Skin Lesion Classification using Deep Learning Neural Networks

by CHRYSANTHI KATRINI

BSc, Technological Educational Institute of Crete, 2014

THESIS

Submitted in partial fulfillment of the requirements for degree of Master in Science

HERAKLION,CRETE

Approved by

Professor Nikolaos Papadakis

Statement of Originality

The work contained in this thesis has not been previously submitted for a degree or diploma at any other higher education institution or any other purpose. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except as specified in references, acknowledgments or in footnotes. I certify that the intellectual content of this thesis is the product of my own work and all the assistance received in preparing this thesis and sources have been acknowledged.

Abstract

Melanoma is a type of skin cancer and it is characterized from the experts as the most aggressive. An early diagnosis and a surgery removal can give to the patient almost 99% survival rate. Several Computer-Aided Diagnosis (CAD) systems have been proposed to assist dermatologists in an early diagnosis. This thesis, it is dealing with the processing of color images that depict images of patients with possible melanoma. The main point is to build a system to identify cases that could be potentially dangerous. The system performs feature extraction using the SIFT and SURF algorithm and these features fed into several classifiers such as Support Vector Machines (SVM), K-Nearest Neighbor (K-NN) and Convolutional Neural Network (CNN) and achieve $94,51\%$ accuracy.

Keywords: Skin cancer, SIFT, SURF, Medical image segmentation, Deep Learning, Convolutional neural networks

Acknowledgments

I would like to express my gratitude to my thesis advisor Dr.Nikolaos Papadakis of the Department of Informatics Engineering at Technological Educational Institute of Crete. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it. Also, I would like to thank especially Mojdeh Rastgoo for her guidance and help and all the colleagues from Le2i Lab in Le Creusot for their support. Finally, I must express my very profound gratitude to my family as well as to all the friends of mine for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

List of Figures

Chapter 1

Introduction

.

1.1 Human skin and moles

Human skin is the largest organ of a human body. Its composition consists of three layers: the epidermis, the dermis and the subcutaneous tissue (usually called as fat).

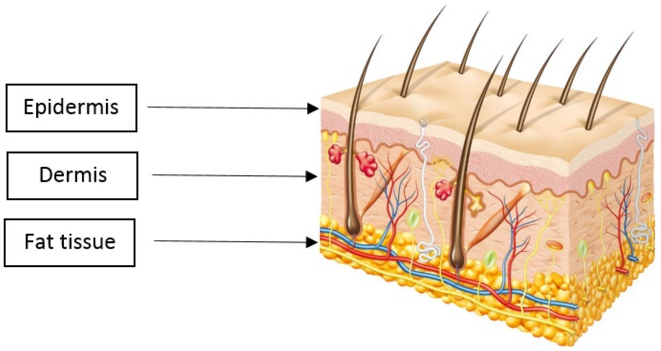


Figure 1.1: Layers of human skin

The epidermis has two part of cells, the melanocytes and the keratinocytes, which are producing melanin, a pigment that protects against the harmful effects of sunlight. Sometimes the melanocytes may create darker accumulations, which are known as nevi (or nevus). Nevi appears on the surface of the skin and under some occasions it can be transformed in melanoma. Some of these causes are[1]:

- **Age**

  Statistics shows that as the age is increasing as the danger for someone to develop melanoma is bigger.

- **Previous illness**

  If someone in the past had diseased from melanoma, there are high probabilities to get sick again as this is happening with all the types of cancer.

- **Many moles**

  Sometimes, on the surface of the human skin there are too many moles of a different kind.

- **Strong family history of melanoma**

  As happened in all the types of cancer, if someone has illness history to his family from 2 or more first-degree relatives affected, the probabilities to get affected are high.

- **White skin that burns easily**

  When the skin color is lighter then ultraviolet radiation is more harm compare with someone with darker skin tone.

.

The nevis appearance varies in color, texture and size. Dermatologists have concluded that there are five key features for diagnose malignant or benign melanoma[1]:

$A-$ ASYMMETRY

If a line separates the mole into half and the one half is not the same as the other half. Melanoma is likely to be asymmetrical or irregular.

$B-$ BORDER

The borders of an early melanoma is usually uneven with jagged edges.

$C-$ COLOR

A mole has more than one color, usually brown mixed with black, red, white and pink.

$D-$ DIAMETER

Moles are smallest than 6mm. In case of bigger mole there is the suspicion of melanoma.

$E-$ EVOLVING

There are changes in moles size, color, shape or bleeding.

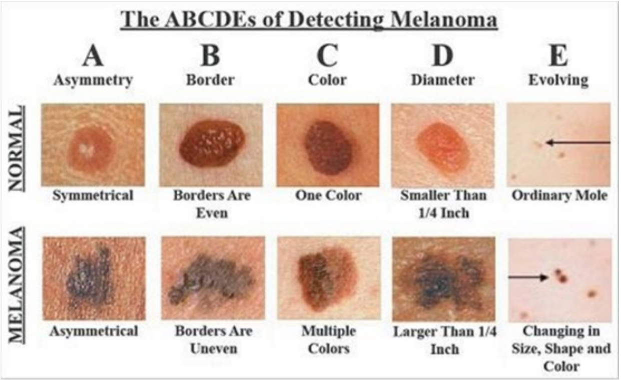


Figure 1.2: ABCDE rule for detecting melanoma

1.2 Research Motivation

Melanoma is a very dangerous dermatological disease and unfortunately more and more people suffering from this. As in every disease, an early diagnosis could save a life. A dermatologist must be sure and quick for the diagnosis. But it is not every time so easy to be right in brief time. There are many countries around the world where hospitals or even doctors are rare[2].
How a doctor will do a diagnosis if he can not be near the patient? Nowadays there are several ways for people to communicate from distance, and all this technology can be used for medical reason. This work proposes an automatic classification of moles to assist dermatologists in their diagnosis.

1.3 Thesis Outline

The thesis is organized as follows:
Chapter 1: The current chapter has briefly introduced the topic and highlighted the scope and the objectives of the present thesis.
Chapter 2: Analyze the meanings of medical imaging and image processing and examined all the methods which are suitable for skin cancer.
Chapter 3: Explanation of machine learning and how deep learning integrated in the field of computer science.
Chapter 4: Extend explanation of Convolution neural network.
Chapter 5: Analyzes all the technical implementation, the used algorithms and methods will be detailed about their parameters.
Chapter 6: Presents all the result and discusses over the results will be made.
Chapter 7: The last chapter is the conclusion of the present study which also identifies directions for future work.

Chapter 2

Medical Image Processing

Image processing is a method to apply operations on an image, in order to get some useful information from it. During this process images are converted into digital. Image processing is a type of signal processing in which as input is considered an image and as output an image or some characteristics associated with the input image. Segmentation of an image is part of image processing and is the method which is used for segmented a digital image is multiple sections. The main scope is to make the images description simpler to analyze. This works by putting a label in each pixel of the image so pixels with the same label usually have the same characteristics. Segmentation has a significant role to digital image processing and pre-processing for object recognition and image retrieval.

2.1 Methods of Segmentation

Several methods had been developed for segmentation and each of these is related to different applications and different images. There is not a general rule under what occasions these methods work. For that reason, Haralick and Shapiro determined that a good segmentation method must[3] :-

- Separate the regions based on one characteristic so results should be as homogeneous as possible.

- The content of each region should be as simple as it can be.

2.1.1 Thresholding

The most common and most easy method of image segmentation is called thresholding method. The main point of this method is to select the threshold value so all the extracted information from the image will be useful for next steps [4]. A popular thresholding method is Otsu method which convert a gray-scale image into binary image. The two levels of the binary image are assigned to pixels those are above or below the threshold value. So, according to this, threshold value will be between 0 and 1. During this method frequency and mean value should be calculated[5]. This method is fully unsupervised segmentation and requires no changes in images parameter.

