جامعة الأخوين

ⵜⴰⵙⴷⴰⵡⵉⵜ ⵏ ⵓⵅⴰⵡⴰⵢⵏ

AL AKHAWAYN UNIVERSITY

SCHOOL OF SCIENCE AND ENGINEERING

MELANOMA CANCER DETECTION USING IMAGE PROCESSING

Capstone Design Final Report

Fall 2021

Mehdi Karmouche

Supervised by:
Naeem Nisar Sheikh

MELANOMA CANCER DETECTION USING IMAGE PROCESSING

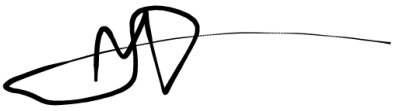
Capstone Report

Student Statement:

I, Mehdi Karmouche, have applied ethics to the design process and in the selection of the final proposed design. Moreover, I held the safety of the public to be paramount and has addressed this in the presented design wherever may be applicable.

_____

Mehdi Karmouche



Approved by the Supervisor(s)

_____

Dr. Naeem Nisar Sheikh

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude towards my capstone supervisor, Naeem Nisar Sheikh, who continuously helped and guided me throughout this project by brainstorming ideas and suggesting efficient ways of solving problems. I would like to thank the AUI faculty that equipped me with the necessary knowledge and skills for me to complete this milestone as a computer science student.

Moreover, I would like to thank my family who believed in me since the beginning of my journey at Al Akhawayn University in Ifrane and supported me mentally and financially. Thank you for all your sacrifices and effort you have done for me.

CONTENTS

ABSTRACT IN ENGLISH

Image processing is the practice of making different operations on images in order to enhance or extract important features from them. Image processing has a very important role in the medical field. Dermatologists can make use of the advancements in the image processing and machine learning field to have better insights when it comes to disease diagnosis such as melanoma. The latter is a deadly skin cancer that affects both males and females at any age.

The main purpose of this project is to develop software that would automatically classify images into two categories: melanoma and not melanoma. To achieve this goal, I used an approach composed of the following different phases. First, find a labeled dataset with melanoma and non-melanoma images. Second, pre-process, segment, post-process, and extract the main features of the images. Third, train a classification model for the images previously processed.

Developing such a solution would allow dermatologists and medical professionals to get the opinion of the machine on the case of a specific patient that could potentially have melanoma. With that help, dermatologists are expected to make a better, quicker, and cheaper diagnosis for patients.

ABSTRACT IN FRENCH

Le traitement d'images consiste à effectuer différentes opérations sur des images afin d'en améliorer ou d'en extraire des caractéristiques importantes. Le traitement d'images joue un rôle très important dans le domaine médical. Les dermatologues peuvent utiliser les progrès réalisés dans le domaine du traitement de l'image et de l'apprentissage automatique pour mieux comprendre le diagnostic de maladies telles que le mélanome. Ce dernier est un cancer de la peau mortel qui touche aussi bien les hommes que les femmes, à tout âge..

L'objectif principal de ce projet est de développer un logiciel qui classerait automatiquement les images en deux catégories : mélanome et non mélanome. Pour atteindre cet objectif, j'ai utilisé une approche composée de trois différentes phases. Premièrement, trouver un dataset étiqueté avec des images de mélanome et de non-mélanome. Deuxièmement, prétraiter, segmenter, post-traiter et extraire les principales caractéristiques des images. Troisièmement, former un modèle de classification pour les images précédemment traitées.

Le développement d'une telle solution permettrait aux dermatologues et aux professionnels de la santé d'obtenir l'avis de la machine sur le cas d'un patient spécifique potentiellement atteint d'un mélanome. Grâce à cette aide, les dermatologues devraient pouvoir établir des diagnostics plus précis, plus rapides et moins coûteux pour les patients.

LIST OF FIGURES

1 INTRODUCTION

The act of using computers to perform various operations on digital images in order to enhance or extract essential features from them is known as digital image processing. In the medical field, image processing is employed more and more used to help doctors detect various types of disorders. For example, dermatologists (skin specialists) are using image-processing based systems for the diagnosis of skin cancers. With the help of artificial intelligence, computers are able to make as good predictions as dermatologists. In 2018, an artificial intelligence system outperformed dermatologists when it comes to classifying skin lesions as benign or malignant by more than 8% [7].

Melanoma skin cancer is one of the deadliest cancers and one of the most common cancers in the world since many countries don't officially record melanoma cases [1]. Moreover, the rate of melanoma cases increased by 44% from 2008 to 2018 with a significant increase in fatalities [1]. Melanoma is the world's 19th most prevalent cancer, with roughly 300,000 cases around the world in 2018. Morocco recorded 248 new melanoma cases and 114 fatalities in 2020 [1], according to the World Health Organization. Excessive exposure to ultraviolet light, a weakened immune system, or a family history of melanoma can all lead to this disease.

Traditionally, dermatologists check the following characteristics of the skin lesion: asymmetry, borders, colors, diameter, and elevation [3]. If the lesion of the patient is asymmetric, has fuzzy borders, has more than four colors, has a large diameter, and its shape evolves during time then it is probably a melanoma case. These characteristics are together called the "ABCDE rule". Dermatologists usually check these characteristics to get an initial insight, then they proceed to do a skin biopsy if they are incapable of drawing a clear conclusion [3]. This way of diagnosing melanoma is time-consuming and can be expensive for some individuals. Moreover, the skin biopsy usually leaves a visible scar in the skin since a major part of the skin lesion is taken to be tested [3].

This capstone project attempts to automate the identification of melanoma skin cancer based on raw images of skin lesions in order to create a faster and less expensive method of detecting this disease without leaving a scar. The project's findings are aimed to help dermatologists and other professionals make more informed judgments, but they are not meant to take the place of their professional judgment. Furthermore, this project would enable dermatologists to see more patients each day, work less, and concentrate on the most critical cases.

To reach this goal, it is necessary to design the software solution upfront. The solution I opted for contains four major parts. First, collect a labeled dataset of raw melanoma and non-melanoma images. Second, use image pre-processing techniques to enhance the quality of the images, segment the lesion from the surrounding skin, post-process the images by implementing morphological operations and removing isolated pixel islands. Third, extract the most important features from the images. Fourth, feed the retrieved features to a machine learning model that can accurately detect melanoma and non-melanoma cases after enough training.

I used a variety of tools and resources to create my software solution. I picked Python for the project's overall development since it is simple to use thanks to all of the libraries, packages, and open source support it provides. I utilized PIL and OpenCV, two well-known libraries in the field of computer vision and image processing, to pre-process my image dataset. I also used Scikit-Learn to train on the transformed images and implement the K-nearest-neighbors machine learning algorithm. Finally, I used Git and GitHub to save and track the various versions of my code in the cloud.

2 FEASIBILITY STUDY

A feasibility assessment was done prior to the start of this project. This research looked into the tools and resources that would be required to determine whether or not the project was feasible. Dataset collection, pre-processing, segmentation, post-processing, feature extraction, and classification are the phases that the project is separated into. I had to manage my time over the semester and invest it in researching and learning about the image processing field because these responsibilities were new to me. I planned to use the first few days of the first week to look for interesting literature on Google Scholar. I also found interesting articles about projects that worked on similar tasks on the internet especially on an open platform entitled Medium. Furthermore, the remainder of the semester, or 8 weeks, was set aside for implementation. Obtaining a dataset took one week. Pre-processing and segmentation took two weeks. Post-processing took two weeks, feature extraction took two weeks, and classification took one week. Given the time constraints, I knew this project could be completed. Various tools and resources were planned to be used on the technical side of this capstone project. I chose Python as the main programming language for implementing this project because it is easy to use and offers many libraries that can help with image processing and machine learning tasks. PIL, for example, is a library that can be used in Python to resize, enhance, and pre-process images by using the interface provided. Moreover, advanced image processing and computer vision algorithms and techniques are all implemented in a library called OpenCV which stands for Open Source Computer Vision. Furthermore, saving the code and the dataset somewhere secure and accessible anywhere and anytime was a priority. For that reason, I decided to use GitHub to save my code to the cloud and use it on my local machine whenever I needed to. Even though the main focus of this project is not machine learning, I looked for libraries that offer users the possibility to use ready machine learning models. I found Scikit-Learn and opted to use it in my project. Finally, I decided to use my own machine that runs on a Windows operating system and it provides 8 GB of RAM and enough storage space. With all the previous time and technical resources, I was sure that I had access to all that I needed to complete the project on time.

