermoscopy images,” *arXiv preprint arXiv:1610.04662*, 2016.
- [11] C. Barata, M. Ruela, M. Francisco, T. Mendonça, and J. S. Marques, “Two systems for the detection of melanomas in dermoscopy images using texture and color features,” *IEEE Systems Journal*, vol. 8, no. 3, pp. 965–979, 2014.
- [12] M. Walter, “Is this the end? machine learning and 2 other threats to radiology’s future,” goo.gl/M9X3SF, 2016.
- [13] S. Jha, “Will computers replace radiologists?” <http://www.medscape.com/viewarticle/863127>, 2016.
- [14] J. Kawahara, A. BenTaieb, and G. Hamarneh, “Deep features to classify skin lesions,” *IEEE International Symposium on Biomedical Imaging (IEEE ISBI)*, pp. 1397–1400.
- [15] H. Liao, “A deep learning approach to universal skin disease classification,” https://www.cs.rochester.edu/~hliao6/projects/other/skin_project_report.pdf.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.
- [17] “Dermofit image library,” <https://licensing.eri.ed.ac.uk/i/software/dermofit-image-library.html>.
- [18] “Dermnet - skin disease atlas,” <http://www.dermnet.com/>.
- [19] “IEEE International Symposium on Biomedical Imaging,” <http://biomedicalimaging.org/>.
- [20] “Keras documentation,” <https://keras.io/>.
- [21] “Theano 0.8.2. documentation,” <http://deeplearning.net/software/theano/>.
- [22] “Scipy Python Library,” <https://www.scipy.org/>.
- [23] “Python Imaging Library (PIL),” <http://www.pythonware.com/products/pil/>.
- [24] “CUDA, Nvidia,” http://www.nvidia.com/object/cuda_home_new.html.
- [25] “Nvidia GeForce GTX TITAN X,” <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x>.
- [26] “RMSProp Optimizer,” <https://keras.io/optimizers/#rmsprop>.
- [27] “Usage of objectives,” <https://keras.io/objectives/>.



Appendix B

SIIM 2017 abstract

Appendix B includes the presented *SIIM 2017* abstract.

THE IMPACT OF SEGMENTATION ON THE ACCURACY AND SENSITIVITY OF A MELANOMA CLASSIFIER BASED ON SKIN LESION IMAGES

Jack Burdick, Oge Marques
Florida Atlantic University
Boca Raton, FL, USA
{jburdick2015, omarques}@fau.edu

Adria Romero Lopez, Xavier Giro-i-Nieto
Universitat Politecnica de Catalunya
Barcelona, Catalunya, Spain
{adria.romero@alu-etsetb., xavier.giro@}upc.edu

HYPOTHESIS

The accuracy and sensitivity of a Deep Learning based approach for a 2-class classifier for early melanoma detection based on skin lesion dermoscopic images increases when the classifier is trained with segmented inputs (i.e., images containing only the lesions as binary masks, without the surrounding context) instead of entire images.

1 Introduction

In the United States alone, there were an estimated 76,380 new cases of melanoma and an estimated 6,750 deaths due to melanoma in 2016 [1]. Early screening can increase life expectancy [2], but undiagnosed melanoma can be fatal. Dermatologists use many heuristic classification methods to diagnose melanoma [3][4], but to limited success with only 65 - 80% accuracy [5]. A tool capable of aiding physicians to classify skin lesions could potentially save numerous lives each year.

Deep neural networks (DNNs) are a promising solution. DNNs are reportedly capable of classifying dermoscopic images with an accuracy of 92.3%. Typically, segmentation is used as a preprocessing method in the classification process to remove potentially non-relevant information from the classification process [6]. Through segmentation, all pixel values, other than those contained within the lesion, are discarded so that only the lesion values are considered when training the network.

Ostensibly, segmenting the skin lesion will remove nonessential information, such as hair or non-target surrounding lesions, and aid in accurately classifying the lesion. However, segmentation may lose information that could be contextually relevant to the DNN. The impact and tradeoffs of segmenting this extra information is not entirely understood.

The hypothesis that segmentation is beneficial when classifying dermoscopic medical images should be investigated. By isolating and comparing results from both unaltered and segmented skin lesion images, we aim to better understand the impact on performance results when segmenting an image. Specifically, we hope to better understand whether the values outside the lesion are detrimental

to lesion classification, or are instead beneficial to lesion classification by providing contextual information relevant to each lesion. Segmenting images using state of the art techniques provides promising but not perfect results and has yet to be proven to be robust across different datasets. Understanding the role segmentation plays in the classification of skin lesions will help to determine whether segmenting images is a necessary task. To test this hypothesis, we will compare the performance metrics of classification performed on segmented and unaltered dermoscopic images under otherwise constant conditions.

2 Methods

The VGGNet [7], a convolutional network, attained popularity after achieving excellent results on the ImageNet dataset in the ILSVRC-2014 competition [8]. In this work, the VGG-16 architecture, shown in Figure 1, which has fewer parameters than the other VGG ConvNet configurations, was chosen since it has been shown to generalize well to other datasets [7].

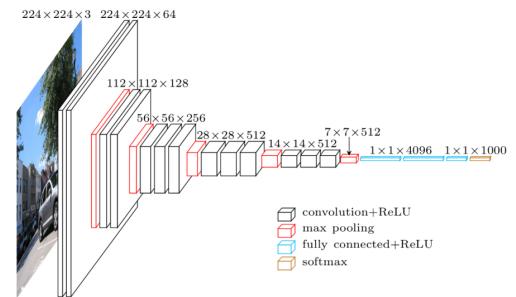


Figure 1: VGG16 architecture [9]

The VGG architecture is constructed with five convolutional blocks and is concluded with a softmax classifier block consisting of three fully-connected layers. The network expects a 224x224 pixel RGB input image which is obtained using preprocessing methods outlined below. An image crop is sometimes necessary to preserve the aspect ratio, as asymmetry and overall shape are often considered important features by classical classification methods [10].

The experiment uses techniques described to give state-of-the-art results [11]. The first three layers in the con-

volutional block are initialized with weights obtained from training the network on the ImageNet dataset while the remaining layers are initialized following a uniform distribution (Uniform Initialization). This transfer learning technique leverages previously learned features from a larger, separate, dataset.

Input images are preprocessed to be accepted by the architecture though: (i) normalizing the pixel values from [0,255] to [0,1] (ii) cropping the image to the same aspect ratio as needed, and (iii) resizing the original image to 224x224 pixels.

Input images, with and without segmentation applied, are created from the dataset. The unsegmented images were used in their unaltered form from the dataset. The perfectly segmented image was generated by performing a bitwise and operation on the unaltered image and its corresponding mask. The resulting photo is a perfectly segmented skin lesion where the values outside the lesion are converted to a zero value. Figure 2 shows an example of an unaltered image (left), a binary mask (center), and the result of the bitwise and operation (right).



Figure 2: Comparison between the raw image (left), binary mask (center) and the perfect segmentation (right) [12]

3 Results

The dataset used is publicly available and contains 1279 images that are pre-partitioned into 900 training images and 379 testing images (International Skin Imaging Collaboration). Due to the unbalanced nature of the dataset, the testing and training datasets were reduced through down sampling to produce an equivalent number of images for each class. The final training set contains 346 images and the final testing set contains 150 images. All images are labeled as either benign or malignant and include a binary image mask to label the lesion within the image.

In order to ensure the architecture produces the same results after each epoch, the random number generator was seeded with an arbitrary value that remained constant throughout testing. The input images were the only parameter modified between sets.

RMSProp [13] and binary cross entropy [14] were implemented as the optimizing and loss function, respectively. The fully connected layers included a 0.5 value for the dropout optimization. A batch size of 16 was used since the small dimension of our dataset. A total of 20 epochs were performed.

Data augmentation techniques including size rescaling, rotations (angles), horizontal shift, zooming (factor), and horizontal flipping were utilized.

Unaltered Skin Lesion Classification			
Loss	Accuracy	Sensitivity	AUC
0.5999	81.33%	84%	0.8097

Table 1: Unaltered Skin Lesion Classification results

Perfectly Segmented Skin Lesion Classification			
Loss	Accuracy	Sensitivity	AUC
0.6309	76.66%	85.33%	0.7898

Table 2: Perfectly Segmented Skin Lesion Classification results

Creating the balanced dataset involved removing images from the testing and training datasets. In the training set, images were removed from the dataset by selectively obtaining the first 173 images from the alphabetically listed directory of each class, for a total of 346 images. The same methodology is performed for the testing dataset and produce 75 benign and malignant images, for a total of 150 images.

