POLITECNICO DI TORINO

SECOND REPORT ON THE ICT for HEALTH LABORATORY

ICT4SS

Information and Communications Technologies for Smart Societies

A.Y. 2018-2019

Professor:

Visintin Monica

Student:

Bellone Lorenzo

Summary

Introduction

Teledermatology is a branch of the telemedicine that deals with the exchange of medical information concerning skin diseases. Two different approaches are applied in teledermatology: *store and forward* and *live interaction*. The first one concerns sending digital images and medical details to a dermatologist's data storage. The doctor can analyse the data and make a diagnosis at a later stage. Instead, the *live interaction* is based on a real time application, where the specialist visits the patient through interactive technologies (e.g. live video conference).

The most common approach is the store and forward, since it does not require the use of expensive equipment and, in addition, the simultaneous presence of the dermatologist and the patient is not necessary.

Some studies revealed that, dealing with diagnostic accuracy, the traditional dermatology is better than the teledermatology, even if the patients are more likely to prefer the second one because of the sharp decrease of the waiting times and costs.

When the store and forward approach is used, some image processing techniques can be applied in order to help the dermatologist with the diagnosis, and hence to increase the diagnostic accuracy. In the second lab these techniques are implemented on a specific skin disease, the melanoma.

The identification of a melanoma is based on the analysis of the moles. The dermatologist has to characterize five features (ABCDE):

1. Asymmetry;
2. Borders;
3. Colour;
4. Dimension;
5. Evolution.

Once that these features are evaluated, the mole can be classified as normal or as melanoma. It is not easy to derive the features from a digital image, that is why the image processing techniques are necessary.

In this lab it is required to find an algorithm in order to characterize the borders of the mole. The other four features are supposed to be already defined by other researchers.

It was possible to work directly on fifty-four real images, each one representing one of the three types of moles: eleven low risk moles, sixteen medium risk moles and twenty-seven melanomas. An example for each category is shown in figure 1.

TYPES OF MOLES

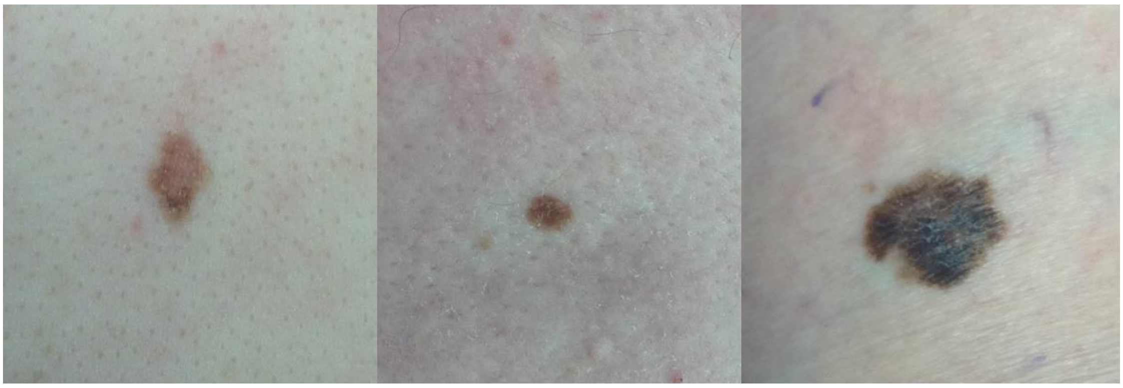

Figure 1 - From the left: an example of low risk mole, medium risk mole and melanoma

The Algorithm

The quantization of the image

This lab was performed with Python, using some libraries that will be explained later on. The jpeg images were imported through the library "*matplotlib.image*" [1], which turns the image in a three-dimensional array. The first two dimensions specify the image size in pixels, while the third dimension shows three numbers that represent the total amount of red, green and blue colour in each pixel.

The first task to perform is to distinguish the mole from the rest of the image. In order to do this, the image has to be quantized with a smaller number of colours (mainly three levels). The darkest level will match with the mole. The quantization was performed with the hard k-means algorithm [2], which is a clustering technique that matches each pixel with a cluster according to predefined features.

The hard k-means works in this way: once the number of clusters (the number of colours with which the image must be quantized) is decided, the algorithm chooses a random guess for each cluster, called centroid. Then, the distance between the colour of a pixel (identified with three numbers) and each centroid is evaluated. The pixel will match with the closest centroid. Once the first quantization is done, the centroids are updated by the evaluation of the barycentre of each cluster. The algorithm is repeated until the barycentre of every cluster does not change any longer.

"*Scikit-learn*" is the Python's library that allows to exploit this algorithm through the method "*KMeans*" [3].