3 LITERATURE REVIEW

Over the years, many computer scientists and researchers have worked on employing image processing and machine learning approaches to identify melanoma. Reading their work and being inspired by it was a significant part of the process of creating this capstone project. My supervisor and I were mostly inspired by [4] and [5]. These publications discuss and describe how to use image processing techniques to detect skin cancer early in great detail. Furthermore, these papers refer to other publications and review their methodologies which helped me tremendously in comprehending the project's challenges and procedures. In my project, I was able to implement some of the techniques used in those papers but also creatively came up with new ideas with my supervisor.

The research [4] proposes a generic technique that consists of the following phases to make early melanoma identification faster, more accurate, less painful than skin biopsy, and without leaving a scar on the skin. The authors suggested that the photos should be pre-processed by eliminating hair and noise and enhancing the contrast. After pre-processing the pictures, the lesion must be segmented using a variety of methods, including edge-based, morphological, and region-based segmentation. Furthermore, after lesion segmentation, features must be extracted. Asymmetry, diameter, boundaries, and compactness should all be taken from the images, according to the authors. In addition, the extracted features are saved in a dataset or a data frame so that a machine learning model can train on them. The last step is to predict if a new image refers to skin cancer or not.

In [5], it is stated that a dermatologist's accuracy when it comes to diagnosing melanoma using dermoscopy is 75% to 84% and that a computer-aided system can improve that accuracy. The overall methodology is similar to what was stated in [4], however, they differ in some pre-processing techniques and in the features to be extracted. In [5], the authors suggest enhancing the brightness, using automatic thresholding to segment the lesion from the surrounding skin, extracting the area of the lesion and perimeter, and using those data points to calculate the circularity and irregularity index.

I was able to gain a thorough understanding of how such projects are carried out, which assisted me in developing my own methodology that is tailored to this specific capstone project.

4 DESIGN AND IMPLEMENTATION

4.1 GENERAL STRUCTURE

Before beginning the project, it was necessary to create a process that outlined the many phases that must be completed to finish the project. The following is the overall structure. A color image is used as the system's input. The images are then pre-processed to reduce noise and enhance contrast. Furthermore, the images are given to functions that use various segmentation algorithms and techniques to separate the lesion from the surrounding normal skin. Furthermore, images are post-processed to remove isolated pixels and focus on the focal lesion. Following the processing and the segmentation of the images, the feature extraction phase begins, in which all of the image's significant features, such as the number of colors, diameter, and border fuzziness, are extracted and saved in a data frame. Following the application of these procedures to a large number of images, the data frame containing the extracted features is supplied to a machine learning model, which trains on it and predicts if a new image is a melanoma or not.

4.2 IMAGE REPRESENTATION

Understanding how computers represent images is critical. Images are, in fact, a two or three-dimensional array of pixels. If the image has different channels, it is represented as a 3-D matrix. If the image is in grayscale it is represented as a 2-D dimensional matrix. Each pixel has a value between 0 and 255, resulting in a total of 256 possible pixel values in that interval. The column and row numbers are used to identify each pixel. All colored digital photos are made up of a combination of red, blue, and green. These are known as RGB channels. Each pixel in these photos has three values that indicate how much red, blue, and green are present on that particular pixel. Figure 1 shows how an RGB image is represented by a computer.

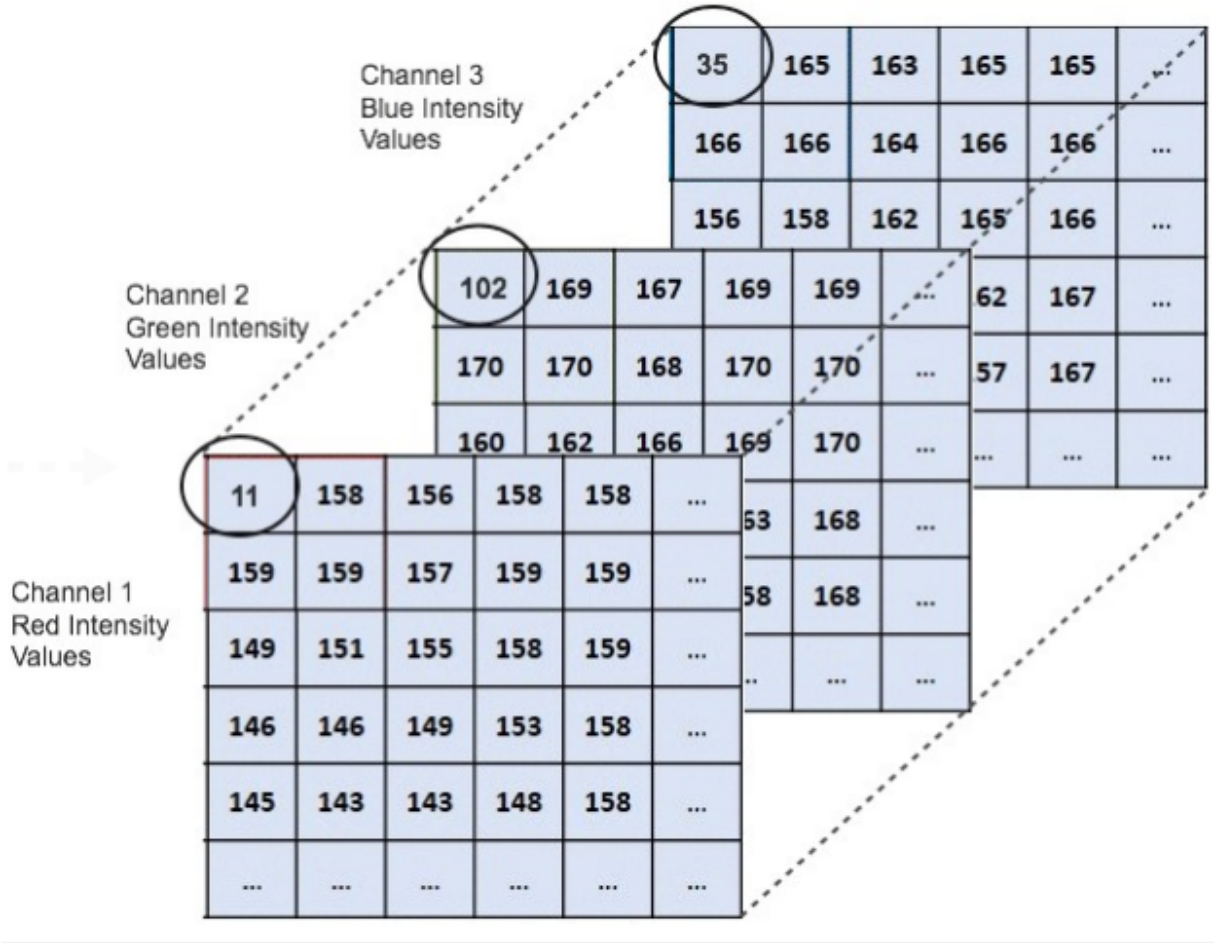


Figure 1: RGB image representation (*source: https://freecontent.manning.com*)

Sometimes, one needs to use any image without lots of colors, and for that reason, grayscale images can be used. A grayscale image is one with only one channel, which depicts the light intensity in pixels. Grayscale images only show pixels that are white, black, and grey. Figure 2 illustrates how a grayscale image is represented by a computer.