Model evaluation was performed using accuracy, sensitivity and loss metrics. Accuracy is defined as the number of correct predictions divided by the total number of predictions made. Sensitivity is defined as the number of true positives divided by the sum of true and false positives and measures the percentage of positives that are correctly identified. Loss is defined as the quantification agreement between the predicted images and the ground truth labels. Accuracy values of 81.33% and 76.66%, sensitivity values of 84% and 85.33% and loss values of 0.5999 and 0.6309, as shown in Table 1 and 2 are produced when using an unaltered and segmented input image, respectively. Area Under Curve (AUC), a quantitative measure that assesses the quality of the ROC curve, is also included. Figure 3 shows the corresponding confusion matrices for both cases; Figure 4 shows sample images corresponding to the confusion matrix.

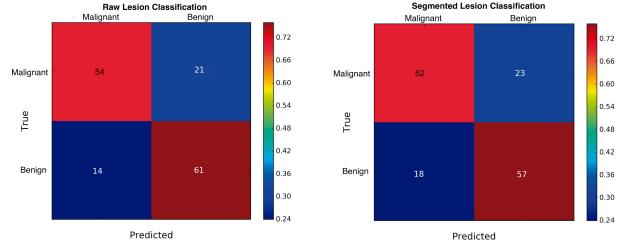


Figure 3: Confusion Matrices

4 Discussion

While sensitivity values increased, accuracy and area under the curve decreased when using perfectly segmented inputs. These results were unexpected and bring to question the impact and potential trade-offs of segmenting dermoscopic images before classification using deep learning



Figure 4: Classification results for the unaltered method

methods.

Skin lesion images often contain background noise that may be detrimental to image classification. A few sample images that have a distracting background are shown in Figure 5 . Ostensibly, the background values serve as distractions that negatively affect lesion classification and must be segmented. However, the values outside of the lesion may not affect classification as adversely as previously thought.

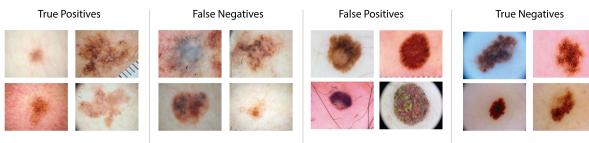


Figure 5: Classification difficulties caused by background noise

When classifying the ISIC dataset with a deep learning approach, segmentation did not significantly improve classification performance when strictly measuring accuracy. Indeed, performance even suffered when segmentation was applied. However, accuracy is not the full story and may not be the best measure of performance in a medical setting. Sensitivity is often considered a more important metric in the medical setting. In these situations, where early diagnosis is of great importance, it is better to raise a false positive than to create a false negative – we would rather be overly pessimistic than optimistic in our prediction. Considering sensitivity, segmenting an image prior to classification produces better results than not segmenting an image.

Segmenting an image before classification, though producing a lower accuracy, may be beneficial in a medical setting because of the increase in sensitivity. The selected dataset is small and despite the indication of the need to segment before classification, the trade-offs of segmentation are not entirely clear. Further research, using additional datasets and classifiers, is needed to determine the full impact of segmenting a lesion image before classifying an image using deep learning techniques.

5 Conclusion

The hypothesis that segmentation aids classification of skin lesions using deep neural networks was investigated. The results of classifying images in the ISIC Archive dataset, when using the same, VGG-16, classifier, show little difference in accuracy and a slight gain in sensitivity when using

perfectly segmented images. Using state of the art methods to classify images, the perfectly segmented images achieve 76.66% accuracy, while the accuracy improves to 81.33% when using unsegmented images. The sensitivity is recorded at 84% when using unsegmented images and increases to 85.33% when using perfectly segmented images.

Future work could involve further testing the hypothesis that segmentation is a necessary preprocessing step to classification when using deep learning techniques. Additional, larger, datasets could be used to help determine if and when segmentation does provide a significant performance increase.

• References:

- [1] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: a cancer journal for clinicians*, 66(1):7–30, 2016.
- [2] Kenneth A Freedberg, Alan C Geller, Donald R Miller, Robert A Lew, and Howard K Koh. Screening for malignant melanoma: a cost-effectiveness analysis. *Journal of the American Academy of Dermatology*, 41(5):738–745, 1999.
- [3] Giuseppe Argenziano, Gabriella Fabbrocini, Paolo Carli, Vincenzo De Giorgi, Elena Sammarco, and Mario Delfino. Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: comparison of the abcd rule of dermatoscopy and a new 7-point checklist based on pattern analysis. *Archives of dermatology*, 134(12):1563–1570, 1998.
- [4] Franz Nachbar, Wilhelm Stolz, Tanja Merkle, Armand B Cognetta, Thomas Vogt, Michael Landthaler, Peter Bilek, Otto Braun-Falco, and Gerd Plewig. The abcd rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, 30(4):551–559, 1994.
- [5] Giuseppe Argenziano and H Peter Soyer. Dermoscopy of pigmented skin lesions—a valuable tool for early. *The Lancet Oncology*, 2(7):443–449, 2001.
- [6] Xiang Li, Ben Aldridge, Lucia Ballerini, Robert Fisher, and Jonathan Rees. Depth data improves skin lesion segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1100–1107. Springer, 2009.
- [7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

- [8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [9] Leonard Blier. A brief report of the heuritch deep learning meetup num. 5, 2016.
- [10] Stephan Dreiseitl, Lucila Ohno-Machado, Harald Kittler, Staal Vinterbo, Holger Billhardt, and Michael Binder. A comparison of machine learning methods for the diagnosis of pigmented skin lesions. *Journal of biomedical informatics*, 34(1):28–36, 2001.
- [11] Jack Burdick Oge Marques Adria Romero Lopez, Xavier Giro-i-Nieto. Skin lesion classification from dermoscopic images using deep learning techniques presented at the iasted 13th international conference on biomedical engineering (biomed 2017), innsbruck, austria, 2021 february 2017, 2016.
- [12] ISIC Archive. International skin imaging collaboration: Melanoma project website. <https://isic-archive.com/>, 2016.
- [13] Rmsprop optimizer. <https://keras.io/optimizers/#rmsprop>.
- [14] Usage of objectives. <https://keras.io/objectives>, 2016.



Appendix C

Segmentation Evaluation

Appendix C includes the evaluation for each segmented image from the ISIC validation dataset.