2.1.2 Clustering methods

Clustering is able to group sets of objects in such way that all the objects those belongs to the same group have more similarities between them instead to those in other groups. There are plenty of algorithms for clustering but choosing one of those it is depended on the individual dataset as clustering can be applied in many field such as machine learning, image analysis, computer graphics and pattern recognition. A popular and most common used method for clustering is K-means. K-means is a classifier which finds a specific number of clusters which are represented by their centroids. This clustering technique is based on minimizing the sum of the squares of all elements distances in each class form center of that class [6] .

$$min \sum_{x \in S_j(K)} |x - z_j|^2$$

The steps of this algorithm are the following:

1. Determine the number of classes, named K.

2. Select randomly elements as K classes centroids.

3. For the remaining elements, distance from all the centers is calculated and the elements are placed in the appropriate class based on the minimum distance.

4. Classes centroids recalculated.

5. Minimum distances from the centers recalculated and the data are repositioned.

6. Step 4 and step 5 are repeated until the centers of the classes have no change.

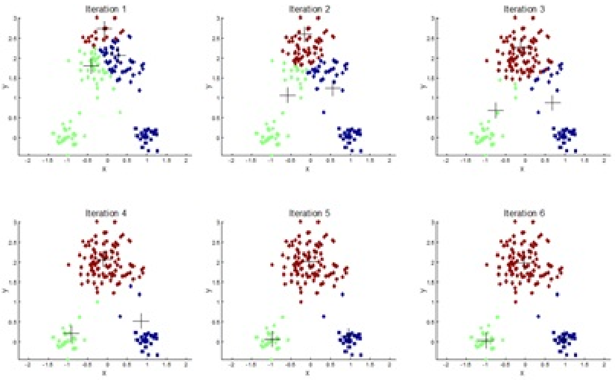


Figure 2.1: K-means clustering

2.1.3 Edge detection

Edge detection is a technique for finding objects boundaries within images. It is a technique which is used for image segmentation purposes and it based on the detection of a distinct break in continuity in brightness. By applying an edge detector to an image, it will reduce the amount of data as its result will be the boundaries of objects. The points of an image where brightness is changing, called edges. Edges can be horizontal, vertical or diagonal. Most of the shape information which is useful for next steps is enclosed in edges. The three main steps for edge detection are [7] :-

1. Filtering: it is a frequent occurrence for images to be corrupted by noise. As the filtering is been increasing, the noise result in a loss of edge strength is been reducing.

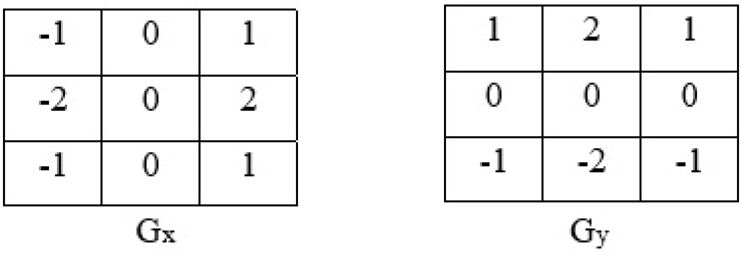
2. Enhancement: it is emphasizes pixels where change in local intensity values has been observe by calculate the gradient magnitude, so the detection of edges will be easiest.

3. Detection: many points in an image may have a nonzero value for the gradient but still they will not be edges. A threshold can be used to determine which of these points belong to an edge.

For finding the edges in an image, there are several masks for someone to apply such as:

- **Sobel operator**

This operator consists of a pair of 3x3 convolutional kernels.



-1	0	1
-2	0	2
-1	0	1

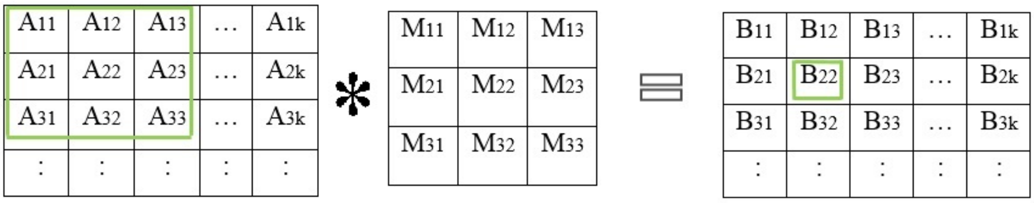
G_x

1	2	1
0	0	0
-1	-2	-1

G_y

Figure 2.2: Masks used for Sobel operator

These masks are usually smaller than the actual image so they can easily slide over of the image, change the pixels value and continue to the right until they rich to the end of the row. They continue with the next row until they have pass through all the pixels [8]. An illustration of how Sobel operator works, it is shown below.



$$\begin{array}{|c|c|c|c|c|}\hline A_{11} & A_{12} & A_{13} & \ldots & A_{1k} \\ \hline A_{21} & A_{22} & A_{23} & \ldots & A_{2k} \\ \hline A_{31} & A_{32} & A_{33} & \ldots & A_{3k} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline M_{11} & M_{12} & M_{13} \\ \hline M_{21} & M_{22} & M_{23} \\ \hline M_{31} & M_{32} & M_{33} \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|}\hline B_{11} & B_{12} & B_{13} & \ldots & B_{1k} \\ \hline B_{21} & B_{22} & B_{23} & \ldots & B_{2k} \\ \hline B_{31} & B_{32} & B_{33} & \ldots & B_{3k} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline \end{array}$$

The value of a specific pixel after the application of the mask is

$B_{22} = (A_{11} * M_{11}) + (A_{12} * M_{12}) + (A_{13} * M_{13}) + (A_{21} * M_{21}) + (A_{22} * M_{22}) + (A_{23} * M_{23}) + (A_{31} * M_{31}) + (A_{32} * M_{32}) + (A_{33} * M_{33})$

The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Typically, an approximate magnitude, for speed purpose, is computed using:

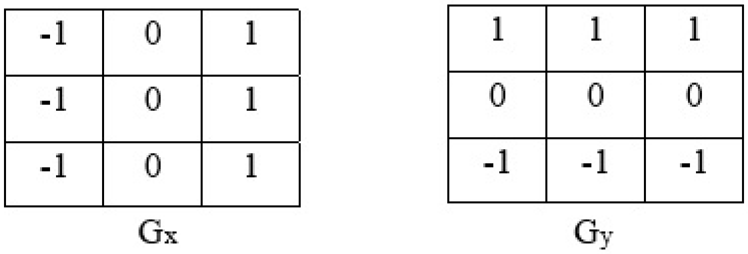
$$|G| = |G_x| + |G_y|$$

The angle of orientation of the edge giving rise to the spatial gradient is given by:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) - \left(\frac{3\pi}{4}\right)$$

- **Prewitt operator**

This edge detector is a discrete differentiation operator and can estimate the magnitude and the orientation of an edge. It is used to detect vertical and horizontal edges in images. It uses two 3x3 kernels with the same logic as Sobel [9].



-1	0	1
-1	0	1
-1	0	1

Gx

1	1	1
0	0	0
-1	-1	-1

Gy

Figure 2.3: Masks used for Prewitt operator

- **Canny edge detector**

This algorithm is created to be an edge detector according to three main criteria. First criterion is to decrease the error rate as it is important to detect all the edges in an image. Second criterion is the points of the detected edge to be as presided as possible to the actual edge. Third criterion is to be only one response per edge. The algorithms steps are[10] :-

1. Remove the noise and smooth the image-usually applying Gaussian filter.

2. Find the intensity gradients of the image.

3. Apply non-maximum suppression for disburden any false response to edge detection.

4. Determine edges by applying threshold.

5. Remove the edges that are not connected to strong edges.

2.2 Features extraction

A very important part in object recognition projects is the features extraction from the image. During this procedure, the segmented image is being processed using a user/programmer-selected method of features extraction, with only purpose to gathering a set of characteristics that efficiently describe the most important information. To analyze many variables requires computation power and memory. Talking about object recognition and classification, the amount of the data is huge. Features extraction come again to solve this problem as involves reducing of this amount of data. Many approaches have been proposed, analyzed and tested for features extraction. These approaches can be divided into four main categories according the edge, shape, texture and color[11]. In this thesis only two of these approaches will be tested, the Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF).