170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Figure 2: Image representation of a grayscale image

*(source:*https://www.researchgate.net/figure/Matrix-for-certain-area-of-a-grayscale-image-17_fig3_325569674*)*

4.3 DATASET

The success of this initiative hinges on the availability of a suitable dataset. Kaggle is an online platform that provides access to a large number of datasets and allows data scientists to share their datasets with the general public. Many melanoma datasets were discovered after a comprehensive search in Kaggle. However, not all of the datasets were labeled, so I couldn't identify whether a certain image depicted a melanoma instance or not. Only a few datasets had high-quality labeled photos. Images were categorized as "melanoma" or "not melanoma" in the dataset used for this experiment. Furthermore, this collection comprises raw images that have not been enhanced or changed since they were taken. The dataset contained initially 1113 melanoma images and 8902 non-melanoma images. Data augmentation was applied to the dataset by the maintainers to increase the number of images in both classes to have more images and have an equal number of images in each class. Data augmentation is a technique used to increase the amount of data by modifying moderately the currently available data. In the case of the dataset used in this capstone project, the images were cropped and rotated by different angles. It was important to augment the data because the machine learning models trained on different and more cases. For example, a melanoma lesion pointing to the right is the same as one that is pointing to the left [6].

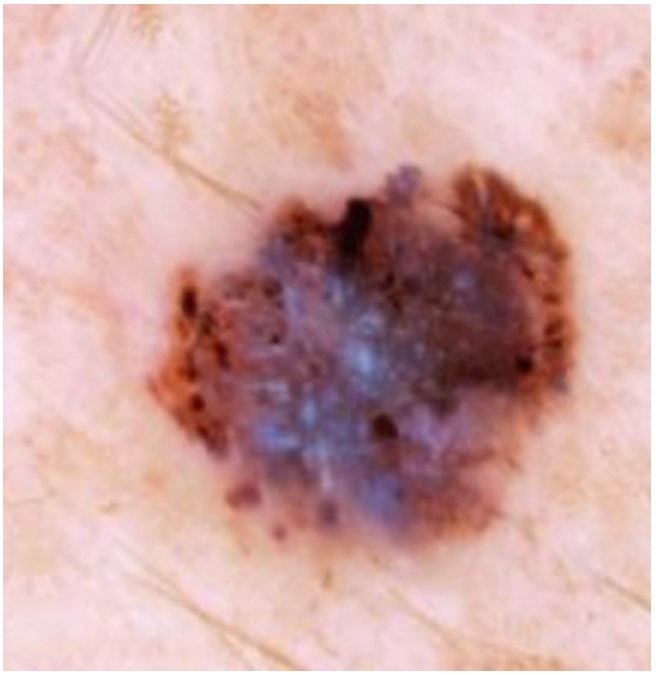


Figure 3: Sample image of a melanoma skin lesion from the chosen dataset

Figure 4: Sample image of a non-melanoma skin lesion from the chosen dataset

4.4 PRE-PROCESSING

The goal of this part of the phase is to improve the quality of the image by enhancing the contrast and removing noise. To enhance the contrast of the images, Contrast Limited

Adaptive Histogram Equalization (CLAHE) was used. To remove the noise, Gaussian and median filters were both used and implemented using the OpenCV library.

4.4.1 CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION (CLAHE)

Histograms are a visual representation of the colors in an image [8]. The intensity of each and every pixel in the image is described by the histogram [8]. The histograms of the dataset images in this project are frequently focused on a small range. It is possible to extend the histogram to hold a considerably bigger range using Contrast Limited Adaptive Histogram Equalization, resulting in a greater difference between dark and bright pixels and the ability to notice the various shapes and forms present in the image [9]. The CLAHE algorithm is used once the photos have been converted to grayscale. This technique equalizes and expands the image histogram while generating minimal noise. This technique is used to improve the contrast of underexposed or overexposed images. When compared to other histogram equalization methods, CLAHE separates the image into a huge number of little matrices and equalizes the histogram of each one, resulting in reduced noise.

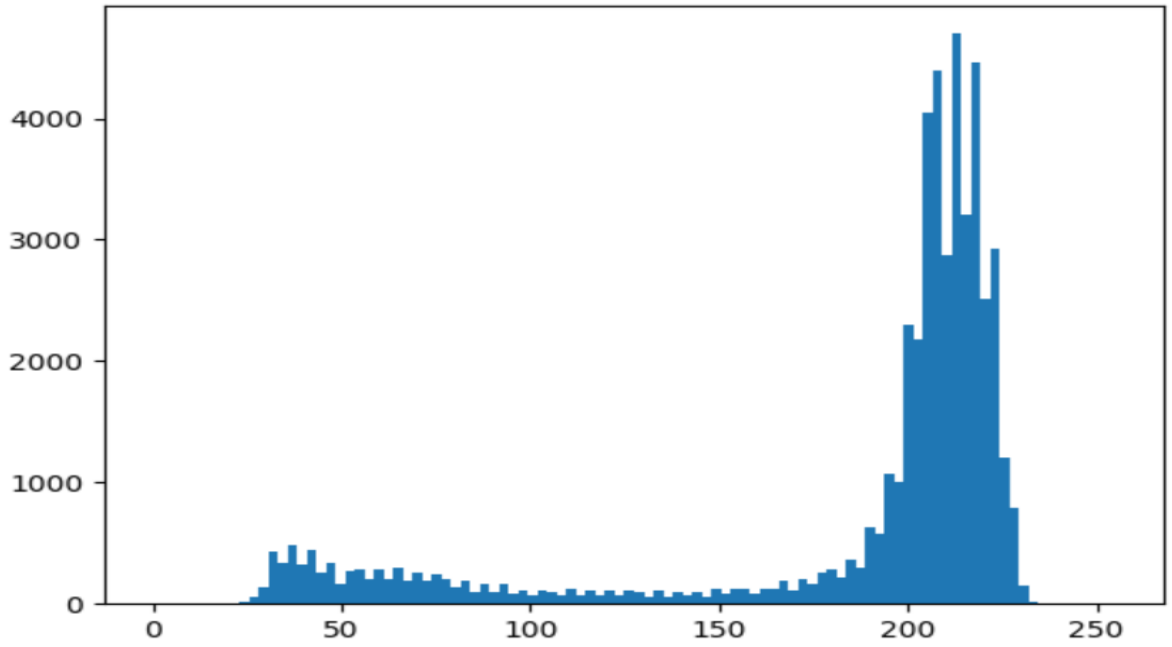


Figure 5: Histogram of a melanoma image in grayscale

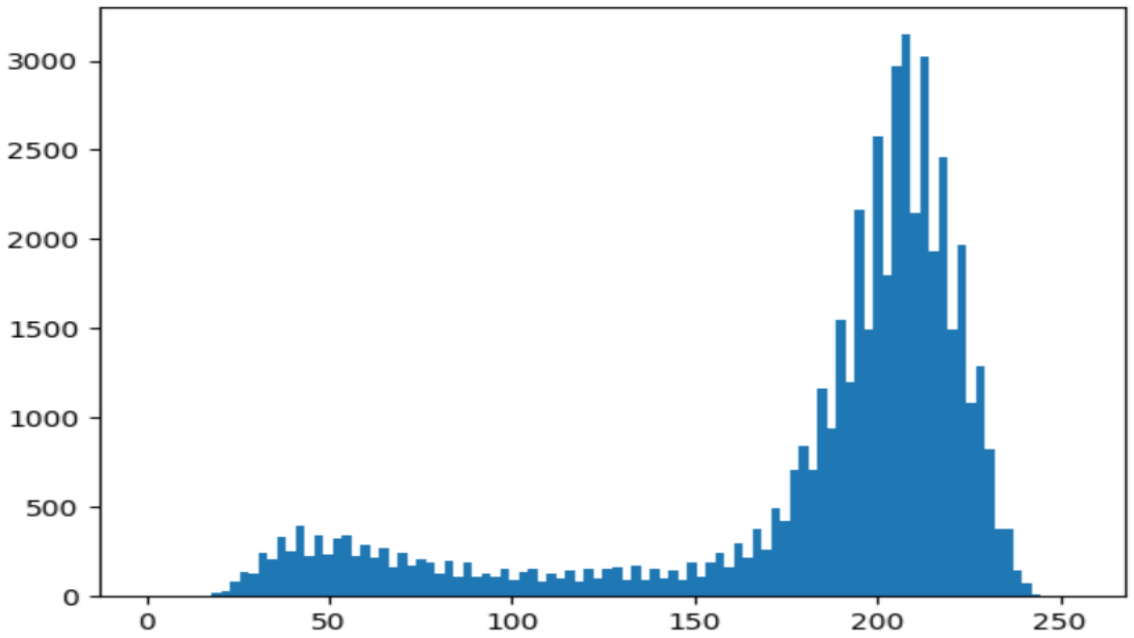


Figure 6: Histogram of melanoma skin lesion image after applying CLAHE

Figures 5 and 6 show how the histogram has changed significantly when it grew more stretched. The pixels' frequency has been stretched out.