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0000003	0.953515625	0.967753926	0.953515625	0.998006736	0.990138333
ISIC_0000012	0.900802951	0.568991909	0.900802951	1	0.909689147
ISIC_0000013	0.963585069	0.965243214	0.963585069	0.999252336	0.994102035
ISIC_0000014	0.940277778	0.955360306	0.940277778	1	0.967489966
ISIC_0000015	0.936328125	0.905703289	0.936328125	0.942241464	0.979316783
ISIC_0000020	0.947135417	0.921276312	0.947135417	0.955217992	0.987956013
ISIC_0000022	0.895117188	0.886028857	0.895117188	0.873808443	0.982907388
ISIC_0000023	0.962217882	0.941571214	0.962217882	0.986022364	0.989205839
ISIC_0000027	0.944140625	0.921807747	0.944140625	0.992382617	0.968355982
ISIC_0000036	0.913997396	0.90820033	0.913997396	0.920770257	0.971638033
ISIC_0000037	0.903320313	0.884117191	0.903320313	0.880672989	0.99980154
ISIC_0000040	0.754318576	0.621473177	0.754318576	0.999714408	0.735564565
ISIC_0000043	0.877669271	0.907548301	0.877669271	0.999557106	0.85469955
ISIC_0000052	0.938042535	0.967018447	0.938042535	0.987060726	0.984723642
ISIC_0000053	0.921896701	0.938203802	0.921896701	0.959275408	0.990071188
ISIC_0000056	0.915733507	0.944952752	0.915733507	0.997277887	0.935330462
ISIC_0000057	0.967556424	0.954736898	0.967556424	0.990816935	0.995909515
ISIC_0000064	0.951432292	0.971524186	0.951432292	0.999318569	0.981168242
ISIC_0000066	0.925303819	0.93076045	0.925303819	0.945600072	0.999356055
ISIC_0000069	0.961740451	0.94241913	0.961740451	1	0.981913254
ISIC_0000071	0.930772569	0.79103563	0.930772569	1	0.943998651
ISIC_0000072	0.950802951	0.87354205	0.950802951	0.988118426	0.971828513
ISIC_0000088	0.956119792	0.93187512	0.956119792	0.972738006	0.996639421
ISIC_0000092	0.973133681	0.944089975	0.973133681	1	0.990169923
ISIC_0000098	0.98999566	0.89919865	0.98999566	0.997635934	0.999375042
ISIC_0000099	0.866710069	0.847282286	0.866710069	0.809326366	0.960731516
ISIC_0000101	0.921875	0.956640595	0.921875	1	0.941972706
ISIC_0000107	0.935894097	0.880505082	0.935894097	0.963991271	0.97013073
ISIC_0000111	0.924153646	0.945399906	0.924153646	0.956168165	0.995069554
ISIC_0000113	0.952300347	0.90361719	0.952300347	0.956716609	0.981534397
ISIC_0000115	0.957139757	0.918918919	0.957139757	0.955178416	0.9910719
ISIC_0000117	0.958181424	0.915480852	0.958181424	0.945357931	0.998791923
ISIC_0000125	0.989171007	0.837710438	0.989171007	1	0.995114329
ISIC_0000126	0.951692708	0.932048802	0.951692708	0.962760971	0.989453849
ISIC_0000129	0.978537326	0.923731661	0.978537326	0.989883268	0.999927322
ISIC_0000132	0.971809896	0.833811521	0.971809896	1	0.983427666
ISIC_0000136	0.912673611	0.916134655	0.912673611	0.915671117	0.998963731
ISIC_0000138	0.957118056	0.936072229	0.957118056	0.99926557	0.978181627
ISIC_0000141	0.966080729	0.951151965	0.966080729	0.990950837	0.992578354
ISIC_0000149	0.880989583	0.898064409	0.880989583	0.913544293	0.92778278
ISIC_0000160	0.83921441	0.900678516	0.83921441	0.871242596	0.909498208

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0000161	0.931445313	0.952811614	0.931445313	0.976154025	0.98478711
ISIC_0000164	0.937565104	0.913607386	0.937565104	0.97530356	0.964039676
ISIC_0000165	0.79312066	0.54274323	0.79312066	0.992348054	0.794519188
ISIC_0000169	0.892881944	0.880052407	0.892881944	0.996231362	0.899481737
ISIC_0000172	0.9453125	0.961971045	0.9453125	0.995663601	0.988344177
ISIC_0000174	0.847417535	0.78433635	0.847417535	0.996652805	0.836975512
ISIC_0000177	0.967686632	0.926153846	0.967686632	0.989416971	0.988498547
ISIC_0000188	0.874609375	0.933118128	0.874609375	1	0.78948237
ISIC_0000195	0.923307292	0.958049019	0.923307292	0.965392889	0.984852207
ISIC_0000196	0.941427951	0.930037	0.941427951	0.959877889	0.990285846
ISIC_0000197	0.935091146	0.942175692	0.935091146	0.994486784	0.965390879
ISIC_0000198	0.931966146	0.903649427	0.931966146	1	0.946977585
ISIC_0000200	0.926453993	0.849592581	0.926453993	0.828450406	1
ISIC_0000201	0.967708333	0.960940781	0.967708333	1	0.99125851
ISIC_0000202	0.910633681	0.950757731	0.910633681	0.944966295	0.984267952
ISIC_0000212	0.877907986	0.908907084	0.877907986	0.890226398	0.995162144
ISIC_0000213	0.922460938	0.939292049	0.922460938	0.993840952	0.941788994
ISIC_0000222	0.937478299	0.974511131	0.937478299	0.998995238	0.97217042
ISIC_0000226	0.893988715	0.888189727	0.893988715	0.881821225	0.987759885
ISIC_0000227	0.974522569	0.961788307	0.974522569	0.997854691	0.997422884
ISIC_0000228	0.94702691	0.953953954	0.94702691	0.967498194	0.991868381
ISIC_0000230	0.874609375	0.917189304	0.874609375	0.899549059	0.98314898
ISIC_0000231	0.957313368	0.966772205	0.957313368	0.998156206	0.995151002
ISIC_0000233	0.907226563	0.969561926	0.907226563	1	0.835894438
ISIC_0000234	0.939778646	0.94511595	0.939778646	0.963997022	0.990090444
ISIC_0000246	0.85703125	0.914313554	0.85703125	0.99976496	0.813026991
ISIC_0000253	0.97358941	0.949565565	0.97358941	0.998764517	0.991812041
ISIC_0000254	0.966362847	0.94302554	0.966362847	0.980472517	0.995160364
ISIC_0000257	0.942773438	0.905219997	0.942773438	0.97922176	0.965800162
ISIC_0000258	0.919335938	0.935285242	0.919335938	0.948815728	0.976098539
ISIC_0000271	0.970811632	0.956099059	0.970811632	0.991491267	0.999220208
ISIC_0000279	0.926236979	0.961157626	0.926236979	0.983551176	0.98073508
ISIC_0000287	0.935177951	0.938303431	0.935177951	0.995854333	0.958451164
ISIC_0000289	0.707226563	0.779481879	0.707226563	0.676044804	0.997998314
ISIC_0000298	0.929492188	0.822247446	0.929492188	1	0.942209757
ISIC_0000299	0.864084201	0.917190019	0.864084201	0.901644952	0.899725617
ISIC_0000302	0.902278646	0.931696007	0.902278646	0.922932787	0.987995515
ISIC_0000306	0.906293403	0.806270863	0.906293403	1	0.917338556
ISIC_0000310	0.933550347	0.976244018	0.933550347	0.994574713	0.925326096
ISIC_0000311	0.744986979	0.837568171	0.744986979	0.771324212	0.815421228
ISIC_0000319	0.966796875	0.920139126	0.966796875	0.94945741	0.997803749

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0000320	0.93374566	0.937781217	0.93374566	0.974502644	0.964126973
ISIC_0000325	0.95952691	0.964572266	0.95952691	0.995374503	0.989745973
ISIC_0000327	0.961979167	0.892385578	0.961979167	0.907828825	0.999289698
ISIC_0000328	0.880685764	0.869565217	0.880685764	0.868255758	0.951580835
ISIC_0000340	0.973632813	0.942715074	0.973632813	0.986735002	0.998827025
ISIC_0000343	0.815256076	0.797136038	0.815256076	0.969282066	0.791470764
ISIC_0000354	0.944140625	0.960519336	0.944140625	0.982588287	0.983457371
ISIC_0000357	0.974262153	0.926434744	0.974262153	0.978865704	0.99304017
ISIC_0000368	0.928645833	0.95578312	0.928645833	0.965530556	0.98062488
ISIC_0000371	0.863650174	0.848609937	0.863650174	0.792110824	0.998954385
ISIC_0000377	0.953927951	0.92814781	0.953927951	0.974156401	0.983906443
ISIC_0000378	0.910763889	0.917905794	0.910763889	1	0.930986757
ISIC_0000380	0.906358507	0.922710043	0.906358507	0.959800683	0.94384176
ISIC_0000388	0.973220486	0.945009023	0.973220486	0.998843931	0.990734629
ISIC_0000392	0.926302083	0.874742829	0.926302083	0.866754292	0.989743116
ISIC_0000393	0.890733507	0.900435084	0.890733507	0.883728714	0.990947525
ISIC_0000407	0.916145833	0.92644075	0.916145833	0.996405722	0.931045805
ISIC_0000418	0.934700521	0.907960309	0.934700521	0.990069723	0.958719772
ISIC_0000420	0.934027778	0.914393388	0.934027778	0.98356472	0.96043817
ISIC_0000433	0.821809896	0.824213553	0.821809896	0.753307943	1
ISIC_0000448	0.947005208	0.897512226	0.947005208	0.943360305	0.980289887
ISIC_0000449	0.987999132	0.902164502	0.987999132	0.987554905	0.99979632
ISIC_0000456	0.894748264	0.887898621	0.894748264	0.998563476	0.907550867
ISIC_0000466	0.938237847	0.953645751	0.938237847	0.996926005	0.970790716
ISIC_0000470	0.956206597	0.888945298	0.956206597	0.92961634	0.999330674
ISIC_0000476	0.859157986	0.84126042	0.859157986	0.820101742	0.994779827
ISIC_0000479	0.793142361	0.869869945	0.793142361	0.9835917	0.721789178
ISIC_0000482	0.957747396	0.960397737	0.957747396	0.999901787	0.993067946
ISIC_0000484	0.850217014	0.881590786	0.850217014	0.967512526	0.858310076
ISIC_0000487	0.889084201	0.891807787	0.889084201	0.996873168	0.920038276
ISIC_0000490	0.912369792	0.889882954	0.912369792	0.963366566	0.938445918
ISIC_0000494	0.955794271	0.8937583	0.955794271	0.940568475	0.985391393
ISIC_0000499	0.959657118	0.902622852	0.959657118	0.955242291	0.991769337
ISIC_0000501	0.895030382	0.826708607	0.895030382	0.999464955	0.907017968
ISIC_0000508	0.903645833	0.861649268	0.903645833	1	0.917693337
ISIC_0000509	0.945594618	0.957429501	0.945594618	0.999534125	0.97308948
ISIC_0000510	0.938042535	0.937736146	0.938042535	0.999436408	0.964481279
ISIC_0000515	0.895876736	0.887349654	0.895876736	0.874511376	0.986260553
ISIC_0000518	0.948871528	0.922069044	0.948871528	0.996633469	0.974401958
ISIC_0000525	0.946050347	0.942392299	0.946050347	0.967080512	0.997627311
ISIC_0000527	0.937825521	0.924479274	0.937825521	0.98824558	0.966677394