2.2.1 Feature Extraction using SIFT

This approach is used for detection and extraction of local descriptors of images. This algorithm got published in 1999 from David Lowe and has patented in USA from University of British Columbia[12]. SIFT is used for object recognition, robotics, 3D image reconstruction, gesture recognition and detect moving object in video. For every object in the image can be extracted interesting points of it, so these points can provide feature description. This feature description can be inserted in some system for training it and in a future use of this system it can detect the object with the same characteristics in the test image. For a reliable recognition, it is important the characteristics of an object can also be remaining in a scale-change of the image, change of noise and lighting. Usually, these points are placed to the edges of the object where contracts are bigger.

2.2.2 Feature Extraction using SURF

This method is a fast algorithm for local descriptor-based approach. Points of interest of an image are defined as features from a scale-invariant representation. SURF was presented in 2006 to the 9th European Conference on Computer Vision on Graz, Austria by Herbert Bay. It is partly inspired by SIFT descriptor, but the basic version of SURF is faster and is considered by the creators to be more resilient to the various image transformations than the SIFT method. SURF is based on Haar 2D wavelet and makes effective use of embedded images[13].

Chapter 3

Machine Learning

3.1 Machine Learning

The main goal of Machine Learning (ML) is the development of systems that are able to autonomously change their behavior based on experience[14].
ML methods use training data to induce general models that are able to detect the presence or absence of patterns in new (test) data. In the case of images, training data may take the form of a set of pixels, regions or images, which can be labeled or not. Patterns may correspond to low-level attributes, e.g. a label for a group of pixels in a segmentation task, or to high-level concepts, e.g. the presence or the absence of a pathology in a diagnostic image. In this case, the addressed problem is image classification. The set of training images contains images associated with labels indicating the class of the image. A feature extraction process is applied to each image in order to generate a vector representation, i.e., an image is represented as a point in a n-dimensional vector space. A learning algorithm is applied to the set of training vectors, generating a classifier model. When new images arrive to the system, the feature extraction process is applied to the image to

generate a feature vector, then, this vector is given as input to the classifier model, which produces as output the predicted class of the image. There are different types of learning problems addressed by ML techniques. The differences are determined by the available information for training and by the nature and goal of the model that will be induced. The following is a list of the most representative learning problems categories [15]:

- Supervised learning. The problem consists in inducing a function that maps inputs (features) to an output. If the output belongs to a discrete domain, it is a classification task. If the domain is continuous, it is a regression task.

- Unsupervised learning. In this case, training data has not associated labels. The ML algorithm must find patterns which are connected to the intrinsic structure of the data.

- Semi-supervised learning. The general problem to solve is a supervised task. However, not all the training samples have labels associated to them.

  In principle, unlabeled samples may be ignored, and a conventional supervised learning algorithm may be applied to the labeled data. However, unlabeled data may give useful information about the structure/distribution of input samples. The challenge is to use unlabeled data to improve the performance of the classifier/regression model.

- Active learning. In some contexts, unlabeled samples may be abundant, but generating a label for a sample may be a costly task, e.g., when this task is performed by human experts. In this case, it may be a good idea to carefully choose which samples to label. An active learning algorithm actively queries the user for labels. In general, this strategy may reduce the number of required samples to reach a given accuracy of the induced model.

- On-line learning. In general, ML techniques use the training data many times along the training process and this usually requires it to be in main memory. However, in some contexts this may be unfeasible, e.g. with huge amount of training data or with training data continuously generated by a real-time process. In this case, the ML algorithm can only keep the training sample for a limited amount of time and then discard it.

Machine learning integrates techniques from different areas including: statistics, signal processing and pattern recognition, among others. The problem of extracting patterns from data is a complex problem and it can be approached from different perspectives. Here we briefly review some of the main techniques:

- Bio inspired methods. Biological systems provide a good source of inspiration to build complex adaptive systems. Artificial neural networks (ANN), the most representative set of techniques in this category, draw their inspiration from biological neural systems. ANN are composed of artificial neurons, which are interconnected mimicking the complex interconnection of neurons in the brain. The first ANN models were developed as early as 1943[16] and there was a good deal of work in the early 60s [17].

- In the 80s there was a renewed interest in ANN thanks to the development of powerful training algorithms for complex multi-layer networks [18]. Nowadays ANN are some of the most widely used ML techniques. Other examples of bioinspired methods are genetic algorithms [19], artificial immune systems [20] and ant-colony optimization [21].

- Rule-induction methods. In general, these kind of methods represent the induced model as a set of rules. Decision-tree-induction algorithms are the most popular among algorithms in this category. The set of rules are represented implicitly by the tree, the leaves represent decisions (class assignments) and the path from the root to a given leaf represent a conjunction of feature values that lead to the respective decision. Examples of these algorithms are ID3 [22], C4.5 and CART. Other rule induction strategies include inductive logic programming [23], rough sets [24] and association rules [25].

- Instance-based methods. These methods do not attempt to build a model, instead they use the full training data set as model to classify new samples. These methods are also called lazy since the processing of training data is delayed until a new sample must be classified. The most popular method in this category is k-nearest neighbor (KNN) classification [26], which classifies new instances using the class information of the closest k training instances. KNN may use different voting schemes to decide the class to assign. Other examples of instance-based algorithms are locally weighted regression[27] and case-based reasoning [28].

- Bayesian methods. These methods model the learning problem using a probabilistic Bayesian framework. In general, Bayesian methods calculate the posterior probability $P[C_i|(x_1, ..., x_n)]$, the probability of a sample $(x1, ..., xn)$ of belonging to the class Ci, for each class C_i and assigns the sample to the class with higher probability. Nave Bayes (NB) is one of the most widely used Bayesian methods for classification [29]. It makes a strong, nave, assumption regarding the independence of the attributes, which simplify the calculation of the posterior probability. Regardless of this strong assumption, NB has shown a competitive performance in different classification problems. An alternative to BN are Bayesian belief networks [30], which can express a richer set of conditional relationships among attributes using a graph representation. Other examples of Bayesian methods are expectation maximization [31], hidden Markov models [32] and Markov random fields [33].

- Kernel methods. Kernel methods represent a relatively new approach to perform machine learning [34]. One of the main distinctive characteristics of these methods is that they do not emphasize the representation of objects as feature vectors. Instead, objects are characterized implicitly by kernel functions.

  A kernel function, $k : AxA \rightarrow R$, takes as arguments two objects, which belong to a set A, and returns a real number, which could be intuitively interpreted as a similarity measure. An interesting property of kernel functions is that they implicitly induce a mapping, $\Phi : A \to F$, from the original space A to a high-dimensional feature space F. In fact, the kernel function corresponds to a dot product in the space F. Kernel methods deal with complex patterns using the mapping induced by the kernel, well known as the kernel trick. For instance, a non-linear pattern in the input space A may become linear in the feature space F using an appropriate kernel.

Kernel methods do not require that training objects be represented by feature vectors, since they only use the information provided by the kernel function. This allows the easy application of kernel methods to complex objects such as trees, graphs, and images. Support vector machine (SVM) classification [35] is the most representative method in this category, but there exist kernel methods for all the main types of learning tasks including regression [36], dimensionality reduction, clustering [37] and semi-supervised learning [38], among others.

3.2 Deep Learning

Deep learning, also known as Deep Structured Learning, Hierarchical Learning or Deep Machine Learning, is a chapter of Machine Learning supported by mass of algorithms, which are intended to model high-level abstractions in data by using multiple processing layers, often with complex structures or conversely, composed of multiple non-linear transformations. Deep Learning belongs to Machine Learning but classical machine learning techniques can solve challenging image classification problems but their results are not good when they applied directly to images.

Deep learning is a new technique to apply machine learning as it is based on learning representations of data. In that it makes use of methods pattern learning methods of data, such an example of input as image, can be mirrored in many different ways, such as a vector of force values per pixel, or in a more abstract way as a set of acmes, districts of a particular shape. Many representations are much better than others at simplifying the learning task. One of the Deep Learning guarantees, is that the features of different algorithms, will be replaced with unsupervised or semi supervised feature learning and hierarchical feature extraction [39]. The exploration of this topic, strides to make a better and better representations and construct models from large-scale unlabeled data. Some of the representations are inspired by advances in neuroscience and are loosely based on the interpretation of information processing and communication patterns in a nervous system, such as neural coding, which attempts to define the relationship between various stimuli and associated neuronal responses in the brain [40]. A lot of Deep Learning tools such as Deep Neural Networks, Convolutional Neural Networks, Deep Belief Networks and Recurrent Neural Networks have been used in fields like Computer Vision, speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks. Deep learning has been characterized as a buzzword, or a rebranding of neural networks [41].