Figure 8 shows how the original image has changed dramatically after applying CLAHE. The white pixels became brighter while the black pixels went darker. Furthermore, the human eye can see the different shapes and small details both inside and outside the lesion. However, there is a lot more noise in the image, notably outside the lesion's border.

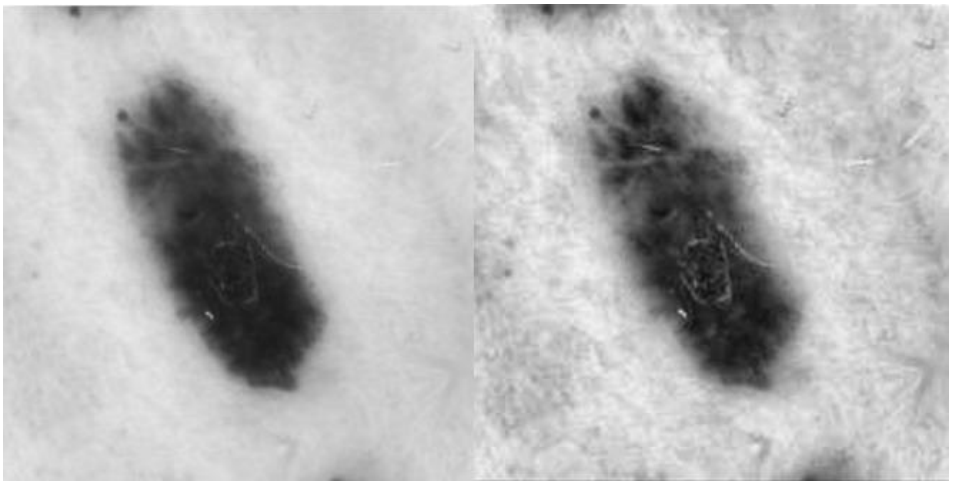


Figure 7: Image before (left) and after (right) applying CLAHE

4.4.2 GAUSSIAN FILTER

Filters are used to reduce noise, smooth images, and detect edges. A kernel has a pre-defined size. The kernel is usually positive and odd in size. The Gaussian filter is a technique for smoothing images and removing noise in the image processing field. This method entails creating a kernel that is both positive and odd in size just like any other filter. The Gaussian filter uses a kernel that is normally distributed. This signifies that the kernel's center is highly bright (pixel value close to 255), while the edges are dark (pixel value close to 0) [10]. The normal distribution of the gaussian kernel is seen in Figure 8.

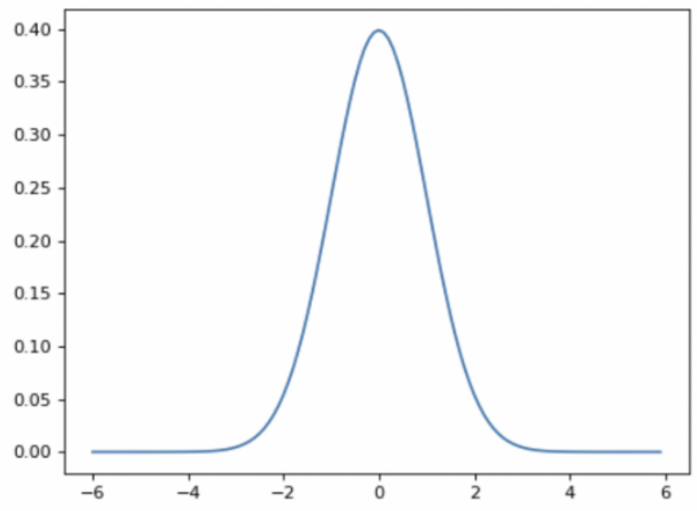


Figure 8: Normal distribution of the gaussian kernel *(source: https://medium.com/jun-devpblog/cv-2-gaussian-and-median-filter-separable-2d-filter-2d11ee022c66)*

The Gaussian filter can be used with the OpenCV library in Python. The input picture and the kernel size are passed as input parameters to the gaussian filter function [11]. The input image to the gaussian filter is the result of applying Contrast Limited Adaptive Histogram Equalization to the image. Figure 9 shows the outcome of employing a gaussian filter. and how the image has become blurred and smoother, with less noise.

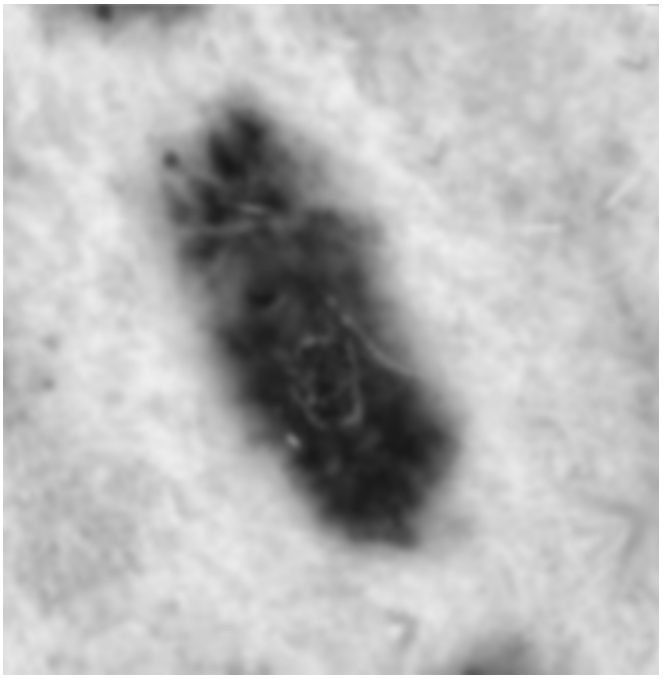


Figure 9: Melanoma skin lesion after applying Gaussian Filter

4.4.3 MEDIAN FILTER

Another method for reducing the noise in the image produced by the Contrast Limited Adaptive Histogram Equalization algorithm can be utilized. As an alternative to the Gaussian filter, the median filter was used. A median filter is a non-linear approach in which a kernel or a window slides over all of the pixels of the image's matrix and the median of the adjoining pixels is substituted for the center of the kernel in each iteration [12]. Before determining the median element, the surrounding pixels are sorted first [12]. Figure 10 depicts a median blur example.