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0000533	0.914800347	0.912333201	0.914800347	0.94745462	0.965439519
ISIC_0000534	0.805750868	0.819503928	0.805750868	0.890329752	0.802243818
ISIC_0000537	0.736284722	0.744990849	0.736284722	0.873742948	0.725626844
ISIC_0000539	0.904036458	0.941683694	0.904036458	0.983638392	0.906280144
ISIC_0000540	0.927690972	0.964649733	0.927690972	0.998072527	0.948416373
ISIC_0000547	0.793793403	0.833906669	0.793793403	0.991645286	0.765866515
ISIC_0000549	0.798046875	0.841107977	0.798046875	0.999525005	0.771875766
ISIC_0000550	0.914257813	0.941288329	0.914257813	0.999627283	0.929295453
ISIC_0001100	0.967990451	0.919055893	0.967990451	0.960243668	0.996541846
ISIC_0001103	0.875173611	0.732581223	0.875173611	0.656817978	0.988169048
ISIC_0001128	0.960894097	0.935861819	0.960894097	0.993760635	0.983221654
ISIC_0001131	0.927734375	0.935990302	0.927734375	0.979820927	0.960500667
ISIC_0001142	0.908398438	0.937140625	0.908398438	0.983313122	0.916999405
ISIC_0001181	0.92046441	0.52905437	0.92046441	0.991549296	0.934160004
ISIC_0001185	0.981032986	0.781584582	0.981032986	0.841031794	1
ISIC_0001186	0.985980903	0.87076537	0.985980903	0.965690377	0.99900731
ISIC_0001190	0.976931424	0.793451569	0.976931424	0.984562607	0.988174078
ISIC_0001204	0.95562066	0.885901228	0.95562066	0.99665392	0.97343296
ISIC_0001242	0.451801215	0.11951567	0.451801215	1	0.443178235
ISIC_0001299	0.970876736	0.891911657	0.970876736	0.995117982	0.989309691
ISIC_0001427	0.975846354	0.902013025	0.975846354	0.983333333	0.99369137
ISIC_0001484	0.980490451	0.903306301	0.980490451	0.989186113	0.996581942
ISIC_0001685	0.978993056	0.921213808	0.978993056	0.987432065	0.997919277
ISIC_0001960	0.967751736	0.861238938	0.967751736	0.996545769	0.98458676
ISIC_0002107	0.97578125	0.950323141	0.97578125	0.999075999	0.996029509
ISIC_0002246	0.868619792	0.559769101	0.868619792	0.999317406	0.874870295
ISIC_0002673	0.978862847	0.822239936	0.978862847	0.876160991	0.999837708
ISIC_0002829	0.978428819	0.941534144	0.978428819	0.996051889	0.996427116
ISIC_0002871	0.957769097	0.573405833	0.957769097	0.591939547	0.984501071
ISIC_0003056	0.962955729	0.428018303	0.962955729	0.415153907	0.991475217
ISIC_0003559	0.839171007	0.398649234	0.839171007	0.948347107	0.8468135
ISIC_0003728	0.406206597	0.170000623	0.406206597	1	0.385541613
ISIC_0004115	0.96000434	0.87419283	0.96000434	0.991972831	0.977064329
ISIC_0004346	0.98187934	0.895933839	0.98187934	0.98989899	0.994580439
ISIC_0005000	0.984635417	0.915395284	0.984635417	0.999198718	0.997017421
ISIC_0006193	0.982421875	0.900388973	0.982421875	0.957630443	0.999276715
ISIC_0006711	0.980230035	0.702254642	0.980230035	0.748211731	1
ISIC_0007322	0.97719184	0.851931686	0.97719184	0.902320359	0.999296567
ISIC_0007693	0.982682292	0.902605803	0.982682292	0.981021238	0.999559471
ISIC_0008116	0.974631076	0.921527486	0.974631076	1	0.989779893
ISIC_0008207	0.979861111	0.912417663	0.979861111	0.999231951	0.993520176

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0008406	0.978602431	0.720758069	0.978602431	0.788448623	1
ISIC_0008507	0.987456597	0.882236842	0.987456597	0.982471516	0.999707168
ISIC_0008600	0.964735243	0.932772489	0.964735243	0.985583685	0.992463724
ISIC_0008626	0.971072049	0.908072214	0.971072049	0.971562809	0.998288006
ISIC_0008659	0.980837674	0.894058154	0.980837674	0.999233716	0.992919513
ISIC_0008992	0.968815104	0.755254554	0.968815104	0.7675	0.991889422
ISIC_0008998	0.927539063	0.934484247	0.927539063	0.974696551	0.98109984
ISIC_0009035	0.937478299	0.854056437	0.937478299	0.997564935	0.954141016
ISIC_0009078	0.976063368	0.789646513	0.976063368	0.84057971	1
ISIC_0009083	0.966970486	0.759113391	0.966970486	0.76122449	0.996405816
ISIC_0009201	0.962478299	0.909389216	0.962478299	0.959938102	0.990520964
ISIC_0009298	0.974631076	0.818471338	0.974631076	0.856132075	1
ISIC_0009564	0.985546875	0.906028369	0.985546875	0.99915683	0.99808187
ISIC_0009869	0.97734375	0.8363974	0.97734375	0.88855806	0.999534526
ISIC_0009872	0.899045139	0.351589315	0.899045139	0.971014493	0.906483986
ISIC_0009874	0.955664063	0.859482689	0.955664063	0.863172462	0.994574631
ISIC_0009879	0.93218316	0.898453469	0.93218316	0.904491649	0.986535538
ISIC_0009880	0.927213542	0.835592047	0.927213542	0.804992349	1
ISIC_0009881	0.965668403	0.944833476	0.965668403	0.986901246	0.997155859
ISIC_0009882	0.962326389	0.942718405	0.962326389	0.979149291	0.999556121
ISIC_0009885	0.98344184	0.904274108	0.98344184	0.973771857	0.999860413
ISIC_0009889	0.973155382	0.936942238	0.973155382	0.98632398	0.995665695
ISIC_0009891	0.968402778	0.956153546	0.968402778	0.995846444	0.995849722
ISIC_0009898	0.884288194	0.72741281	0.884288194	0.647398844	1
ISIC_0009901	0.939474826	0.953988118	0.939474826	0.988009765	0.976650351
ISIC_0009902	0.927864583	0.903404644	0.927864583	0.905059045	0.991328794
ISIC_0009906	0.930577257	0.945611944	0.930577257	0.99985306	0.954200215
ISIC_0009918	0.866297743	0.889516273	0.866297743	0.914806876	0.895604125
ISIC_0009920	0.881966146	0.851510919	0.881966146	0.910608896	0.912764153
ISIC_0009923	0.92172309	0.915714915	0.92172309	0.989379542	0.944343295
ISIC_0009926	0.974479167	0.917654463	0.974479167	0.973600587	0.998511542
ISIC_0009927	0.960177951	0.924097632	0.960177951	0.960367526	0.990821295
ISIC_0009928	0.931749132	0.968433882	0.931749132	0.984678889	0.965048088
ISIC_0009930	0.987456597	0.797391304	0.987456597	0.926352129	0.999955256
ISIC_0009931	0.950716146	0.955249975	0.950716146	0.98756793	0.988414614
ISIC_0009943	0.950390625	0.850393701	0.950390625	0.901319677	0.984906703
ISIC_0009945	0.974674479	0.926060845	0.974674479	0.972108844	0.997324168
ISIC_0009946	0.909852431	0.826005133	0.909852431	0.795577355	0.993643092
ISIC_0009948	0.913151042	0.931130899	0.913151042	0.940703971	0.970872011
ISIC_0009954	0.935503472	0.761777063	0.935503472	0.712432432	0.991014706
ISIC_0009955	0.926019965	0.76568573	0.926019965	0.716919456	0.999919335