Chapter 4

Artificial Neural Networks & Classifiers

Neural Networks (NN) is a virtual simulation of the human brain consisting of neurons. It consists of a large number of processing units in proportion to the human neural system, and is the smallest independent unit of the network [43]. Neurons are the building blocks of a network and there are three types of neurons:

1. Input neurons: input can be a matrix of data. These neurons do not produce any result on the network, but they offer the input data and communicate with the hidden layers of neurons.

2. Hidden neurons: the most important role is played here which multiply each entry with the concise weight and they give as result the sum of the products which is taken as an argument by the activation function which activates internally each node.

3. Output neurons: offer the neural networks output. The output can be the desired one according to the user or not.

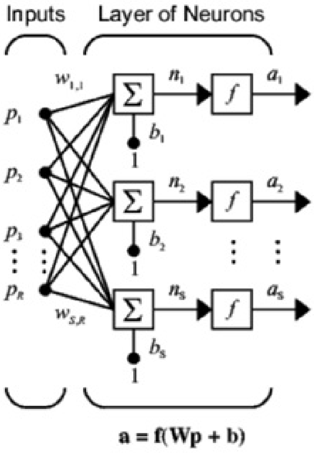


Figure 4.1: Layer of Neurons-source Matlab Toolbox

Neurons of a neural network are the nodes those are interconnected with links called weights and are organized in layers. The number of layers used in each NN, the number of neurons and the way of connection called network architecture. One neuron can have a certain amount of inputs but only one output.

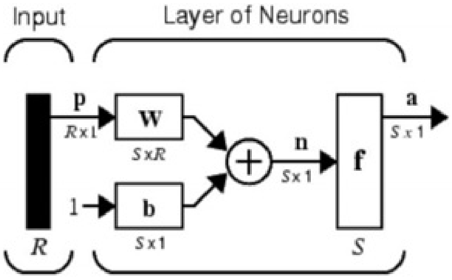


Figure 4.2: Mathematical representation of vector induction at neural layer-source Matlab Toolbox

The input data got matched with the coefficient w (weight) and after inserted in the neuron. These values get summed before their entrance in the neuron and a coefficient b (bias) also added. The output n_i is a number and constitute the input for the neuron which implement the activation function. The output a_i is also a number. All the above computed in each neuron of the layer. As result of each layer is the vector a with S characteristic values, where S is the neurons number.

Some observations until this point are:

1. It is not necessary the number of neurons of each layer to be equal with the characteristic values of the input vector.

2. It is not necessary all the neurons to implement the same function f.

3. There is fully connection between the input vector and the weights. So, it is efficient to use the coefficient w_{ij}, where i the number of neuron and j the number of connections starting point. For example, when the connection starts from the 2nd characteristic value and ends to the 3rd neuron, the weight is symbolized as w_{32}.

General, all the connections combined a matrix which called weights matrix:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

4.1 Neural Network training and learning

The term training means the process of weight change in the most effective way for learning purpose. The network relates the training prototypes with the desired outputs so that problems as pattern recognition can be solved. There are lots of training algorithms but only two learning categories: supervised and unsupervised learning [42].

In supervised learning, the system must learn a concept from a set of input data where each set can be a description of a model. In fact, there is a supervisor who provides the correct output values for the set of data being examined. In these systems, learning must be induced with the target because this is the way in which the predictions are made based on the values of a data set, called variables.

In unsupervised learning, the system must discover associations or groups of data according to its attributes. It creates prototypes without knowing if they exist and what they might be.

An interesting point of a neural network is that it can increase the performance during the training. Improving performance happens over time in relation to some specific processes. During these repetitive processes, continuous adjustments are applied to weights and thresholds. So, the network after every iteration of learning process learns to know better its environment.

The training of a NN is the process by which its parameters are adapted through a continuous process of stimulation from the environment to which the network is adapted. The form of training is determined by how the network parameters change. The above definition implies the following three steps:

1. The network is stimulated from the environment.

2. Because of this stimulation, the network will change.

3. The network responds in a different way to the environment due to changes in its internal structure.

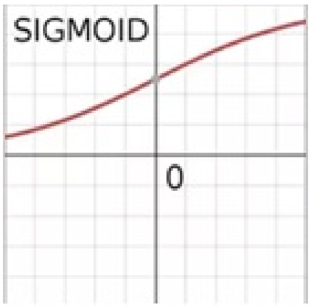
The mathematical expression of changes in the weights of the network is generally expressed by the relation:

$$W_{kj}(n+1) = W_{kj}(n) + \triangle W_{kj}(n)$$

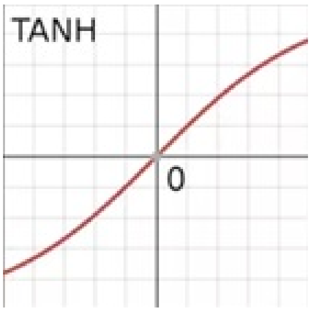
Where $w_{kj}(n)$ is the weight between k and j neurons in the time n, $w_{kj}(n+1)$ is the weight between k and j neurons in the time $n+1$ and $w_{kj}(n)$ is the weight variation in the time n.
Weight variation are governed by specific and well-defined rules. The set of these rules defines the training algorithm. Training algorithms differ in how the weights are adjusted, but also how the network is related to its environment.
Multilayer networks consist from layers of neurons those are connected neurons by neurons. Recognition sign of each connection is the weight w_{ij} which take the weighted output of the j neuron and insert it in the i neuron. Each neuron has, also, the singular value b (bias). After this procedure,$2D$ matrixes are made between layers.
In each layer, there is a bias matrix for each neuron [43].

This kind of networks can solve nonlinear problems, so there are different kind of transfer function which are implemented:

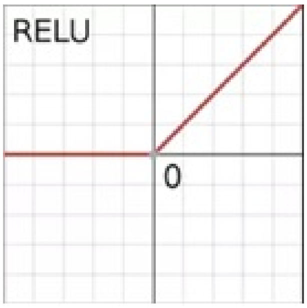
- Sigmoid transfer function



- Tan-sigmoid transfer function with output between -1 and 1



- Rectified Linear Unit (ReLU) function This function is thresholded at zero: $f(x) = max(0, x)$



4.1.1 Convolutional Neural Networks (CNNs / ConvNets)

Convolutional Neural Networks (CNN) are main tools for deep learning and they are ideal for image recognition because they can learn direct from images.

This kind of neural networks has similarities with the neural networks which have been described above: they contain neurons, weights and biases. Each neuron receives an input and gives an output.

CNN have one or more fully connected layers as happened in a multilayer neural network but they are easier to train with fewer parameters instead of fully connected networks with the same amount of hidden neurons.

An important advantage of CNN is that a network can have hundreds of layers that each learn to detect different features of an image, this is the reason why CNNs can be used also for extracting features from an input image and use these features to train a classifier. The output of each layer is used as input in the next layer[44].

CNNs consistence is described by four main types of layers [45]. Those are:

- Convolution Layer: a set of learnable filters which is slided over the image, computing dot products between the entries of the filter and the input image. These filters extend to the full depth of the input image, and they will activate when they see same structure in the images.
- Pooling Layer: the goal here is to reduce the spatial size of the representation and to reduce the parameters and computation in the network. There are several functions for this purpose, but max pooling is the most common one. For example, a pooling layer of size 2x2 will reduce the input image to one quarter of its original size.