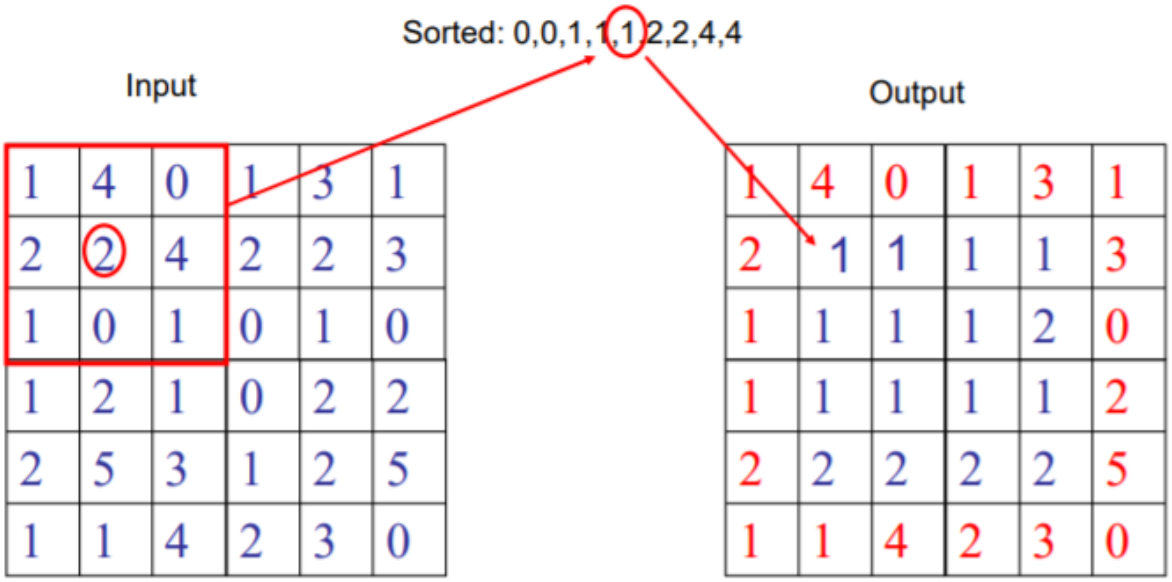


Figure 10: Example of median filter *(source: https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Image%20Filtering.pdf)*

Figure 11 shows the resulting image after using the median filter algorithm from the OpenCV library. I opted to use the Gaussian filter instead of the median filter because it preserves the edges better.

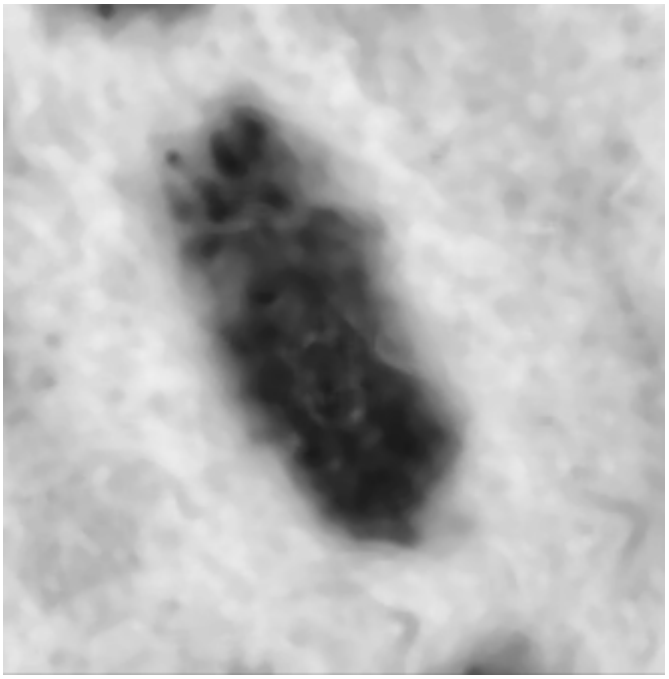


Figure 11: Image after applying the median filter

4.5 SEGMENTATION

This section of the project's goal is to isolate the lesion from the surrounding skin. This phase is critical because it must preserve all of the lesion's properties and characteristics so that they may be extracted during a later stage. If a segmentation method causes harm to the images by changing the lesion's shape, for example, feature extraction will fail to extract the information effectively, and the machine learning model will learn on incorrect data, resulting in incorrect predictions. Different segmentation methods, such as K-means clustering, edge-based segmentation using canny, and adaptive thresholding, were applied. Experimenting with these alternative algorithms was a critical step because it allowed me to learn about their performance, benefits, and drawbacks. Because the morphology of each lesion varies from image to image, segmenting the lesion is a difficult task. Some lesions have sharp edges, while others have fuzzier margins. Furthermore, some lesions can be broken down into small condensed blobs of pixels, and others include very small regions of healthy skin within the central lesion.

4.5.1 K-MEANS CLUSTERING

Clustering is the task of dividing data into different sections or groups. Each data part should have items that are somewhat comparable to one another[13]. The similarity between the elements of the same partition should be greater than the similarity between the partitions [13]. The Euclidean or hamming distance can be used to calculate the similarity between data points. The Euclidean distance is calculated as illustrated in figure 12, where x1 and y1 are the first data point's coordinates and x2 and y2 are the second data point's coordinates.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Figure 12: Euclidean distance

The hamming distance is the number of differences between two strings or numbers. For example, the word “take” and “make” have a distance of 1 since the letter “t” is the only different letter. Another example would be to consider the following binary numbers: 100101 and 110100. These two numbers have 2 as the distance between them. Figure 14 shows a visual example of the hamming distance of different binary numbers.

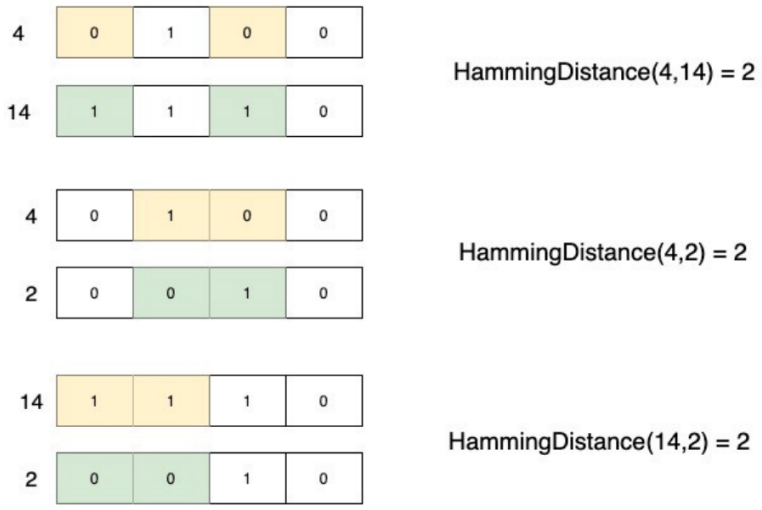


Figure 13: Hamming distance example (source: *https://medium.com/geekculture/total-hamming-distance-problem-1b74decd71c9*)

K-means clustering is one of the most common clustering algorithms. This algorithm consists of grouping data points into K clusters [13]. The number K is given as an input to the algorithm along with the data [13]. Then the algorithm keeps assigning each data point to a certain group while calculating the centroid position of the group accordingly. The centroid of a cluster in the K-means algorithm refers to the center of the cluster. When inserting a new element into a cluster, the position of the centroid is re-calculated and usually changes. The algorithm stops when the centroids seize from changing. It is important to mention that the number K should be wisely chosen and it should suit the distribution of the data. In the case of this project, K-means clustering was used to segment the lesion from the surrounding skin using K equals 2. More specifically, the K-means algorithm was fed RGB pixels. In other words, the algorithm will cluster the image pixels into two groups which are skin and lesion since these two are very distinct. The resulting image is shown in figure 14. Other values of K were used, however, to segment the lesion from the surrounding skin, K equals 3 gave the best

result since the skin and the lesion have a significant difference in the color intensity. Using K equals 3 or more shows more details in the image especially in the regions that have more than two colors.

Figure 14: Result of segmentation using k-means

The lesion is in a dark shade in figure 14. However, on the edges of the images, there are islands of pixels that have the same color as the central lesion. At this stage, the isolated set of pixels can be ignored until the post-processing phase.

4.5.2 EDGE SEGMENTATION

As mentioned in the literature review, edge segmentation is one of many other techniques to separate a lesion from its background. Edge detection consists of searching the image matrix and aiming to find significant changes in the intensities between adjacent pixels. The radical change of pixel intensities describes the existence of a clear edge. However, sometimes the change in the intensity is not so large which means that the edge is not very clear and might in some cases not be considered an edge at all. Canny edge detection is a technique out of others that is used to find edges in an image [14]. This algorithm is very common in the image processing field as it gives better results than the Sobel operator for example [14]. Canny edge detector finds bold edges and then thinning them to have a one-pixel width [14]. Having thin edges is very useful because the goal in this phase of the project is to separate as accurately as possible the lesion from the surrounding skin without losing pixel information in the process.