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0009956	0.963346354	0.959551619	0.963346354	0.991271607	0.996143803
ISIC_0009958	0.945486111	0.866061226	0.945486111	0.857621233	0.996438669
ISIC_0009959	0.966536458	0.854644561	0.966536458	0.869813665	0.996938999
ISIC_0009965	0.894552951	0.86148619	0.894552951	0.831062204	0.997454423
ISIC_0009970	0.909092882	0.957148173	0.909092882	0.965110457	0.924828394
ISIC_0009977	0.952018229	0.861754181	0.952018229	0.887299316	0.991146548
ISIC_0009980	0.901128472	0.954890159	0.901128472	0.940565508	0.982526316
ISIC_0009982	0.944075521	0.773533671	0.944075521	0.73533018	0.999593093
ISIC_0009988	0.809917535	0.899371771	0.809917535	0.83881517	0.978873239
ISIC_0009990	0.89140625	0.946777083	0.89140625	0.924795628	0.94581861
ISIC_0009992	0.86312934	0.879269729	0.86312934	0.852227484	0.959948639
ISIC_0009993	0.942100694	0.951510644	0.942100694	0.994915949	0.97588294
ISIC_0009994	0.959570313	0.949090263	0.959570313	1	0.986376166
ISIC_0009998	0.936002604	0.865733877	0.936002604	0.855616943	1
ISIC_0010004	0.984722222	0.795911414	0.984722222	0.870597871	0.999729272
ISIC_0010009	0.976866319	0.928	0.976866319	1	0.993853991
ISIC_0010011	0.93812934	0.890166912	0.93812934	0.891170431	0.995508579
ISIC_0010013	0.932139757	0.804689949	0.932139757	0.769369369	1
ISIC_0010016	0.982118056	0.911656245	0.982118056	0.975702354	0.99922751
ISIC_0010018	0.971592882	0.948679472	0.971592882	0.995456928	0.996785714
ISIC_0010020	0.885091146	0.874541489	0.885091146	0.842731981	0.996079601
ISIC_0010023	0.936957465	0.90664319	0.936957465	0.941241076	0.983563632
ISIC_0010028	0.964474826	0.923749503	0.964474826	0.960789119	0.999317443
ISIC_0010033	0.961132813	0.932817915	0.961132813	0.988076312	0.990607519
ISIC_0010034	0.944943576	0.947293993	0.944943576	0.978613381	0.988750657
ISIC_0010037	0.844704861	0.906977715	0.844704861	0.868179114	0.979735556
ISIC_0010038	0.846050347	0.844081084	0.846050347	0.788211999	0.999802771
ISIC_0010041	0.650238715	0.745251878	0.650238715	0.62501682	0.998666074
ISIC_0010047	0.939409722	0.9134188	0.939409722	0.926414949	0.992671552
ISIC_0010055	0.932269965	0.865729828	0.932269965	0.853636364	0.997770776
ISIC_0010058	0.963888889	0.949386213	0.963888889	0.98572768	0.997204154
ISIC_0010059	0.978472222	0.91951292	0.978472222	0.989528796	0.99664201
ISIC_0010061	0.97875434	0.914306641	0.97875434	0.973337075	0.999399875
ISIC_0010062	0.90483941	0.941687049	0.90483941	0.934571575	0.992787561
ISIC_0010073	0.872439236	0.817762216	0.872439236	0.771221972	0.992460045
ISIC_0010077	0.868793403	0.684133916	0.868793403	0.980041037	0.876666751
ISIC_0010088	0.981141493	0.918395155	0.981141493	0.981481481	0.998895547
ISIC_0010089	0.983550347	0.933232856	0.983550347	0.995703545	0.999671014
ISIC_0010092	0.968250868	0.950178444	0.968250868	0.998643838	0.991906152
ISIC_0010100	0.900368924	0.897454942	0.900368924	0.892321672	0.975622747
ISIC_0010103	0.782052951	0.633154433	0.782052951	0.902143418	0.780867383

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0010171	0.945008681	0.924646572	0.945008681	0.947045708	0.996871672
ISIC_0010173	0.95405816	0.789008212	0.95405816	0.7597742	0.998712266
ISIC_0010175	0.871896701	0.910111828	0.871896701	0.887618516	0.997059841
ISIC_0010180	0.958203125	0.900733496	0.958203125	0.936042848	0.988590795
ISIC_0010183	0.946267361	0.684123026	0.946267361	0.954640907	0.960493598
ISIC_0010187	0.946180556	0.94976029	0.946180556	0.980189544	0.992833477
ISIC_0010190	0.935503472	0.974401836	0.935503472	0.975344638	0.98827504
ISIC_0010192	0.950846354	0.879495817	0.950846354	0.899307108	0.999972043
ISIC_0010193	0.709960938	0.661424956	0.709960938	0.603804319	0.840820854
ISIC_0010201	0.935134549	0.952961892	0.935134549	0.972788815	0.985616214
ISIC_0010202	0.908007813	0.947763578	0.908007813	0.927412578	0.980896948
ISIC_0010206	0.978255208	0.94123556	0.978255208	1	0.994843278
ISIC_0010207	0.956510417	0.79533911	0.956510417	0.999014778	0.972230572
ISIC_0010215	0.951605903	0.889471752	0.951605903	0.905200433	0.99870572
ISIC_0010216	0.962217882	0.890386825	0.962217882	0.935714286	0.991296136
ISIC_0010229	0.96328125	0.951573455	0.96328125	0.996491668	0.993239375
ISIC_0010231	0.902061632	0.918779444	0.902061632	0.915177159	0.996484396
ISIC_0010234	0.964084201	0.719520548	0.964084201	0.708843537	1
ISIC_0010238	0.910698785	0.864846959	0.910698785	0.837342423	1
ISIC_0010254	0.982638889	0.879690698	0.982638889	0.941143181	0.997233527
ISIC_0010255	0.988302951	0.902621723	0.988302951	0.991056911	0.999481339
ISIC_0010257	0.59921875	0.652139629	0.59921875	0.526143687	0.894860571
ISIC_0010261	0.930490451	0.877023066	0.930490451	0.999821747	0.955638751
ISIC_0010326	0.856228299	0.771494416	0.856228299	0.68938857	0.99798702
ISIC_0010331	0.908094618	0.953678154	0.908094618	0.96178461	0.935209248
ISIC_0010336	0.931727431	0.897420241	0.931727431	0.908944659	0.986620092
ISIC_0010340	0.953233507	0.966732168	0.953233507	0.998832321	0.989057828
ISIC_0010346	0.928059896	0.883779939	0.928059896	0.872973815	0.994768193
ISIC_0010347	0.911393229	0.800238048	0.911393229	0.754093732	0.999852894
ISIC_0010348	0.809049479	0.518153442	0.809049479	0.393397638	1
ISIC_0010360	0.928515625	0.880945259	0.928515625	0.871025526	0.999325561
ISIC_0010368	0.96547309	0.963319371	0.96547309	1	0.990982045
ISIC_0010369	0.810329861	0.865466151	0.810329861	0.795363894	1
ISIC_0010373	0.888042535	0.725920417	0.888042535	0.670395447	0.986999826
ISIC_0010374	0.937239583	0.962227696	0.937239583	0.986366048	0.97591461
ISIC_0010377	0.933051215	0.957556703	0.933051215	0.998709927	0.963136706
ISIC_0010378	0.974414063	0.845578127	0.974414063	0.931232092	0.994031254
ISIC_0010379	0.892382813	0.828226478	0.892382813	0.943771358	0.912790532
ISIC_0010437	0.970681424	0.719343494	0.970681424	0.730932203	0.999492749
ISIC_0010444	0.980186632	0.855199773	0.980186632	1	0.991694614
ISIC_0010448	0.861610243	0.932652822	0.861610243	0.981521455	0.728557468