- Non-linear Layer: In the architecture of a CNN someone can find a variety of functions such as rectified linear units (RELUs) which implements the function $y = max(0, x)$, so the input and output sizes of this layer are the same.
- Fully-connected Layer: neurons in this type of layer have fully connections to all activations in the previous layer

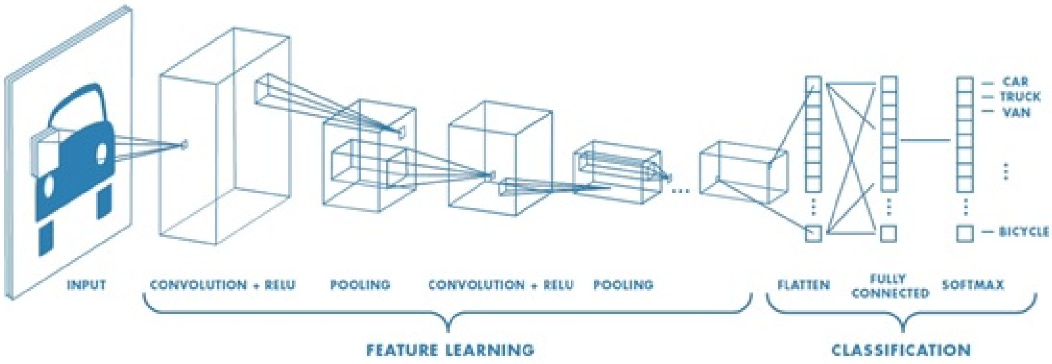


Figure 4.3: Example of multilayer CNN

4.2 Classifiers

4.2.1 Support Vector Machine

A support vector machine (SVM) is a classifier and is considered as a very important notion in the field of statistics and computer science for a set of related supervised learning methods, which analyze data and recognize prototypes and used as classification and regression analysis. The basic algorithm SVM algorithm was invented by Vladimir Vapnik and Alexey Chervonenkis in 1963. The inceptive to develop this method lies in the fact that data classification is a regular task in the field of machine learning. The principle of SVM is that there are some data points and two or more classes. The goal is to decide in which of these classes each point belongs. This procedure is making the machine a non-probabilistic binary linear classifier. SVM creates a model that separate new samples between the categories. This SVM model is a representation of samples, mapped and separated by a clear gap as wide as possible [46].

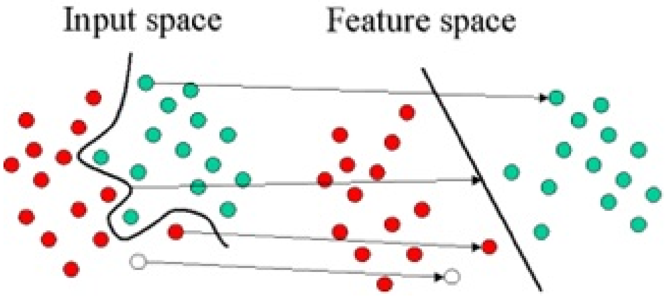


Figure 4.4: Support Vector Machine

4.2.2 K-Nearest Neighbor

K-NN algorithm is a non-parametric method for classification and is a type of instance-based learning, or lazy learning. K-NN is characterized as the simplest of all machine learning algorithms and that because it does not use the training data points to do any generalization. All the training data are used during the testing phase, K-NN makes decision based on the entire training data set and stores all available cases and classifiers new cases based on a similarity measure such as distance functions [47]. Most used distance functions in K-NN are:

- Euclidean Distance $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
- Manhattan Distance $\sum_{i=1}^{k}|x_i - y_i|$
- Minkowski Distance $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}}$
- Hamming Distance $D_H = \sum_{i=1}^{k}|x_i - y_i|$

In K-NN classification, an object is classified by the majority vote of its neighbors with the object being assigned to the class most common among its k nearest neighbors

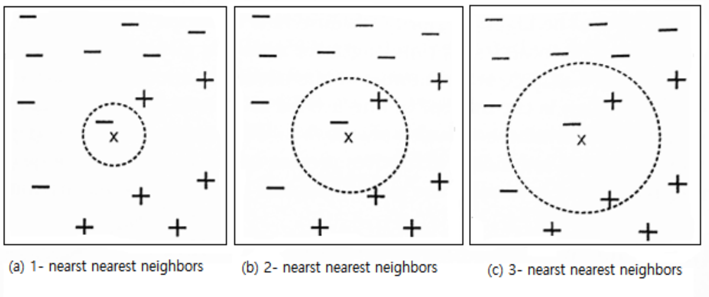

(a) 1- nearst nearest neighbors (b) 2- nearst nearest neighbors (c) 3- nearst nearest neighbors

Figure 4.5: Example of K-NN

Chapter 5

Implementation

5.1 Related Work

- Situ et al[48] the authors proposed a method based on graph cut algorithm. For classification, they used wavelet and Gabor texture features and they extracted features using the color, Scale-Invariant Feature Transform (SIFT) and wavelet features. Their database consists 100 lesions with 30 melanomas and they achieve 83% sensitivity and 80.93% specificity.

- Ganster et al [49] proposed to segment lesions using a fusion method of several thresholding stages. For classifier, they used K-NN and a multiclass classification. Their database has 5380 lesions, including 108 melanomas and as result they got 87% sensitivity and 92% specificity.

- Mahbod et al [50] the authors used the available images from the International Skin Imaging Collaboration (ISIC) 2017 challenge as their dataset which include 2000 dermoscopic skin images with corresponding labels.

For compatibility purposes, they resized the images to 227 × 227 and 224 × 224 so they can be fed to the networks. AlexNet and VGG-16 used as features extractors networks and multi-class non-linear Support Vector Machines for classifying the features.

They split the dataset as 80% for training and 20% for testing and they achieved accuracy for melanoma using AlexNet 83.3% and accuracy for melanoma using VGG-16 82.7%.

- Harangi et al [51] used the available images from the International Symposium on Biomedical Imaging (ISBI) 2017 challenge which include 2000 images as training set, 150 images as validation set and 600 images as test set. They used an ensemble-based system of Deep Convolutional Neural Networks including GoogleNet, AlexNet and VGG-VD-16 architectures and achieved accuracy 86.7%.

- In 2016, Shoieb et al. [45] detect the skin lesion using the color and texture features of the image and K-Means is used to identify similar groups (clustering). They extracted features using CNN, and use these extracted features as input to a SVM to classify skin infected lesion.

  They use three different datasets, the first one is the Dermatology Information System (DermIS) which contains 337 images with their ground truth to be binary (menlanoma or non-melanome) and achieve 93.75% accuracy, the second one is the DermQuest with 134 images again with their ground truth to be binary and achieve 94.12% accuracy and the third dataset is obtained from DermNet Skin Disease Atlas website which contains 134 images with four different diagnosis and for melanoma achieve 98.04% accuracy.

5.2 Materials

For this research, medical images got used. The Vienna dermoscopy dataset was obtained by the Department of Dermatology at the Vienna General Hospital in Austria. This set contains 5380 images, which captured with a hand-held CCD camera with a microscope with resolution of 632×387 pixels. This dataset includes, also, the diagnosis for each sample which 4270 benign, 1002 named as dysplastic nevus and 108 as superficial spreading melanoma (melanoma)[52].

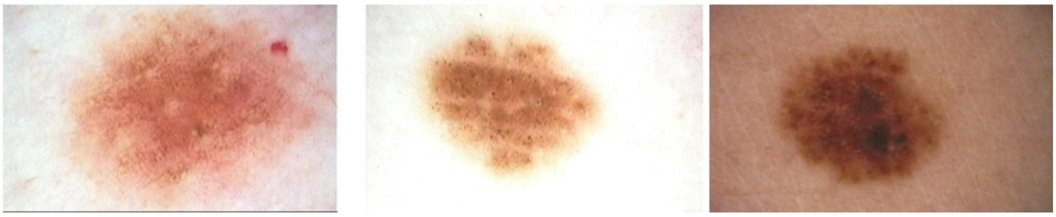


Figure 5.1: Left-Benign Center-Dysplastic Nevus Right-Melanoma

From this dataset and for the purpose of this experiment, 415 images selected, 108 are the melanoma images and the rest were randomly selected from the other two categories as the classification will be binary (melanoma or non-melanoma).

5.3 Methodology

The important steps to diagnosis of melanoma are [?] :-

- Detect correctly the area with the lesion.
- Segment this area from the outer area.
- Extract features from the lesion.
- Classify the images based on the extracted features.

5.3.1 Image Segmentation

Image segmentation refers to partitioning of an image into region. The main goal of segmentation is to simplify and change the representation of an image into easier way of analysis. In this system, Otsu automated thresholding will be used as this method is better for thresholding lesions from the background skin [53].