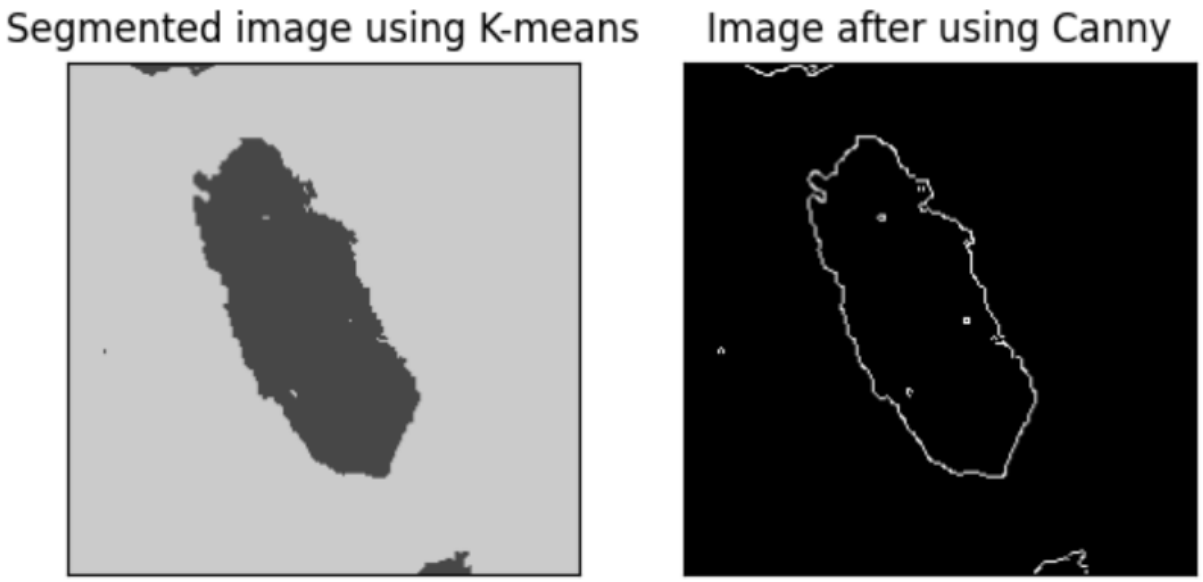


Figure 15: Using Canny edge detector on the image outputted by K-means

Using a Canny edge detector on the original image gives very low accuracy as the edges are all over the images. Applying the Canny edge detector after using K-means usually gives a good result depending on the shape of the lesion. Figure 16 shows the result of applying the Canny edge detector on a pre-processed image. It is quite obvious that the edges are not accurate even though the image was pre-processed and the noise was removed.

Figure 16: Resulting of applying Canny edge detection on the pre-processed image

4.5.3 ADAPTIVE THRESHOLDING

Image thresholding is another technique to segment an image into two major parts: background and foreground [15]. Usually, the output of an image thresholding algorithm is a binary image which means that it contains only two colors, black and white [15]. While the input of the algorithm is a grayscale image [15]. Static thresholding consists of considering a value T where all pixel values greater than T are replaced by 255 and all the pixel values less than T are replaced by 0 [16]. However, selecting the right T is problematic since the same value T may not be suitable for all the different regions of the same image. For that reason, adaptive thresholding was used. Adaptive thresholding solves the issue of having the same threshold value T as it considers a constant C and it calculates the mean of a sliding window and subtracts the two values which result in a local T value that would threshold that small matrix of the image. Figure 17 shows the difference between using simple thresholding and adaptive thresholding.

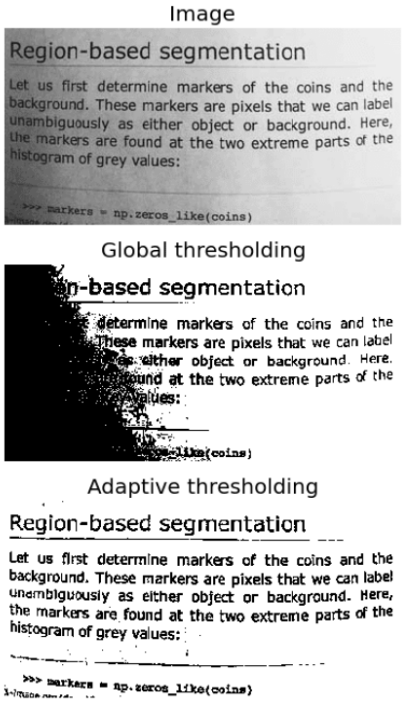


Figure 17: Difference between global and adaptive thresholding (*source:* *https://www.pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/*)

Using the adaptive thresholding with the constant C equals 7 on the pre-processed images gave a good result as illustrated in figure 18.

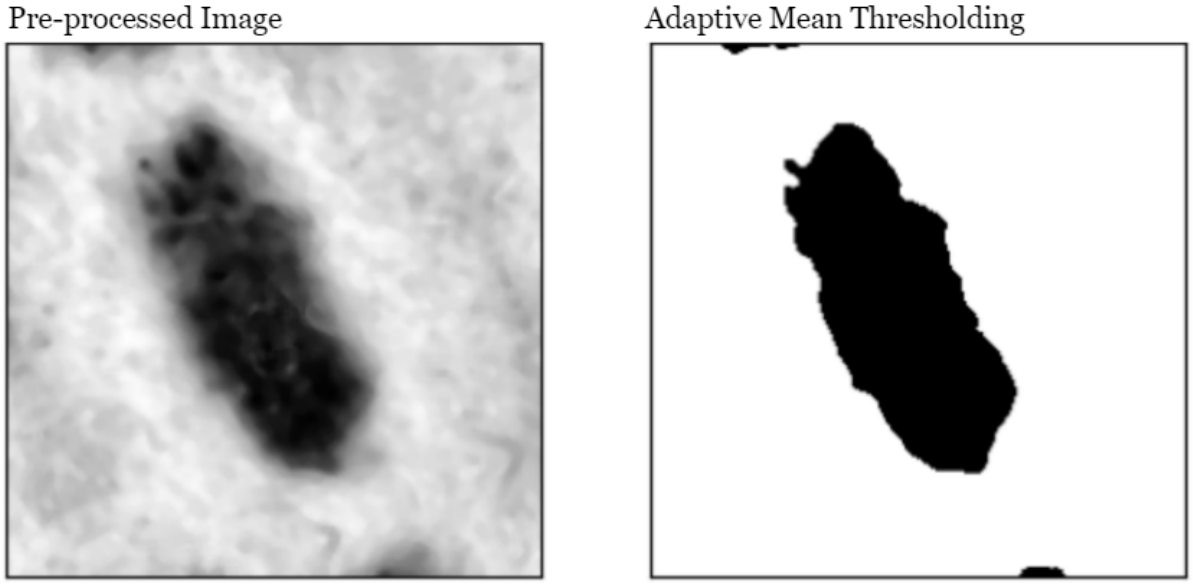


Figure 18: Using adaptive mean thresholding to segment the lesions

4.5.4 COMPARATIVE STUDY OF SEGMENTATION ALGORITHMS

To choose the best segmentation algorithm it was very important to experiment with different techniques and choose the one that has the best result. In figure 14, some pixels still exist on the inside of the central lesion after using K-means clustering. The same thing is noticed in figure 15 after using Canny edge detection. However, the adaptive thresholding gives a good result as it separates the lesion from its background perfectly while preserving edges and reducing noise to a maximum. Thus, adaptive thresholding was chosen as the best alternative when it comes to lesion segmentation.

4.6 POST-PROCESSING

At this stage of the project, the segmented images are not yet ready to be used for data extraction mainly because there are isolated islands of pixels. These islands usually make the data extraction very inaccurate. It is important to remove these pixel islands and leave only the central lesion as the main focus.