Thanks to the application of this method, the identification of the mole is easier for the program. An example is shown in figure 2, where the k-means was applied to the three moles previously displayed in figure 1.

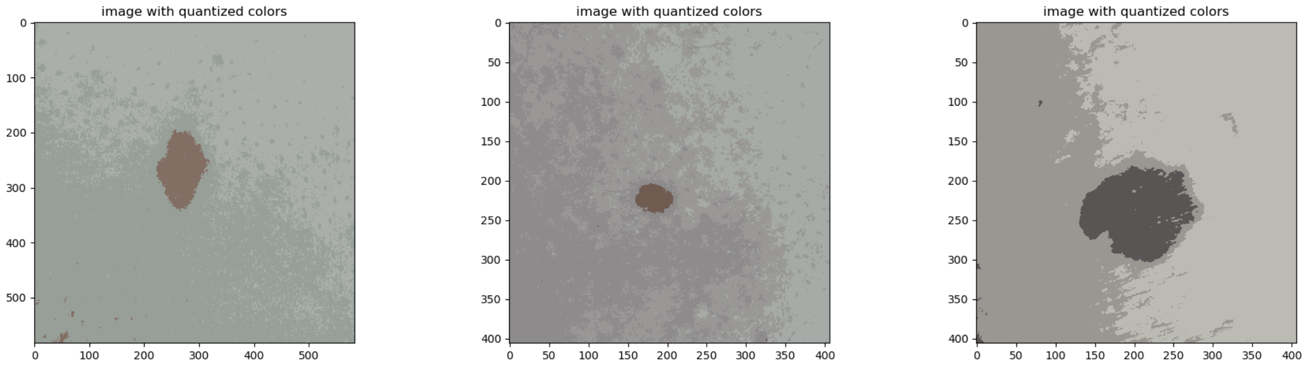

Figure 2 - The quantized images of the three moles shown in figure 1 obtained through the application of the hard k-means algorithm.

It might be the case that some pixels detached from the mole assume the value of the darkest colour. In this situation, the program would incorrectly take them as a part of the mole. Thus, each image must be cropped and centred on the mole. This procedure involved the k-means algorithm again, in order to classify the different darkest spots on the original image. The closest spot to the centre of the image corresponds to the mole while the other spots can be discarded. Then, a subset of the image including the entire mole is taken starting from the median point of the darkest pixels that have identified the mole itself. Some results are shown in figure 3.

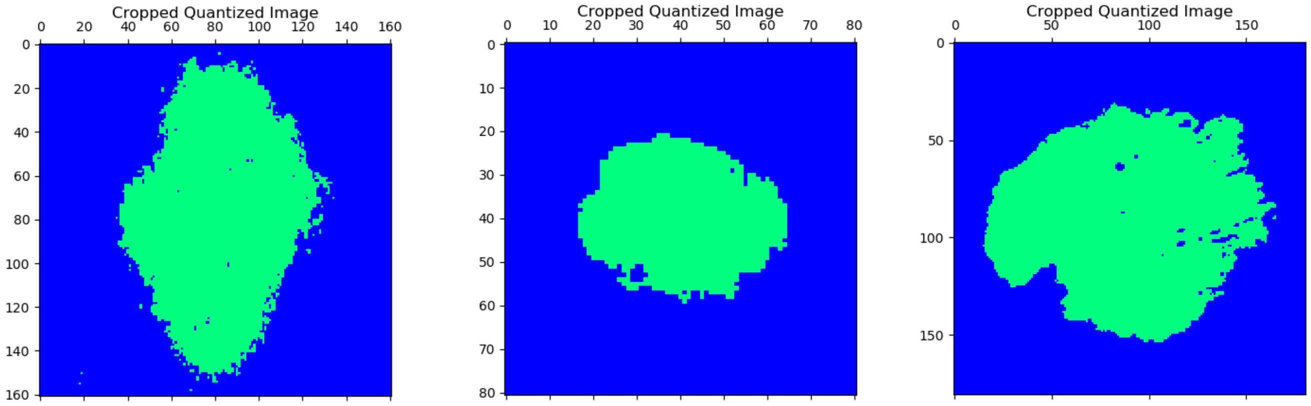

Figure 3 - The original images shown in figure 1 cropped and centred on the moles

The cleaning procedure

In the examples in figure 3, some imperfections are still present in the images. Therefore, there are holes inside the mole and spots situated out of the surface. These images are still not appropriate for the evaluation of the borders. The program would not consider the holes as parts of the mole but, on the contrary, the spots would be acknowledged as parts of it. These inaccuracies must be limited to guarantee a better performance of this measuring instrument.

