ICT for Health
Laboratory # 4
Moles

Monica Visintin

Politecnico di Torino

ICT4SS
Information and Communications Technologies for Smart Societies

2017/18

Table of Contents

Chronic kidney disease [1]

Use the agglomerative clustering of SciPy on the data of Chronic kidney disease and comment the results

Table of Contents

1. Chronic kidney disease II part

2. **Laboratory # 4**
   - The data

The data [1]

- Download file `images.zip` from the folder materiale of the class
- Unzip the file, you'll get several jpeg images of moles, the name of the files are
  1. `low_risk_n.jpg` (where `n` is an integer) for moles that have a low probability of being melanoma (i.e. tumors)
  2. `medium_risk_n.jpg` (where `n` is an integer) for moles that have a low probability of being melanoma
  3. `melanoma_n.jpg` (where `n` is an integer) for moles that have a high probability of being melanoma
- Use a tool of your operating system to view all the pictures, so that you have an idea of what you have to work with (for example use Gwenview in Linux)

Goal of the lab

- We want to help medical doctors in the analysis of the moles
- 5 features are considered by the doctor to diagnose melanoma: ABCDE

  A asymmetry
  B border
  C color
  D diameter
  E evolution

- We want to analyze borders

Main idea

- We use K-means in scikit-learn to find three clusters (quantization of the image with three levels)
- We find the contour of the cluster corresponding to the mole
- We evaluate the area of the cluster corresponding to the mole and the length of the contour (perimeter of the mole)
- We evaluate the perimeter of a perfect circle with area equal to that of the mole
- We evaluate the ratio between the perimeter of the mole and the perimeter of the corresponding circle: the higher is this value the more indented is the border.

The jpeg image

- The jpeg image is an image that has been compressed according to the jpeg standard
- To read the image in Python:

  ```
   import matplotlib.image as mpimg
  filein=...
  im = mpimg.imread(filein)
  ```
- `im` is an Ndarray with shape 583 x 583 x 3 and elements of type unint8 (unsigned integer with 8 bits)
- `im[:,:,0]` is the amount of red color, from 0 to 255
- `im[:,:,1]` is the amount of green color, from 0 to 255
- `im[:,:,2]` is the amount of blue color, from 0 to 255
- Value `[0,0,0]` corresponds to black, value `[255,255,255]` corresponds to white
- To show the image in Python: `plt.figure()`

  ```
  plt.imshow(im)
  plt.title('original image')
  plt.show()
  ```

K-means

- Import K-means by writing:
  `from sklearn.cluster import KMeans`
- To instantiate the K-means object write:
  `kmeans = KMeans(n_clusters=3, random_state=0)`
- To find the clusters you should write:
  `kmeans.fit(im)`
  but Python gives you an error, because it requires a 2D Ndarray, not a 3D Ndarray
- What can we do?