Pixels either falls in foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum[54]. Otsu method measures the spreading of the pixel levels each side of the threshold and considers that the given image is separated in two different classes of pixels, foreground and background, and the threshold is responsible to separate these classes.

In the segmented image, the bigger area is assumed that is the main region of interest [5]. In this point, it is important to apply an edge detector algorithm for finding the borders of the lesion.

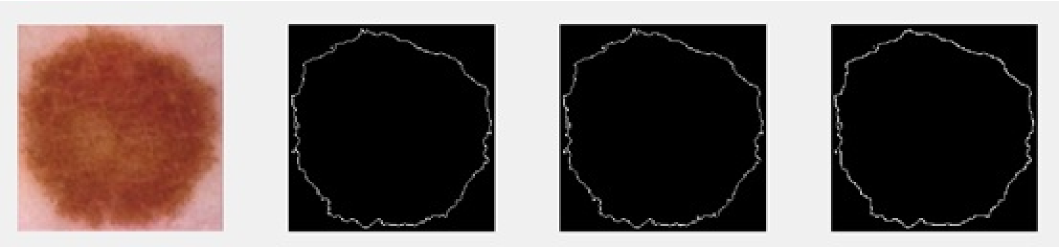


Figure 5.2: Edge detector algorithm

Original Image Sobel Operator Prewitt Operator Canny Operator

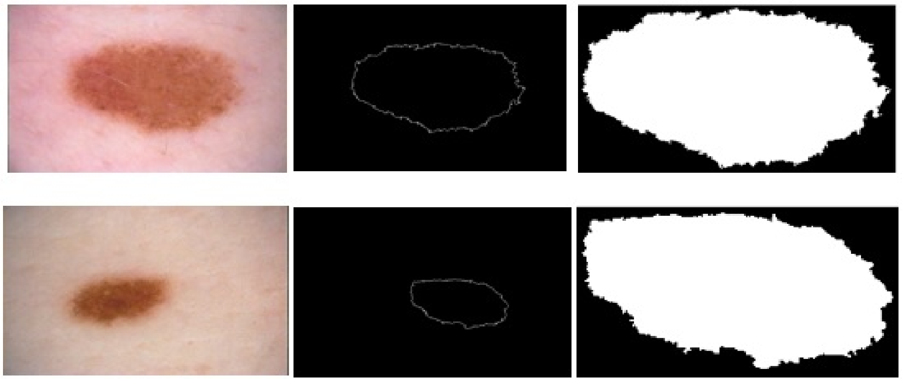
After experiments where edge detectors applied in the whole dataset, it has been observed that Prewitt operator give better results.

An important problem that is been faced during the features extraction was that there were some, and in some cases, many, points those were from the skin and not from the lesion. This can cause in the future during the stage of the neural network.

One solution in this problem is to somehow find and crop the area which contains the lesion. The function regionprops is very useful on this as it can measure the properties of objects in a binary image. In this case, properties will be a bounding box in where the lesion of each image will be enclosed. Using the crop function of Matlab, the area with the lesion is cropped and background pixels are decreased.

After the segmentation, the edge detection and the crop procedure the result is the following:

Next procedure as the image is cropped is to resize it from 632 × 387 pixels to 227 × 227 pixels, so the computational time will reduce and, also, this is the required image size in next steps.



Original image Segmented Image Cropped Image

Figure 5.3: Edge detection and the crop

5.3.2 Features Extraction

As it is mentioned above, database contains colorful images with individual diagnosis each of it. The main scope is to extract features from each one of these images, so those features characterized each case.

First, SIFT method is used, which can detect and describe these local points, called descriptors.

Descriptors is a set of 16 histograms, aligned on a 4×4 grid, each of it with 8 different orientations (the 4 basic directions of a compass and 4 intermediate points of these directions)[55].

So, the size of the vector descriptor is 128-dimensional feature vector, Another important during features extraction is the diagnosis matrix. This matrix has to have the same size as the vector descriptor, so possible matches between these two are realizable.A parameter that directly affects SIFT method is a threshold that limits the amount of exported entities.

For this research, it is better to have a relative uniformity in the samples. This is the reason why after experiments, threshold is finalized in a 15,6 where the amount of descriptors is from $90 - 130$ per image.

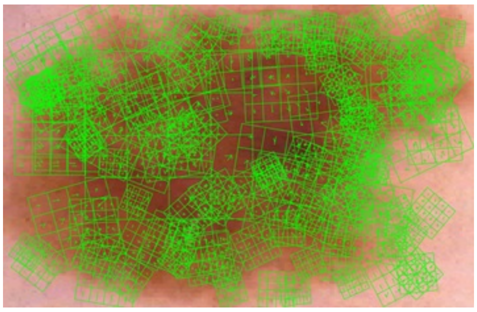


Figure 5.4: SIFT descriptors in skin lesion image

A second attempt for features extraction is SURF. As SIFT and SURF are similar, the implementation for both it was quite similar. So, also in this case, a matrix with diagnosis is used.

The extracted points via this method called SURF Points and their goal is to provide a unique and robust description of an image.

They describe the intensity distribution of the pixels within the neighborhood of the point of interest. Their dimensionality has direct impact on both its computational complexity and point-matching accuracy [56].

For uniformity of this system, only 50 of the strongest points selected for each image.

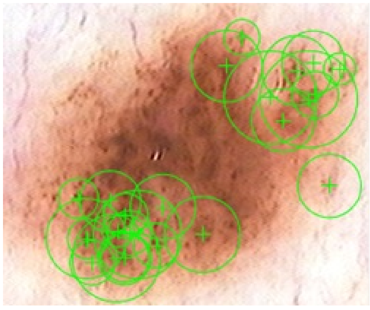


Figure 5.5: SURF descriptors in skin lesion image

5.3.3 Neural Networks

Image processing is a particularly useful and complicated process. Neural networks have much to offer in image processing automation. Thus, they can be used to sort an image by using the original image, some transformation, i.e. Fourier Transform, or certain features of image obtained by known algorithms.

Identifying a goal is a possible application where the neural network separates different target images. A different approach is to find a target that can be in different areas of the image. One application is to fill in the noise-damaged parts of an image.

For better system performance, database was divided as 80% for training the neural networks and 20% for testing the performance of the neural network. In the same logic, vectors exported using SIFT or SURF were divided also in two sets.

Network training parameters have to define for training purpose such as how many epochs will be used, as epochs define how many iterations the network needs to be trained. An important parameter is also the learning rate (lr) whose value shows how much the education will be, as it is a key parameter

for higher quality of the results. The parameters of the network can change the output. For that several values have been tested.
Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton created a large, deep convolutional neural network named AlexNet, which is trained on more than a million images and can classify images into 1000 object categories and also works as feature extractor. It is working by inserting the available dataset and its ground truth and return the results of the classification. This network comprises of 25 layers total, 5 of those are convolutional layers and 3 are fully connected layers. ReLU is been used for activation function, also, softmax loss is minimized, which means to maximize the corresponding column of the true class in the output vector[57] .
To serve the purpose of this thesis and to be trained with the available dataset, AlexNet was remodeled by replacing the last fully connected layer with a new one which include the corresponding datasets classes.

Another ability of AlexNet pretrained neural network is that you can use it for features extraction directly from the dataset and classify them using a SVM.
Features are extracted through activations, which will pull the features learned from the CNN up to that point in the architecture. From the 25 layers of the network only few of them are suitable for features extraction. The layers at the beginning of the network detect edges and blobs.
Typically, the starting layer for features extraction is the layer right before the classification layer.
The implementation was made in a laptop with a CUDA-capable NVIDIA$^{\mathrm{TM}}$GPU with compute capability 3.0 or higher which is highly recommended for implement deep learning.

Chapter 6

Results & Discussion

For all the experiments, a subset of Vienna dataset was used as described above and the image processing part where the images are thresholded and cropped is the same, so the comparison of results will be using the same data. Also, for all the experiments, the 80% of the data is used for training purpose and the rest 20% for testing purpose.

- Experiment 1

  In this experiment, features extracted using SIFT were classified using SVM. Totally, for all the images a vector with 128*40363 descriptors extracted. The 80% used for training (128 × 28254) and the rest 20% for testing (128 × 12109). This method achieved 86% sensitivity and 92% specificity.