The cleaning procedure plans to remove the defects using a submatrix of the original image. This submatrix is scrolled over the image: the holes, as well as the spots, are removed according to the pixels on the borders of the submatrix itself. Concerning the spots, if all these pixels are equal to zero, the matrix is out of the surface of the mole, and hence all its entries are set equal to zero; whereas the holes are filled if all the pixels are equal to one. In order to detect as many inaccuracies as possible, the matrix has not a fixed dimension, but its size grows up to a certain threshold, keeping its spread low and appropriate for the dimension of the mole. An example of this cleaning procedure is displayed in figure 4.

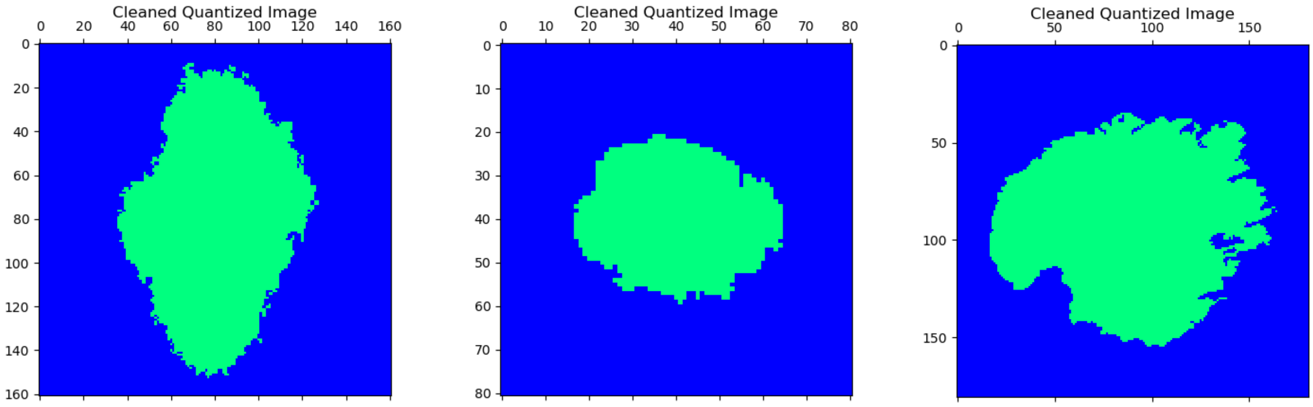

Figure 4 - The images shown in figure 3 with the application of the cleaning procedure.

The identification of the borders

The final phase concerns the identification of the borders of the mole. The program considers the difference between two adjacent pixels in every single row and in every single column. The possible results are minus one, zero and one, but the identification of the borders just needs zeros or ones, hence the absolute value of the difference must be evaluated. The complete image is given by the superimposition of all the outputs of the operations.

Another matrix of the same size of the cleaned image is built with the results of this procedure. An example is given in figure 5.

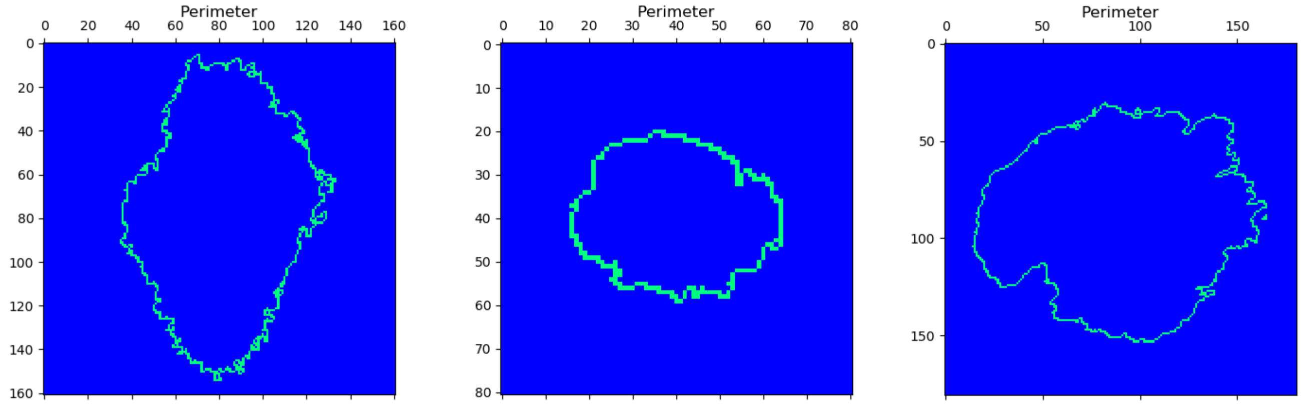

Figure 5 - An example of the final result of the algorithm applied on the moles displayed in figure 1.

