Computer Vision - Report for lab 3

Task 1
It is a pretty simple task that recalls the previous laboratories. The program just loads the image and shows it in a window. The important functions used are *imread()* and *imshow()*, used respectively to read the image and to show it in a window.

Task 2
The program shows the image robocup.jpg and then it handles the mouse's event with a callback function. In fact, in addition to the *main()* it is also defined a function called onMouse that is called every time the user clicks on the image. This function takes five arguments and retrieves the BGR color value of the clicked pixel using a function called at<Vec3b>. The obtained triplet of values is then printed to the console using a cout.
To set up mouse event handling, the program calls the setMouseCallback function with a pointer to the onMouse function and a void pointer to the Mat object. It is also specified the name of the window where the mouse events should be handled.

Task 3
The *task3.cpp* file is very similar to the previous one. The only addition to the program is the computation of the mean for each channel of the neighborhood pixels. So, through a for cycle, we scan the 9x9 filter centered on the desired pixel and we compute the sum of each channel. At the end, we divide the sum for 81 (9x9) in order to obtain the mean value.

Task 4
Through this task, we create an algorithm that creates an image with the pixels of similar colors as the clicked one. So the program handles the mouse click event, it generates a new image called *Mask* where all the non-similar pixels, compared to the clicked one, are set to black (0,0,0 in BGR) and it shows the result. The similarity criterion checks that the absolute value of the difference between the inspected pixel's value and the mean pixel's value is less than the threshold value. This condition must be satisfied for each channel in order to not be deleted from the mask. I tested another criterion where I considered the three channel values as the component of a point in the three-dimensional space. The condition to be selected was based on the distance between the mean point and the inspected one, if it was less than the threshold it was considered, otherwise it was deleted. Unfortunately, the resulting image had almost the same results of the previous, so I opted for the simpler criterion, but I left the other one commented on the code. This method of selection is not very precise, in fact often it selects the ball in the lower part of the image, it depends a lot on the initial click of the user.

Task 5
This task is another addition to the previous algorithm. In fact, beyond the mask, we create a new image, cloned from the original one, but with the mask's pixels changed in color. The result is good, but not perfect. You can definitely see the t-shirts with a different color, but sometimes the selection is not so accurate. This is due to the fact that the pixel similarity criterion is not so precise and sometimes it ends up selecting just some part of the t-shirts or also a part of the ball.