ICT for Health
Lab # 8 on moles

Monica Visintin

Politecnico di Torino

2016/17

Table of Contents

Laboratory #8 [1]

1. We work on images of moles: normal, with medium risk, tumor
2. The idea in to use a clustering algorithm to reduce the number of colors in the image to only 4 or 8, and then to analyze the new image with an algorithm that finds borders of objects in the image.
3. file `images.zip` stores the jpeg images of the moles
4. Each image is made of $N_1 \times N_2$ pixels; the color of each pixel is encoded using three integers from 0 to 255 (`uint8`), for the intensity of colors red, green, blue (RGB encoding). Code $[0, 0, 0]$ corresponds to black, $[255, 255, 255]$ to white.
5. You can import an image in Matlab using the line `A=imread(filename)`; you can view the image using the line `imshow(A)`; you can save the image in a file by writing `imwrite(A,filename)`.

Laboratory #8 [2]

6. When you import the image using the command `A=imread(filename)`, `A` is a 3D-matrix made of 3 $N_1 \times N_2$ matrices, the first one stores unsigned integers from 0 to 255 that encode red, the second green, the third blue.
7. It is convenient to reshape the 3D-matrix into a 2D-matrix:

```
[N1,N2,N3]=size(A);
N=N1*N2;% N is the total number of pixels
B=double(reshape(A,N,N3));
```

It is important to convert unsigned integers to doubles, because otherwise it is not possible to evaluate distances, which are instead required by the clustering algorithm. At this point, `B` is a matrix with N rows and 3 columns: the three columns can be thought of as features, and we can apply the clustering algorithms like the hard and soft K-means.

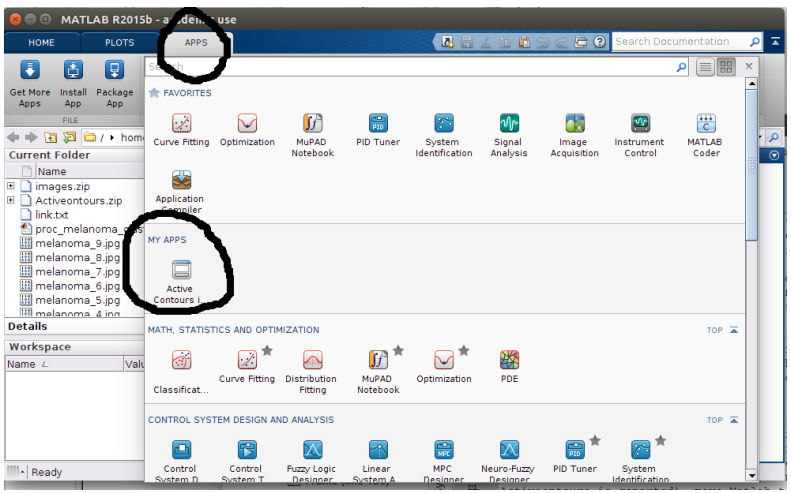
Laboratory #8 [3]

8. First implement the hard K-means algorithm (based on distances only, no evaluation of variances etc) using $k = 4$. The result of the algorithm is made of 4 representative vectors (i.e. 4 colors, made of weighted sums of red, blue, green) and an association of each row/pixel with one of the clusters. Then it is possible to build a new image `Bnew` by substituting each row of `B` with the representative vector of the cluster to which the pixel has been assigned to. In order to view the new image, it is necessary to use the following lines:

```
Bnew=floor(Bnew);
Anew=reshape(uint8(Bnew),N1,N2,N3);
```

Laboratory #8 [4]

9. Once you have the new image, store it in a jpeg file, which will then be processed by the algorithm which finds the borders of objects inside images. The Matlab files that implement the algorithm is in the zipped file `activeontours.zip`. Unzip the file (folder Activeontours is generated), move Matlab to folder Activeontours and (from Matlab) double click on file `Active Contours implementation & test platform GUI.mlappinstall`
   A new application will appear as My Apps in APPS



Laboratory #8 [5]

10. Click on the icon of the application Active contours and a window opens (adjust the size of the window if this helps you).
    1. load the new image file;
    2. select RGB (top right of the panel);
    3. just below the selection of RGB you can choose which color you want to work on: red (R) or green (G) or blue (B). The application actually shows a black and white image since it uses only one basic color and this is interpreted as level of gray
    4. on the bottom of the panel click on Define Mask: a new image opens and with the mouse draw a circle inside the mole, then double click inside the circle and select done. The new image closes
    5. Click on the green button Run and look at the image, which shows the working of the algorithm.
    6. If you are not satisfied with the results, you can change the number of steps of the algorithm (default value 40), right above the green button Run.

Laboratory #8 [6]

7. By default, the image is updated every 4 steps of the algorithm and it is saved every 100 steps. You can modify these values on the top of the panel.