- Experiment 2

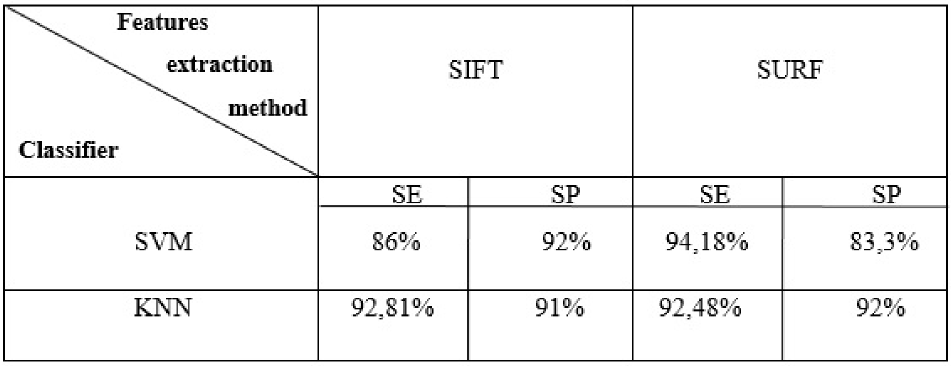
  In this experiment, features extracted using SIFT were classified using KNN with k=2. Also, here the same amount of descriptors was used, divided in the same way as Experiment 1 and achieved $92,81\%$ sensitivity and 91% specificity.

- Experiment 3

  In this experiment, features extracted using SURF were classified using SVM. The vector that contains the extracted descriptors was 128*20750 and divided to (128×16600) for training and (128×4150) for testing and achieved $94,18\%$ sensitivity and $83,3\%$ specificity.

- Experiment 4

  In this experiment, features extracted using SURF were classified using KNN with k=2. Descriptors were the same as Experiment 3 and achieved $92,48\%$ sensitivity and 89% specificity.

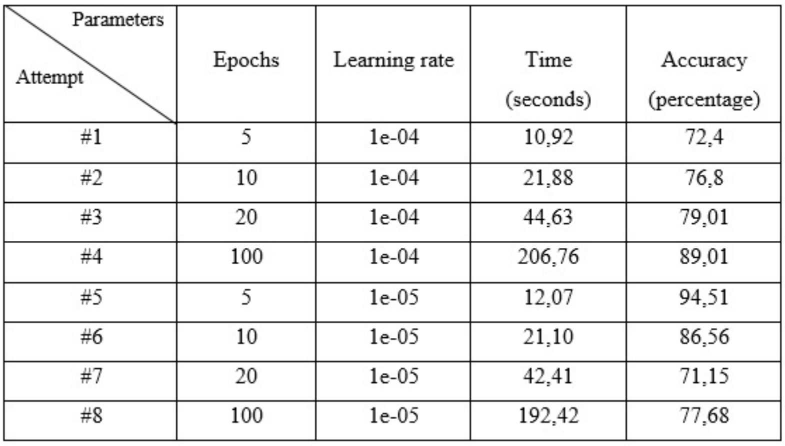


Features extraction method / Classifier	SIFT		SURF	
	SE	SP	SE	SP
SVM	86%	92%	94,18%	83,3%
KNN	92,81%	91%	92,48%	92%

Table 1: Sensitivity and Specificity of Experiments 1, 2, 3, 4

- Experiment 5

  In this experiment, the modified pretrained neural network, AlexNet, was used. As input, ALexNet accepts the full images. Several attempts were made in order to tune the parameters of the network.

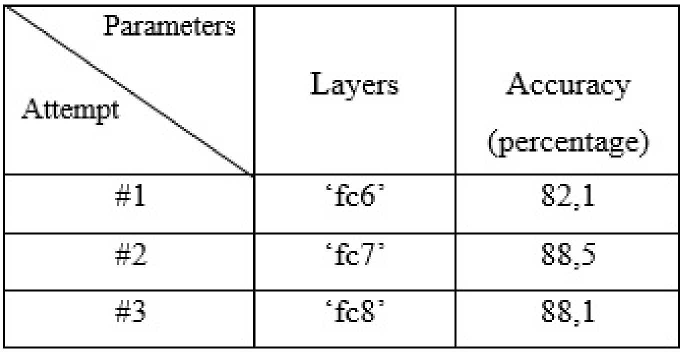


Parameters Attempt	Epochs	Learning rate	Time (seconds)	Accuracy (percentage)
#1	5	1e-04	10,92	72,4
#2	10	1e-04	21,88	76,8
#3	20	1e-04	44,63	79,01
#4	100	1e-04	206,76	89,01
#5	5	1e-05	12,07	94,51
#6	10	1e-05	21,10	86,56
#7	20	1e-05	42,41	71,15
#8	100	1e-05	192,42	77,68

Table 2: Accuracy of Experiment 5

- Experiment 6

  In this experiment, AlexNet was used to define the feature descriptor and classification was performed using SVM. The same training and test set as Experiment 5 was used here as well.

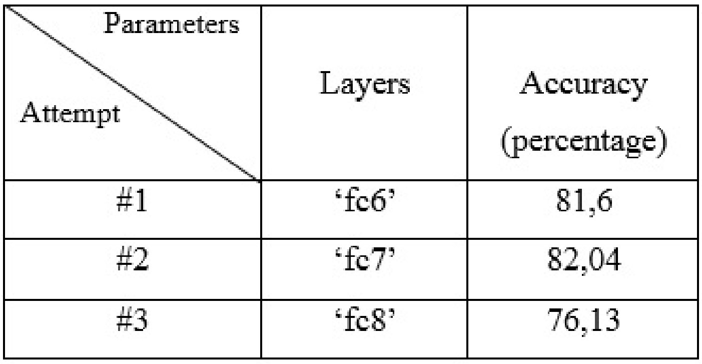


Parameters Attempt	Layers	Accuracy (percentage)
#1	'fc6'	82,1
#2	'fc7'	88,5
#3	'fc8'	88,1

Table 3: Accuracy of Experiment 6

- Experiment 7

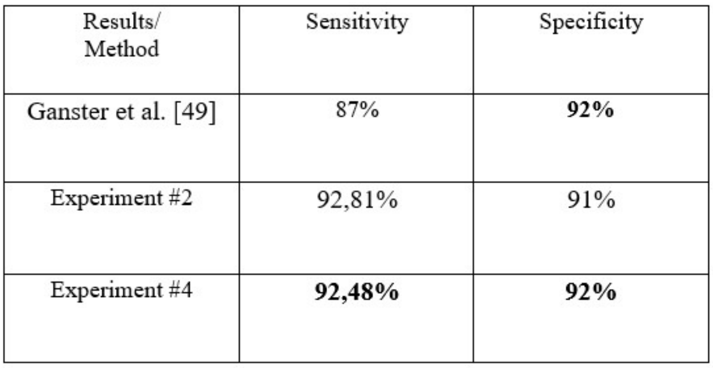
  In this approach AlexNet was used for feature extractor and a KNN was used as classifier with k=2. The same training and test set as Experiment 5 was used here as well.



Parameters Attempt	Layers	Accuracy (percentage)
#1	'fc6'	81,6
#2	'fc7'	82,04
#3	'fc8'	76,13

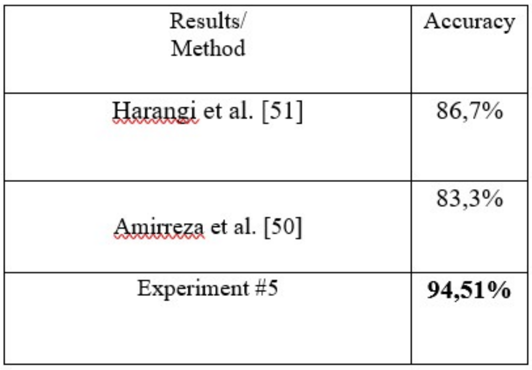
Table 4: Accuracy of Experiment 7

Based on the results, it will be useful to compare them with previous works on the same subject.