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0010449	0.948263889	0.965495022	0.948263889	1	0.974299899
ISIC_0010451	0.930403646	0.772740234	0.930403646	0.746200608	0.99577476
ISIC_0010452	0.895768229	0.89460743	0.895768229	1	0.904879536
ISIC_0010454	0.861545139	0.924665775	0.861545139	0.97949419	0.800483537
ISIC_0010456	0.882790799	0.415565869	0.882790799	1	0.89113678
ISIC_0010460	0.912825521	0.941559384	0.912825521	0.989147928	0.918155826
ISIC_0010463	0.958940972	0.954322461	0.958940972	0.984887186	0.998550295
ISIC_0010474	0.978710938	0.923816679	0.978710938	0.986717268	0.995595849
ISIC_0010477	0.839626736	0.905238709	0.839626736	0.868116304	0.947704082
ISIC_0010483	0.956141493	0.919497441	0.956141493	0.99456959	0.979539706
ISIC_0010494	0.955425347	0.957221652	0.955425347	0.983113018	0.992115244
ISIC_0010498	0.900802951	0.788403369	0.900802951	0.729195442	1
ISIC_0010552	0.897743056	0.922091951	0.897743056	0.99968252	0.903908565
ISIC_0010553	0.971527778	0.947360704	0.971527778	0.993733062	0.997154722
ISIC_0010556	0.939605035	0.889817561	0.939605035	0.912036596	0.9974861
ISIC_0010565	0.732204861	0.678539053	0.732204861	1	0.6815148
ISIC_0010574	0.931358507	0.829009193	0.931358507	0.934651763	0.960067621
ISIC_0010582	0.917925347	0.837971359	0.917925347	0.999455733	0.936702737
ISIC_0010584	0.924023438	0.765380078	0.924023438	1	0.935546168
ISIC_0010587	0.971918403	0.944998615	0.971918403	0.987964989	0.997769189
ISIC_0010588	0.800130208	0.893988198	0.800130208	0.832103275	0.910899654
ISIC_0010591	0.94296875	0.930017944	0.94296875	0.992625556	0.973481538
ISIC_0010592	0.984049479	0.830291262	0.984049479	0.998704663	0.992916333
ISIC_0010596	0.922265625	0.916167665	0.922265625	0.915446504	0.999875941
ISIC_0010597	0.954079861	0.951834565	0.954079861	0.980259487	0.996535066
ISIC_0010598	0.982638889	0.933354901	0.982638889	0.996408619	0.998366509
ISIC_0010599	0.941449653	0.955980042	0.941449653	0.991923204	0.982766377
ISIC_0010604	0.755837674	0.853997039	0.755837674	0.774090865	0.997806216
ISIC_0010606	0.956163194	0.924705237	0.956163194	0.987817259	0.982636726
ISIC_0010607	0.969856771	0.876024412	0.969856771	1	0.984358491
ISIC_0010846	0.798611111	0.815920836	0.798611111	0.769328043	0.907312684
ISIC_0010847	0.65718316	0.700546357	0.65718316	0.833468989	0.576957278
ISIC_0010854	0.894422743	0.709122433	0.894422743	0.93439441	0.911894383
ISIC_0010856	0.937239583	0.905157379	0.937239583	0.927869444	0.987601321
ISIC_0010863	0.888259549	0.934962483	0.888259549	1	0.891647588
ISIC_0011080	0.772048611	0.687917022	0.772048611	0.68890832	0.85608083
ISIC_0011081	0.823155382	0.81134151	0.823155382	0.915221973	0.824371673
ISIC_0011083	0.98812934	0.909323116	0.98812934	0.999139415	0.999031859
ISIC_0011089	0.928884549	0.859959815	0.928884549	0.860532407	0.986001979
ISIC_0011090	0.926736111	0.924949651	0.926736111	0.947655398	0.967608952
ISIC_0011092	0.965277778	0.878402229	0.965277778	0.92921466	0.993376333

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0011098	0.823415799	0.543799772	0.823415799	0.954901495	0.828853485
ISIC_0011101	0.964040799	0.959952291	0.964040799	0.997835052	0.990760808
ISIC_0011104	0.876302083	0.858682183	0.876302083	0.815763132	0.989389183
ISIC_0011107	0.815668403	0.832117372	0.815668403	0.768831169	0.999111427
ISIC_0011110	0.973632813	0.943343653	0.973632813	0.990992582	0.998499224
ISIC_0011112	0.906684028	0.959115932	0.906684028	0.968703428	0.899966167
ISIC_0011122	0.863606771	0.899366314	0.863606771	0.872458145	0.976701719
ISIC_0011129	0.931510417	0.966523921	0.931510417	0.998188687	0.966905298
ISIC_0011132	0.930859375	0.8369111	0.930859375	0.906879958	0.967308469
ISIC_0011143	0.990907118	0.868438991	0.990907118	1	0.997794928
ISIC_0011148	0.927907986	0.90235623	0.927907986	1	0.949424976
ISIC_0011149	0.903168403	0.929252234	0.903168403	0.934250588	0.961595927
ISIC_0011150	0.916341146	0.742381057	0.916341146	0.923750243	0.937790901
ISIC_0011151	0.894118924	0.951810194	0.894118924	0.93624311	0.950449224
ISIC_0011152	0.975260417	0	0.975260417	0	0.979981683
ISIC_0011155	0.886848958	0.855338884	0.886848958	0.816728167	1
ISIC_0011162	0.957530382	0.924485461	0.957530382	0.988073394	0.985348731
ISIC_0011167	0.829947917	0.899659078	0.829947917	0.864355926	0.909349409
ISIC_0011168	0.967100694	0.879008746	0.967100694	1	0.981752673
ISIC_0011171	0.978320313	0.946471739	0.978320313	0.999441185	0.99673391
ISIC_0011172	0.87484809	0.866970517	0.87484809	0.999605795	0.89260355
ISIC_0011175	0.927170139	0.922744238	0.927170139	0.948685137	0.973819248
ISIC_0011176	0.952799479	0.947649693	0.952799479	0.984243435	0.99248655
ISIC_0011177	0.935221354	0.953366476	0.935221354	0.991167574	0.972440658
ISIC_0011204	0.944466146	0.899687197	0.944466146	0.982135972	0.966880457
ISIC_0011205	0.978884549	0.931500873	0.978884549	0.984696438	0.999272057
ISIC_0011213	0.983116319	0.892610425	0.983116319	0.955965182	0.998482793
ISIC_0011219	0.972222222	0.943242457	0.972222222	1	0.99177467
ISIC_0011224	0.905967882	0.812704014	0.905967882	0.862557898	0.95932951
ISIC_0011227	0.977821181	0.876637341	0.977821181	0.970125786	0.992237915
ISIC_0011292	0.966102431	0.894946568	0.966102431	0.945034354	0.997728974
ISIC_0011294	0.901692708	0.850284824	0.901692708	0.999176164	0.921020101
ISIC_0011298	0.981315104	0.936285831	0.981315104	0.995112414	0.998910237
ISIC_0011300	0.707421875	0.822162274	0.707421875	0.738686576	0.804676754
ISIC_0011305	0.740516493	0.537735849	0.740516493	0.960943396	0.736465574
ISIC_0011310	0.785894097	0.688971258	0.785894097	0.609568358	0.945638294
ISIC_0011319	0.953385417	0.94357137	0.953385417	0.990213116	0.984506498
ISIC_0011321	0.972157118	0.946365081	0.972157118	0.991967871	0.996408284
ISIC_0011325	0.909809028	0.93273468	0.909809028	1	0.918981878
ISIC_0011333	0.946202257	0.957080973	0.946202257	0.994712991	0.979528117
ISIC_0011336	0.938693576	0.946739894	0.938693576	0.987572254	0.972808575