The Results

Table 1 shows three results obtained for each analysed mole:

1. Its area, evaluated as the sum of the pixels equal to one in the cleaned quantized image.
2. Its perimeter, evaluated as the sum of the pixels equal to one in the perimeter image.
3. The ratio between the perimeter of the mole and the perimeter of a circle with the same area.

Name of the Mole	Area	Perimeter	Ratio
low_risk_1	8490	703	2.152
low_risk_2	10371	649	1.798
low_risk_3	5142	424	1.668
low_risk_4	5403	513	1.969
low_risk_5	4867	451	1.824
low_risk_6	6593	601	2.088
low_risk_7	3006	485	2.495
low_risk_8	7227	531	1.762
low_risk_9	3291	249	1.224
low_risk_10	1530	227	1.637
low_risk_11	3230	275	1.365
medium_risk_1	1320	179	1.390
medium_risk_2	8781	513	1.544
medium_risk_3	2294	230	1.355
medium_risk_4	2764	220	1.184
medium_risk_5	8441	849	2.607
medium_risk_6	7906	696	2.208
medium_risk_7	32139	1287	2.025
medium_risk_8	14008	768	1.830
medium_risk_9	4466	681	2.875
medium_risk_10	11805	1181	3.066
medium_risk_11	27791	1721	2.912
medium_risk_12	14951	620	1.430
medium_risk_13	6760	469	1.609
medium_risk_14	9786	594	1.694
medium_risk_15	8244	527	1.637
medium_risk_16	18596	1037	2.145
melanoma_1	12292	666	1.695
melanoma_2	7869	630	2.003
melanoma_3	10027	876	2.468
melanoma_4	12490	1185	2.991
melanoma_5	10143	832	2.330
melanoma_6	47326	2273	2.947
melanoma_7	38672	1859	2.667
melanoma_8	14626	1674	3.905
melanoma_9	26158	1517	2.646
melanoma_10	25961	1825	3.195
melanoma_11	15781	1227	2.755
melanoma_12	27300	1159	1.979
melanoma_13	16830	1176	2.557
melanoma_14	14144	825	1.957

melanoma_15	18673	1420	2.931
melanoma_16	11987	1142	2.942
melanoma_17	14066	1758	4.181
melanoma_18	41781	960	1.325
melanoma_19	22746	1289	2.411
melanoma_20	33962	1398	2.140
melanoma_21	11159	989	2.641
melanoma_22	16449	1129	2.483
melanoma_23	27925	2970	5.014
melanoma_24	30519	1775	2.866
melanoma_25	6715	457	1.573
melanoma_26	15171	998	2.286
melanoma_27	4523	1513	6.346

Table 1*. The area, perimeter and ratio between the perimeter of the mole and the perimeter of the circle with the same area for each mole.*

Conclusions

In conclusion, it is important to highlight that this program helps to define an estimation of the borders of a mole, which consists on one of the five features that are able to characterize a melanoma (ABCDE). Hence, the algorithm proposed does not aim to entirely define a melanoma but just one of its possible features. Furthermore, the values of perimeter and area are not extremely accurate since not all the holes are filled by the algorithm and some points of the borders are too jagged. The causes of these inaccuracies are the low quality of the initial image, as well as the simplicity of the algorithm since it requires a low computational power.

There are obviously many libraries available on Python that guarantee a higher quality in the results by exploiting the concepts of erosion and dilation of images [*4*], namely scipy.ndimage [5] and its methods. The problem with these solutions is the hard computation required.

References

[1] Matplotlib team developers. (2017, May 10). *Image tutorial — Matplotlib 2.0.2 documentation*. Retrieved December 16, 2018, from Matplotlib.org: https://matplotlib.org/users/image_tutorial.html

[2] Scikit-learn developers. (2011). *sklearn.cluster.KMeans — scikit-learn 0.20.1 documentation*. Retrieved from Scikit-learn.org: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[3] Wikipedia Contributors. (2018, December 14). *K-means clustering*. Retrieved December 16, 2018, from Wikipedia, The Free Encyclopedia.: https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=873678980

[4] Wikipedia Contributors. (2018, November 8). *Mathematical Morphology.* Retrieved December 27, 2018, from Wikipedia, The Free Encyclopedia.: https://en.wikipedia.org/w/index.php?title=Mathematical_morphology&oldid=867870926

[5] The SciPy community. (2018, December 17). *Multidimensional image processing (scipy.ndimage) — SciPy v1.2.0 Reference Guide.* Retrieved December 27, 2018, from Docs.scipy.org: https://docs.scipy.org/doc/scipy/reference/tutorial/ndimage.html