4.6.1 ISOLATED PIXEL ISLAND REMOVAL

As stated previously, the Adaptive Thresholding segmentation algorithm was used. However, the lesion is in black and the surrounding is in white. To be more concise, it is more appropriate to have the lesion in white and surrounded in black since the surrounding skin doesn't hold any information. Swapping the colors of the lesion and skin was completed before proceeding to remove isolated pixel islands. Figure 19 shows the result after the color swap.

Figure 19: Swapping the colors of the image

To remove the isolated blobs or islands of white pixels in the image, it was necessary to identify the small areas that contain white pixels and are surrounded by black pixels. When those areas or islands are found, the white pixels are replaced by black ones. The resulting image contains only the central lesion which is optimal for feature extraction. Figure 20 illustrates the result of these operations.

Figure 20: Image after removing isolated island pixels

4.7 FEATURE EXTRACTION

For a computer to make sense of an image, it is necessary to extract the characteristics of that image by analyzing it. By extracting the features from the images, the size of data shrinks significantly, thus, the data becomes easy to manipulate. In fact, dermatologists look for the following characteristics in a skin lesion in order to get an initial insight: asymmetry, area and perimeter, border circularity, number of colors existing inside the lesion, diameter, evolution of the lesion over time. In this project, the main focus was on extracting the asymmetry, border circularity, number of colors, and diameter. However, the evolution of the lesion over time cannot be computed as the dataset provides images at one specific point in time. This phase aims to simulate what the dermatologists do in the first diagnosis of skin cancer.

4.7.1 ASYMMETRY

Usually, if a lesion is asymmetric then it is more likely to be a melanoma case. To extract such a feature, four points are drawn on the lesion. The uppermost, down most, rightmost, and leftmost extremities are the places where the points are drawn. Having these four points enables us to calculate the Euclidean distance between these points and choose the maximum distance as the diameter. The resulting diameter is stored in a data frame.

4.7.2 AREA AND PERIMETER

The area of the lesion is also an important aspect since big lesions are more likely to be melanoma [3]. To extract the area of the lesion, the number of white pixels are simply needed to be counted. A nested loop is implemented to go through the pixel matrix and increase the count of the white pixels when encountered. The unit of measure in this case is the pixel. Unfortunately, this calculation cannot be very accurate since the scale of the dataset's images is not fixed. In other words, the distance between the camera and the lesion is a bit different from an image to another which means that the extracted area will not correspond fully to the actual area.

Furthermore, the perimeter is also a good feature to extract since it would enable calculating the border circularity or fuzziness. To extract the perimeter or the edges of the lesion, the

Canny edge detector, as previously explained, can be employed. The resulting image of the Canny edge detection algorithms highlights the edges in white. Thus, calculating the white pixels would yield to computing the overall perimeter of the lesion.

4.7.3 BORDER AND CIRCULARITY

Computing the area and the perimeter is useful because it allows us to calculate the border circularity. The latter is an aspect that dermatologists look at when trying to diagnose skin cancer. The more the lesion is not circular, the more likely it is to be a melanoma case. Figure 21 shows how to calculate the circularity ratio using the area and perimeter features already extracted. The closer the ratio is to 1, the closer the lesion is to a circle. If the ratio is close to 0, it means that the lesion is non-circular thus more chances for the lesion to be melanoma.

$$Circularity = (4 * PI * Area)/Perimeter^2$$

Figure 21: Circularity formula

4.7.4 NUMBER OF COLORS

Melanoma skin cancer is usually characterized by skin lesions with more than one color [17]. Hence the necessity of extracting the number of colors that exist on the image as a feature for the classification model. I made use of the Python library entitled "**extcolors"**. This library allowed me to use a function entitled **"extract_from_path"** that takes a path to an image as an input argument, then the function returns the most common colors in the image along with their pixel concentration. The length of the returned list of colors is the extracted feature since it is the number of the most common colors in a specific image. The number of colors is appended to a data frame along with the other previous features.

4.8 CLASSIFICATION

In machine learning, classification refers to the practice of classifying a certain data point into known classes or groups in order to develop a predictive model [18]. The classes are called targets. In this project, the targets are "melanoma" and "not melanoma". All of the extracted features that were stored in a data frame contain a column entitled "sick" that holds either zero or one. One means that the image represents a melanoma case. Zero means that the image

represents a non-melanoma case. After successfully running the program that pre-processes, segments, post-processes, and extracts the features on 200 images from both types of images, a large data frame was created. The data frame had to be shuffled in order to be random so that the machine learning model learns properly. The machine learning model that was used is the **KNN** or **the K-nearest neighbors** model**.** The latter predicts the target class of a data point based on the K neighbor data points. For example, in a two-dimensional feature space, to predict the target class of data point X, the KNN would start by looking at the closest data points until finding K neighbors. The class of the majority of the K neighbors data points is given to the X data point. The number K should always be odd and not a multiple of the number of classes. In my case, I chose K to be 3 since I have only 2 classes. Figure 22 illustrates how the KNN model works. The green data point will be classified as a triangle since there are two triangles and just one square when K equals 3. When the **KNN** model trains, it is then given a new image that was never seen before by the model and tries to predict its class.

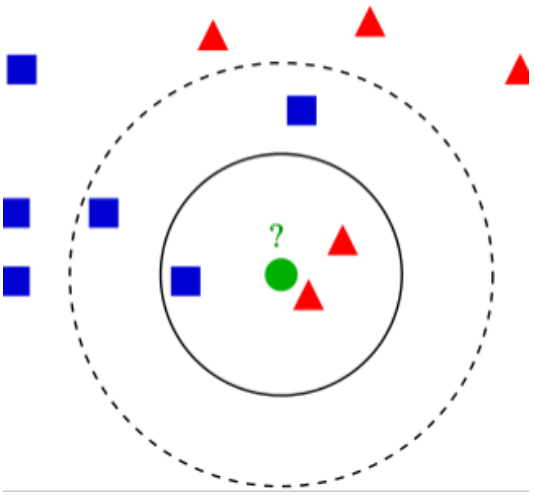


Figure 22: Example of KNN with K equals 3 (source: *https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm*)

5 RESULTS AND FUTURE IMPROVEMENTS

Since the aim of this capstone project is to assist health professionals and dermatologists in diagnosing melanoma, it is important that the machine learning model accurately predicts this disease. At least, the accuracy of the model should be equal or higher to the dermatologists'. In my case, the **KNN** model trained on 200 images from both melanoma and non-melanoma classes. The resulting dataset is divided into a training dataset and a test dataset. The training dataset constitutes 70% of the whole dataset. While the testing dataset constitutes only 30% of the overall dataset. This partitioning ensures that the model learns properly and enables us to evaluate the performance of the model. The accuracy of my model was 89% which is higher by 2.4% than the accuracy of dermatologists [7]. Figure 23 shows the result of the KNN model. However, it is important to highlight the fact that my model did not train on a large dataset that contains thousands of images since it takes a lot of computing power and time to process them.

```
27   296.892236                5.0  0.267330   1.0
172  409.542428                2.0  0.193244   0.0
68   183.043711                2.0  0.059153   1.0
56    60.671245                3.0  0.413268   1.0
135   67.230945                3.0  0.606149   0.0
..          ...                ...       ...   ...
76   262.773667                3.0  0.132100   1.0
78    88.865066                3.0  0.373834   1.0
125  598.000000                4.0  0.054004   0.0
67    80.777472                2.0  0.726470   1.0
44   314.663312                4.0  0.064795   1.0

[200 rows x 4 columns]
0.89
```

Figure 23: Accuracy of the KNN model

This capstone project can be further improved to achieve even better results. If I had more time and resources available I would have experimented with different methods to get better results. First, I would extract more features from the images which would offer more data to the model to train on. Second, I would acquire more performant hardware and train the model

with thousands of images. Third, I would use other machine learning models and choose the algorithm with the best accuracy such as a Decision Forests. Fourth, I would expose my machine learning model through an Application Programming Interface (API) on the web so that anyone in the world can make use of it by calling the API directly. Finally, using Convolutional Neural Networks could also be a good path to take when it comes to implementing a similar project as it provides a different approach to solving the problem of detecting melanoma skin cancer during which the feature extraction is not done manually.