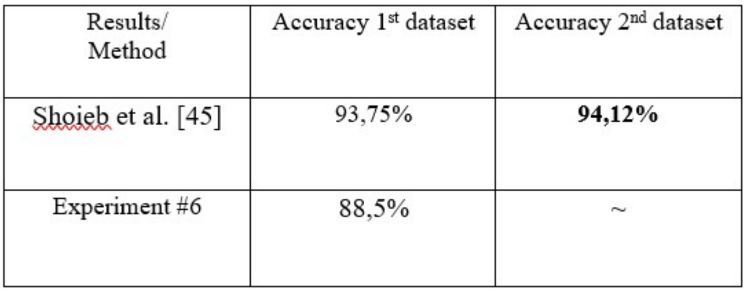
Results/ Method	Sensitivity	Specificity
Ganster et al. [49]	87%	**92%**
Experiment #2	92,81%	91%
Experiment #4	**92,48%**	**92%**

Table 4: Comparison of results from previous works and this master thesis 4 first experiments



Results/ Method	Accuracy
Harangi et al. [51]	86,7%
Amirreza et al. [50]	83,3%
Experiment #5	**94,51%**

Table 5: Comparison of results from previous works and this master thesis experiment 5



Results/ Method	Accuracy 1st dataset	Accuracy 2nd dataset
Shoieb et al. [45]	93,75%	**94,12%**
Experiment #6	88,5%	~

Table 6: Comparison of results from previous works and this master thesis experiment 6

In the automated classification of malignant lesions such as melanoma, correct classification of cancer cases has high importance. For that, high sensitivity is considered as the priority measure.
Observing the results it is obvious that the approach with extracting features using SURF and classify them using a K-NN achieved higher performance than the previous work. Also, the achieved accuracy using the remodeled deep neural network AlexNet is higher than previous works with similar characteristics. As it is obvious in this neural network, few iterations are enough to be re-trained well, as the rest layers remain the same so the only actual different will be the classes that it can recognize from now on.

Chapter 7

Conclusion

According to the importance of an early and quick diagnosis of melanoma, the aim of this master thesis was to develop a system for classifications of skin lesions images. After reviewing the literature on the computer science applications related to melanoma diagnosis, it was really challenging to try these methods and to try to improve them. Analysing images of skin lesions in images, accurate detection of malignant lesions and providing good segmentation of the lesions are important and equivalently challenging steps of automatic melanoma diagnosis. In this research, the images are processed by applying a threshold on them and then define the lesion area using an edge detector.

After, several experiments had been made but through experimental results the better way to treat Vienna dataset and classify the images is to use a Convolutional Neural Network (in this case, the pretrained AlexNet) and modify it so it can recognize the required classes for the dataset. Following this procedure, $94,51\%$ accuracy achieved that outperforms other state-of-art methods.

7.1 Future work

In Computer-Aided Design systems that deal with humans and deadly diseases, it is important to be as more specific as it can be. For that, more tests and analysis using larger datasets are mandatory.

There is also room for improvement by analyzing and applying different aspects of software such as balancing, hair removal, etc. To decrease diagnostic time of such a system, an interesting point is the developed software to be applicable in real time.

Bibliography

[1] Maciej Ogorzaek (2011), Modern Techniques for Computer-Aided Melanoma Diagnosis.

[2] Bernd Rechel (2016) , Hospitals in rural or remote areas: An exploratory review of policies in 8 high-income countries

[3] Robert M. Haralick (1985) , Image segmentation techniques

[4] Nikhil R Pal (1993) , A review on image segmentation techniques

[5] V. Prema (2016) , Brain cancer feature extraction using OTSU's Thresholding segmentation.

[6] K.S.Fu,J.K. Mui (1981), A survey on image segmentation

[7] Kimmi Verma (2013) , Image Processing Techniques For The Enhancement of Brain Tumor. Patterns

[8] Samta Gupta (2013) , Sobel Edge Detection Algorithm.

[9] Raman Maini , Study and Comparison of Various Image Edge Detection Techniques.

[10] Sameh Mohamed (2017) , Efficient Edge Detection Technique Based on Hidden Markov Model using Canny Operator.

[11] Jigisha M. Patel (2016) , A review on feature extraction techniques in Content Based Image Retrieval.

[12] Sebastin Castillo-Carrin (2017) , SIFT optimization and automation for matching images from multiple temporal sources .

[13] Tinne Tuytelaars , SURF: Speeded Up Robust Features .

[14] Alpaydin (2010), Introduction to Machine Learning pp.1-4.

[15] Alpaydin (2010), Introduction to Machine Learning pp.21-26.

[16] Warren S. McCulloch & Walter H. Pitts (1943), A logical calculus of the ideas immanent in nervous activity.

[17] Russell C. Eberhart & Yuhui Shi (2007), Computational Intelligence: Concepts to Implementations pp. 151-153

[18] Rumelhart (1986), Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations.

[19] Goldberg (1989), Genetic algorithms in search, optimization and machine learning

[20] Dasgupta & Nio (2008), Immunological computation: theory and applications.

[21] Dorigo & Sttzle (2004), Ant Colony Optimization.

[22] Quinlan (1986), Induction of Decision Trees.

[23] Muggleton (1994), Inductive logic programming pp.5-11.

[24] Pawlak (1991), Rough Sets-Theroterical Aspects of Reasoning about Data.

[25] Agrawal & Srikant (1994), Fast Algorithms for Mining Association Rules.

[26] Alpaydin (2010), Introduction to Machine Learning pp.177-179.

[27] Cleveland & Devlin (1988), Locally Weighted regression:an approach to regression analysis by local fitting.

[28] Kolodner (1993), Case-based reasoning.

[29] Duda (2000), Pattern classification.

[30] Jensen (1996), Bayesian networks basics.

[31] Dempster (1977), Maximum Likelihood from Incomplete Data via the EM Algorithm.

[32] Baum (1970), A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains.

[33] Chellappa & Jain (1993), Markov Random Fields: Theory and Applications.

[34] Shawe-Taylor & Crisitianini (2004), Kernel Methods for Pattern Analysis pp.25-82.

[35] Boser (1992), A Training Algorithm for Optimal Margin Classifiers.

[36] T.S. Jaakkola & D. Haussler (1991), Probabilistic kernel regression models.

[37] Shawe-Taylor & Crisitianini (2004), Kernel Methods for Pattern Analysis pp.252-286.

[38] Chapelle (2006), Semi-Supervised Learning.

[39] A. H. Song and Y. S. Lee,(2013), Hierarchical Representation Using NMF.

[40] B. A. Olshausen,(1996), Emergence of simple-cell receptive field properties by learning a sparse code for natural images.

[41] R. Collobert, (2011), "Deep Learning for Efficient Discriminative Parsing,"

[42] Daniel Graupe , Principles of Artificial Neural Networks.

[43] Raul Rojas, Neural Networks: A Systematic Introduction.

[44] Samer Hijazi , Using Convolutional Neural Networks for Image Recognition.

[45] Doaa A. Shoieb (2016) , Computer-Aided Model for Skin Diagnosis Using Deep Learning .

[46] Nicols Mnera (2016) , Human features extraction by using anatomical and low level image descriptors from whole body images .

[47] Aman Kataria (2013) , A Review of Data Classification Using K-Nearest Neighbor Algorithm .

[48] Ning Situ (2010) , Modeling spatial relation in skin lesion images by the graph walk kernel .

[49] Ganster(2001), Automated melanoma recognition.

[50] Amirreza Mahbod (2017) , Skin Lesion Classification Using Hybrid Deep Neural Networks.

[51] Balazs Harangi (2017) , Skin Lesion Detection Based On an Ensemble of Deep Convolution Neural Networks .

[52] H. Ganster,P. Pinz,R. Rohrer (2001), Automated melanoma recognition.

[53] Omkar Shridha rMurumkar (2015) , Feature Extraction for Skin Cancer Lesion Detection.

[54] Sujaya Saha , An Automated Skin Lesion Diagnosis by using Image Processing Techniques .

[55] Stephen Milborrow (2014) , Active shape models with SIFT descriptors and MARS

[56] Herbert Bay (2008) ,Speeded-Up Robust Features (SURF).

[57] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton.(2012), ”Imagenet classification with deep convolutional neural networks.”