Image	Accuracy	Dice	Jaccard	Sensitivity	Specificity
ISIC_0011338	0.310481771	0.347677596	0.310481771	1	0.194432729
ISIC_0011344	0.810655382	0.681232212	0.810655382	1	0.799862847
ISIC_0011349	0.871875	0.926283409	0.871875	0.959423729	0.829699511
ISIC_0011359	0.938780382	0.924082401	0.938780382	1	0.962511765
ISIC_0011363	0.906684028	0.941584212	0.906684028	0.95463855	0.943267647
ISIC_0011367	0.982269965	0.88421872	0.982269965	0.943396226	0.998313542
ISIC_0011374	0.683485243	0.813197002	0.683485243	0.699376013	nan
ISIC_0011384	0.799023438	0.694946341	0.799023438	1	0.789814914
ISIC_0011386	0.965342882	0.797968254	0.965342882	0.808039165	1
ISIC_0011392	0.906684028	0.941584212	0.906684028	0.95463855	0.943267647
Average	0.917644603	0.868903788	0.917644603	0.930153063	0.954467315



Appendix D

Segmentation examples

Appendix D includes some examples of perfectly and automatically segmented skin images.



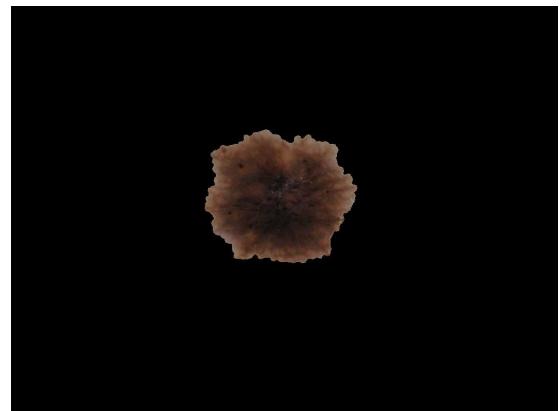
ISIC_0000000.jpg



Perfectly_Segmented_ISIC_0000000.jpg



ISIC_0000001.jpg



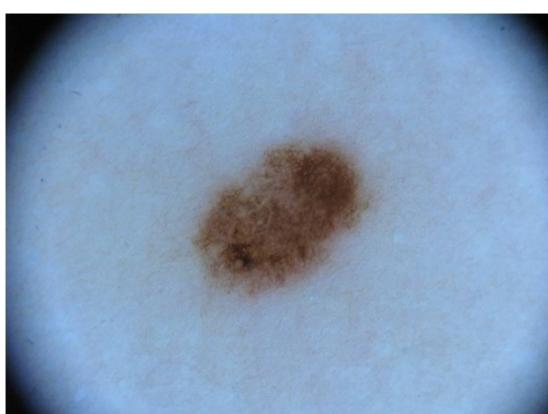
Perfectly_Segmented_ISIC_0000001.jpg



ISIC_0000006.jpg



Perfectly_Segmented_ISIC_0000006.jpg



ISIC_0000007.jpg



Perfectly_Segmented_ISIC_0000007.jpg



ISIC_0000003.jpg



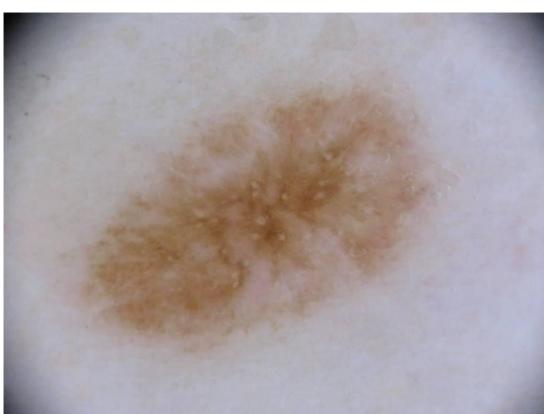
Automatically_Segmented_ISIC_0000003.jpg



ISIC_0000012.jpg



Automatically_Segmented_ISIC_0000012.jpg



ISIC_0000014.jpg



Automatically_Segmented_ISIC_0000014.jpg



ISIC_0000015.jpg



Automatically_Segmented_ISIC_0000015.jpg

Bibliography

- [1] ISIC Archive. International skin imaging collaboration: Melanoma project website. [online]. <https://isic-archive.com/>, 2016.
- [2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 86(11), pp.2278-2324., 1998.
- [3] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1520-1528), 2015.
- [4] E. Mohedano. Deep learning for computer vision barcelona - image classification. [online]. <http://www.slideshare.net/xavigiro/image-classification-dlcv-d112>, 2016.
- [5] W. Sterry and R. Paus. Thieme clinical companions dermatology. In *Thieme Verlag. Stuttgart, New York.*, 2006.
- [6] Skin Cancer Foundation. Skin cancer facts and statistics. [online]. <http://www.skincancer.org/skin-cancer-information/skin-cancer-facts>, 2016. [Accessed: January 12 2017].
- [7] American Cancer Society. Skin cancer prevention and early detection. [online]. <http://www.cancer.org/acs/groups/cid/documents/webcontent/003184-pdf.pdf>, 2015.
- [8] A. G. Goodson and D. Grossman. Strategies for early melanoma detection: approaches to the patient with nevi. [online]. <http://www.cancer.org/acs/groups/cid/documents/webcontent/003184-pdf.pdf>, 2009.
- [9] R.L. Siegel, K.D. Miller, and A.Jemal. Cancer statistics. In *A cancer journal for clinicians*, 66(1), pp.7-30., 2016.
- [10] F. Nachbar, W. Stolz, T. Merkle, A.B. Cognetta, T. Vogt, M. Landthalerv, P. Bilek, O. Braun-Falco, and G. Plewig. The abcd rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions. In *Journal of the American Academy of Dermatology*, 30(4), pp.551-559., 1994.
- [11] H.P. Soyer. Three-point checklist of dermoscopy. In *Dermatology*, 208(1), pp.27-31., 2004.
- [12] G. Argenziano. Seven-point checklist of dermoscopy revisited. In *British Journal of Dermatology* 164, no. 4 (2011): 785-790., 2011.
- [13] S.W. Menzies. A method for the diagnosis of primary cutaneous melanoma using surface microscopy. In *Dermatologic clinics*, 19(2), pp.299-305., 2001.
- [14] J.S. Henning, S.W. Dusza, S.Q. Wang, A.A. Marghoob, H.S. Rabinovitz, D. Polsky, and A.W. Kopf. The cash (color, architecture, symmetry, and homogeneity) algorithm for dermoscopy. In *Journal of the American Academy of Dermatology*, 56(1), pp.45-52., 2007.
- [15] J. Gachon, P. Beaulieu, J.F. Sei, J. Gouvernet, J.P. Claudel, M. Lemaitre, M.A. Richard, and J.J. Grob. First prospective study of the recognition process of melanoma in dermatological practice. In *Archives of Dermatology*, 141(4), 2005, 434-8, 2005.
- [16] G. Argenziano and H.P. Soyer. Dermoscopy of pigmented skin lesions—a valuable tool for early diagnosis of melanoma. In *The Lancet Oncology*, 2(7), 2001, 443-9., 2001.