6 STEEPLE ANALYSIS

This section specifies the social, technological, economic, environmental, political, legal, and ethical impacts of this project. It is important to take into consideration these implications and impacts when developing such a project since it is directly related to its success.

6.1 SOCIAL IMPACT

This project has a direct social impact since it directly touches the health records of the patients. Moreover, this project has a great added value for the community since it has the potential to prevent people from experiencing skin cancer in the late stages as it can facilitate the early detection of this disease. In addition, this project can help health professionals and dermatologists save lives by decreasing their everyday workload and making them focus on the most critical cases.

6.2 TECHNOLOGICAL IMPACT

From a technological perspective, this project made use of other research papers and technology advancements. The implementation of this project included the use of already invented algorithms and techniques. Despite being developed many years ago, those algorithms are still relevant and get improved by other contributors. I used different articles and research papers to help me understand and implement this project. However, the innovation of this project was in the feature extraction phase since most of the projects I consulted online used Convolution Neural Network and didn't work manually on feature engineering. The challenge in this project was to understand the approach of dermatologists and try to replicate that approach using the right technology and algorithms.

6.3 ECONOMICAL IMPACT

This project has the potential to reduce the cost of the skin cancer diagnosis consultation since doctors will be able to have a fast response from the computer thus seeing more people per day which will yield a cheaper service. Moreover, using this project would enable doctors to not use skin biopsy often which will also decrease the price of the diagnosis. The decrease in the price serves the common good of the communities because more people can afford it. The

introduction of computers and image processing in the medical field brings a great added value since it makes the processes faster and more affordable.

6.4 ENVIRONMENTAL IMPACT

This project is environmentally friendly. The main reason is that this project uses a simple image of the lesion for diagnosis which means that patients can send the images from home to the doctor and wait for a response without using a car or public transportation to go physically to the doctor's office. Using this project in real life would help decrease the overall pollution coming from vehicles. However, computer machines still consume electricity but the process is less polluting than commuting to a hospital.

6.5 POLITICAL IMPACT

There is no political impact for this specific project.

6.6 LEGAL IMPACT

This project deals with the patients' health records and introduces the use of computers in the diagnosis of skin cancer. If this project were to be used in a certain country, that country should have a clear regulation that allows the use of such solutions. Moreover, the patient needs to agree and sign a document that describes the whole process of using image processing machine learning in the diagnosis phase. That document should serve also as proof of agreement to use this new way of diagnosing melanoma. Furthermore, the patient should be aware that there could be an error produced by the machine learning model since 100% accuracy couldn't be reached.

6.7 ETHICAL IMPACT

This project is considered to be ethical for two different reasons. First, the images are used anonymously. The software solution does not take into consideration the private information of the patients. Instead, it uses the raw data existing in the image. Second, the project aims to save lives which is a noble objective to achieve.

7 CONCLUSION

This capstone project was a good opportunity for me to get introduced to the field of image processing mainly and machine learning partially. I faced a lot of challenges and roadblocks during the journey of completing this project since my first encounter with image processing techniques and algorithms was during this project. I learned to manipulate images by loading them to memory, pre-process images to remove noise and adjust contrast, segment images to separate an object or shape from its background, post-process the images, extract features images, and train a machine learning model to classify the images into different classes. One of the main challenges faced was the feature extraction phase. The latter was hard to implement since I did not use any Python libraries that directly extract the features wanted, instead I extracted the features in a manual manner by traversing the image pixel by pixel. Moreover, the post-processing phase was a roadblock since the machine learning model was strongly dependent on it and the project couldn't move forward without it. In addition, I learned a lot about the technologies used in such projects as Python, OpenCV, *PIL* library, and Scikit Learn library.

8 REFERENCES

[1] "2O2O Melanoma Skin Cancer Report Stemming the global epidemic." [Online]. Available: https://melanomapatients.org.au/wp-content/uploads/2020/04/2020-campaign-report-GC-version-MPA_1.pdf.

[2] "Morocco." [Online]. Available: https://gco.iarc.fr/today/data/factsheets/populations/504-morocco-fact-sheets.pdf.

[3] "Tests For Melanoma Skin Cancer | Melanoma Diagnosis," *Cancer.org*, 2020. https://www.cancer.org/cancer/melanoma-skin-cancer/detection-diagnosis-staging/how-diagnosed.html (accessed Nov. 30, 2021).

[4] M. Dildar et al., "Skin Cancer Detection: A Review Using Deep Learning Techniques," International Journal of Environmental Research and Public Health, vol. 18, no. 10, p. 5479, May 2021, doi: 10.3390/ijerph18105479.

[5] S. Jain, V. Jagtap, and N. Pise, "Computer Aided Melanoma Skin Cancer Detection Using Image Processing," Procedia Computer Science, vol. 48, pp. 735–740, 2015, doi: 10.1016/j.procs.2015.04.209.

[6] A. Scarlat, "melanoma," Kaggle.com, 2019. https://www.kaggle.com/drscarlat/melanoma (accessed Dec. 02, 2021).

[7] Guardian staff reporter, "Computer learns to detect skin cancer more accurately than doctors," the Guardian, May 29, 2018. https://www.theguardian.com/society/2018/may/29/skin-cancer-computer-learns-to-detect-skin-cancer-more-accurately-than-a-doctor (accessed Dec. 03, 2021).

[8] "ImageHistogram | Scientific Volume Imaging," *Svi.nl*, 2021. https://svi.nl/ImageHistogram (accessed Dec. 03, 2021).

[9] Ravindu Senaratne, "CLAHE and Thresholding in Python - Towards Data Science," *Medium*, Jul. 03, 2020. https://towardsdatascience.com/clahe-and-thresholding-in-python-3bf690303e40 (accessed Dec. 03, 2021).

[10] “Spatial Filters - Gaussian Smoothing,” *Ed.ac.uk*, 2021. https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm (accessed Dec. 03, 2021).
[11] “OpenCV: Smoothing Images,” *Opencv.org*, 2021. https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html (accessed Dec. 03, 2021).
[12] “Image Filtering.” [Online]. Available: https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Image%20Filtering.pdf.
[13] A. Fatima, “What Is Clustering and Common Clustering Algorithms ?,” *Medium*, Jan. 23, 2021. https://medium.com/swlh/what-is-clustering-and-common-clustering-algorithms-94d2b289df06 (accessed Dec. 04, 2021).
[14] SATYAJIT MAITRA, “What Canny Edge Detection algorithm is all about? - SATYAJIT MAITRA - Medium,” Medium, Feb. 24, 2019. https://medium.com/@ssatyajitmaitra/what-canny-edge-detection-algorithm-is-all-about-103d94553d21 (accessed Dec. 04, 2021).
[15] Sagar Kumar, “A straightforward introduction to Image Thresholding using python,” Medium, Oct. 02, 2019. https://medium.com/spinor/a-straightforward-introduction-to-image-thresholding-using-python-f1c085f02d5e (accessed Dec. 04, 2021).
[16] A. Rosebrock, “OpenCV Thresholding (cv2.threshold) - PyImageSearch,” *PyImageSearch*, Apr. 28, 2021. https://www.pyimagesearch.com/2021/04/28/opencv-thresholding-cv2-threshold/ (accessed Dec. 04, 2021).
[17] “Skin Cancer | ABCDE Assessment for Melanoma | Beaumont Health,” *Beaumont.org*, 2021. https://www.beaumont.org/conditions/melanoma/abcde’s-of-melanoma (accessed Dec. 05, 2021).
[18] Amit Upadhyay, “Classification In Machine Learning - Analytics Vidhya - Medium,” Medium, Jul. 16, 2020. https://medium.com/analytics-vidhya/classification-in-machine-learning-ed30753d9461 (accessed Dec. 05, 2021).