6) To segment the image of Diodato I realized a python program. During the work I had many problems. A first problem is about the size of the graph. The image given in input is 800x458 pixels and in order to manage a graph containing 366400 nodes I found some problems due to memory. This is why I decided to resize the image to 100x57 using Photoshop. Another problem I found is in the manipulation of pixels of a colored image, so I loaded the image as black and white. I made these decisions hoping that it will work the same.

I construct a graph from the image. This graph is like a grid graph, each node correspond to a pixel of the image and an edge between two nodes exists if the pixels in the image are close. In the paper of Shi and Malik they said that an edge exists between two nodes if the spatial location between the correspondent pixels is less than r, in my case I consider single pixels and r equal to 2 (so distance between pixels equal to 1). If the edge between two nodes exists a weight is assigned to it, this weight is set to $w_{ij} = e^{\frac{-|pixel(i)-pixel(j)|^2}{\sigma_I^2}} * e^{\frac{-|x(i)-x(j)|^2}{\sigma_X^2}}$. pixel(i) represent the intensity of pixel i, x(i) represent the position of that pixel i, σ_I and σ_X are two constants set to $\sigma_I = 4$ and $\sigma_X = 6$. I decided to set these values after trying many others during runs seeing the result was quite good.

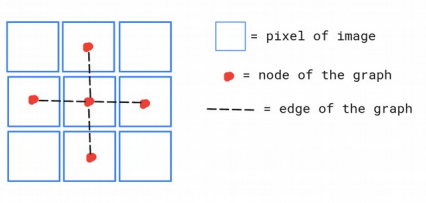

Illustration 1: Relation between pixels and nodes

Once I have created the graph, I compute the adjacency matrix A, the degree matrix D and the Laplacian matrix L=D-A. From the Laplacian matrix I found the eigenvalues and the eigenvectors. Ordering the eigenvalues I can easily found the smallest eigenvalues greater than 0 and their correspondent eigenvectors.
When I have an eigenvector I can compute a cut (S, V\S) on the graph. By scanning the eigenvector and I decided if the correspondent node is in the S or in V\S. As said in the paper we can choose as split value the mean of the components of the eigenvector or the value 0 and I chooses 0.
I cut the graph using four eigenvectors related to the first four smallest eigenvalue greatest than 0. For each cut I realize an image showing me the result of the cut.

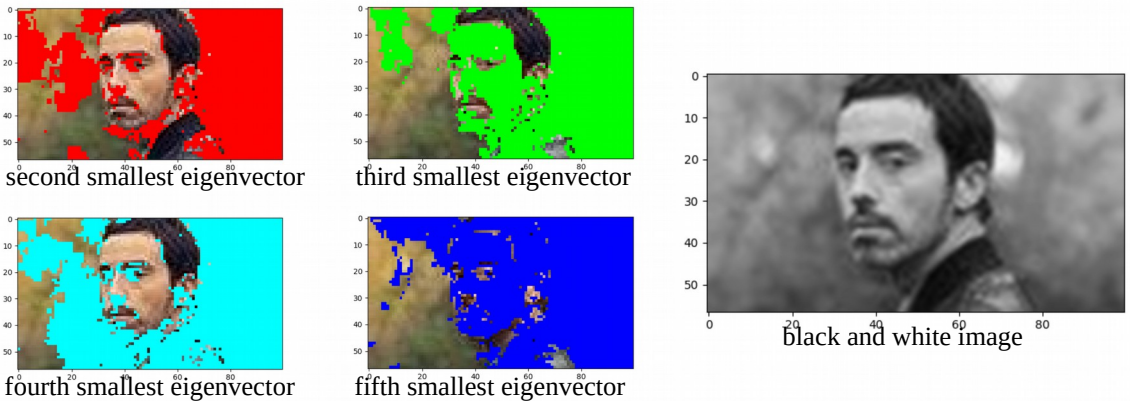

Illustration 2: Segmentation of Diodato.jpg

References: Normalized Cuts and Image Segmentation, Jianbo Shi and Jitendra Malik