- [17] A. Oakley. Dermoscopy. [online]. <http://www.dermnetnz.org/topics/dermoscopy/>, 2004.
- [18] H. Kittler, H. Pehamberger, K. Wolff, and M. Binder. Diagnostic accuracy of dermoscopy. In *The Lancet Oncology*, 3(3), 2002, 159-65, 2002.
- [19] J. Kawahara, A. BenTaieb, and G. Hamarneh. Deep features to classify skin lesions. In *IEEE International Symposium on Biomedical Imaging (IEEE ISBI)*, 2016.
- [20] H. Liao. A deep learning approach to universal skin disease classification. [online]. https://www.cs.rochester.edu/u/hliao6/projects/other/skin_project_report.pdf, 2016.
- [21] N. Codella, Q.B. Nguyen, S. Pankanti, D. Gutman, B. Helba, A. Halpern, and J.R. Smith. Deep learning ensembles for melanoma recognition in dermoscopy images. In *arXiv preprint arXiv:1610.04662*, 2016.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105), 2012.
- [23] Computer & Electrical Engineering and Computer Science Department. Florida atlantic university. [online]. <http://www.ceecs.fau.edu/>, 2016.
- [24] ISBI. Ieee international symposium on biomedical imaging. [online]. <http://biomedicalimaging.org/>, 2016.
- [25] International Symposium on Biomedical Imaging. Isbi 2016: Skin lesion analysis towards melanoma detection. [online]. <https://challenge.kitware.com/#challenge/560d7856cad3a57cfde481ba>, 2016.
- [26] International Symposium on Biomedical Imaging. Isbi 2016 - lesion segmentation. [online]. <https://challenge.kitware.com/#phase/566744dccad3a56fac786787>, 2016.
- [27] International Symposium on Biomedical Imaging. Isbi 2016 - lesion dermoscopic feature extraction. [online]. <https://challenge.kitware.com/#phase/56674518cad3a56fac78678c>, 2016.
- [28] International Symposium on Biomedical Imaging. Isbi 2016 - lesion classification. [online]. <https://challenge.kitware.com/#phase/5667455bcad3a56fac786791>, 2016.
- [29] International Symposium on Biomedical Imaging. Isbi 2016 - segmented lesion classification. [online]. <https://challenge.kitware.com/#phase/56fc2763cad3a54f8bb80e51>, 2016.
- [30] International Skin Imaging Collaboration. Isbi 2017: Skin lesion analysis towards melanoma detection. [online]. https://challenge.kitware.com/#challenge/n/ISIC_2017%3A_Skin_Lesion_Analysis_Towards_Melanoma_Detection, 2016.
- [31] Python Software Foundation. Python programming language. [online]. <https://www.python.org/>, 2016.
- [32] MathWorks. Matlab. [online]. <https://www.mathworks.com/products/matlab.html>, 2016.
- [33] F. Chollet. Keras documentation. [online]. <https://keras.io/>, 2016.

- [34] F. Chollet. Introducing keras 1.0. [online]. <https://blog.keras.io/introducing-keras-1.0.html>, March 2015.
- [35] TensorFlow Development Team. Tensorflow library for machine intelligence. [online]. <https://www.tensorflow.org/>, 2016.
- [36] Theano Development Team. Theano python library. [online]. <http://deeplearning.net/software/theano/#>, 2016.
- [37] SciPy. Scientific computing tools for python documentation. [online]. <https://www.scipy.org/>, 2016.
- [38] PIL. Python imaging library documentation. [online]. <http://www.pythonware.com/products/pil/>, 2016.
- [39] NVidia. Cuda parallel computing platform. [online]. http://www.nvidia.com/object/cuda_home_new.html, 2016. [Online]; [Accessed: January 12 2017].
- [40] NVidia. Nvidia geforce gtx titan x. [online]. <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x>, 2016. [Online]; [Accessed: January 12 2017].
- [41] X. Li, B. Aldridge, L. Ballerini, R. Fisher, and J. Rees. Depth data improves skin lesion segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Sep 2009.
- [42] S. Dreiseitl, L. Ohno-Machado, H. Kittler, S. Vinterbo, H. Billhardt, and M. Binder. A comparison of machine learning methods for the diagnosis of pigmented skin lesions. In *Journal of Biomedical Informatics*, 34(1), 2001, 28-36., 2001.
- [43] C. Barata, M. Ruela, M. Francisco, T. Mendonça, and J.S. Marques. Two systems for the detection of melanomas in dermoscopy images using texture and color features. In *IEEE Systems Journal*, 8(3), 2014.
- [44] M. Walter. Is this the end? machine learning and 2 other threats to radiology's future. [online]. <http://www.radiologybusiness.com/topics/technology-management/end-machine-learning-and-2-other-threats-radiology%20%99s-future>, 2016.
- [45] S. Jha. Will computers replace radiologists?. [online]. <http://www.medscape.com/viewarticle/863127>, 2016.
- [46] X. Giró i Nieto, E. Sayrol, A. Salvador, J. Torres, E. Mohedano, and K. McGuinness. Deep learning for computer vision barcelona. [online]. <http://imatge-upc.github.io/telecombcn-2016-dlcv/>, 2016.
- [47] F. Li, A. Karpathy, and J. Johnson. Cs231n: Convolutional neural networks for visual recognition. [online]. <http://cs231n.stanford.edu/>, 2016.
- [48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A.C. Berg. Imagenet large scale visual recognition challenge. In *International Journal of Computer Vision*, 115(3), 2015.
- [49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *arXiv preprint arXiv:1512.03385*, 2015.

- [50] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431-3440), 2015.
- [51] S.J. Pan and Q. Yang. A survey on transfer learning. In *IEEE Transactions on knowledge and data engineering*, 22(10), pp.1345-1359., 2010.
- [52] CS231n Course. Transfer learning. [online]. <http://cs231n.github.io/transfer-learning/>, 2016.
- [53] 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* (pp. 1-9), 2015.
- [54] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241), 2015.
- [55] IEEE International Symposium on Biomedical Imaging. Cell tracking challenge. [online]. <http://biomedicalimaging.org/2015/hardi-reconstruction-challenge/>, 2015.
- [56] J. Marko. Deep learning tutorial for kaggle ultrasound nerve segmentation competition, using keras. [online]. <https://github.com/jocicmarko/ultrasound-nerve-segmentation>, August 2016.
- [57] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556.*, 2014.
- [58] Keras framework. Usage of optimizers. [online]. <https://keras.io/optimizers/>, 2016.
- [59] Y. Shen. Loss functions for binary classification and class probability estimation. In *A dissertation in statistics*, 2005.
- [60] NVidia. Nvidia geforce gtx titan x. [online]. <http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x>, 2016.
- [61] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980.*, 2014.
- [62] U. Sanchez. Isbi 2016 challenge results. <https://challenge.kitware.com/#submission/56fe2b60cad3a55ecee8cf74>, 2016.
- [63] L. Yu. Isbi 2016 challenge results. <https://challenge.kitware.com/#submission/56feb054cad3a55ecee8ecde>, 2016.
- [64] M. Rahman. Isbi 2016 challenge results. <https://challenge.kitware.com/#submission/56fbfa1bcad3a54f8bb809bf>, 2016.
- [65] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *In Advances in neural information processing systems* (pp. 3320-3328)., 2014.
- [66] Amazon. Amazon web services (aws). [online]. <https://aws.amazon.com/>, 2016. [Accessed: January 12 2017].
- [67] Amazon. Aws calculator. [online]. <https://calculator.s3.amazonaws.com/index.html#r=IAD&s=EC2&key=calc-7BD7FB26-9B47-4B91-A320-DD2C9379D108>, 2016. [Accessed: January 12 2017].

- [68] cs231n. Regularization of convolutional neural networks for visual recognition. [online]. <http://cs231n.github.io/neural-networks-2/>, 2016.
- [69] Dermnet. Dermnet skin disease atlas. [online]. <http://www.dermnet.com/>, 2016. [Accessed: January 12 2017].
- [70] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *arXiv preprint arXiv:1512.03385.*, 2015.
- [71] IASTED. International association of science and technology for development. [online]. <https://www.iasted.org/>, 2016.
- [72] IASTED. Biomedical engineering 2017 conference. [online]. <https://www.iasted.org/conferences/home-852.html>, 2016.
- [73] IEEE Engineering in Medicine and Biology Society. 13th iasted international conference on biomedical engineering. [online]. <http://www.embs.org/events/13th-iasted-international-conference-biomedical-engineering/>, 2016.
- [74] Adria Romero Lopez, Xavier Giro i Nieto, Jack Burdick, and Oge Marques. Skin lesion classification from dermoscopic images using deep learning techniques presented at the iasted 13th international conference on biomedical engineering (biomed 2017), innsbruck, austria, 20–21 february 2016. 2016.
- [75] SIIM. Society for imaging informatics in medicine. [online]. <http://siim.org/>, 2016.
- [76] SIIM. Siim 2017 annual meeting. [online]. <http://siim.org/page/SIIM2017>, 2016.