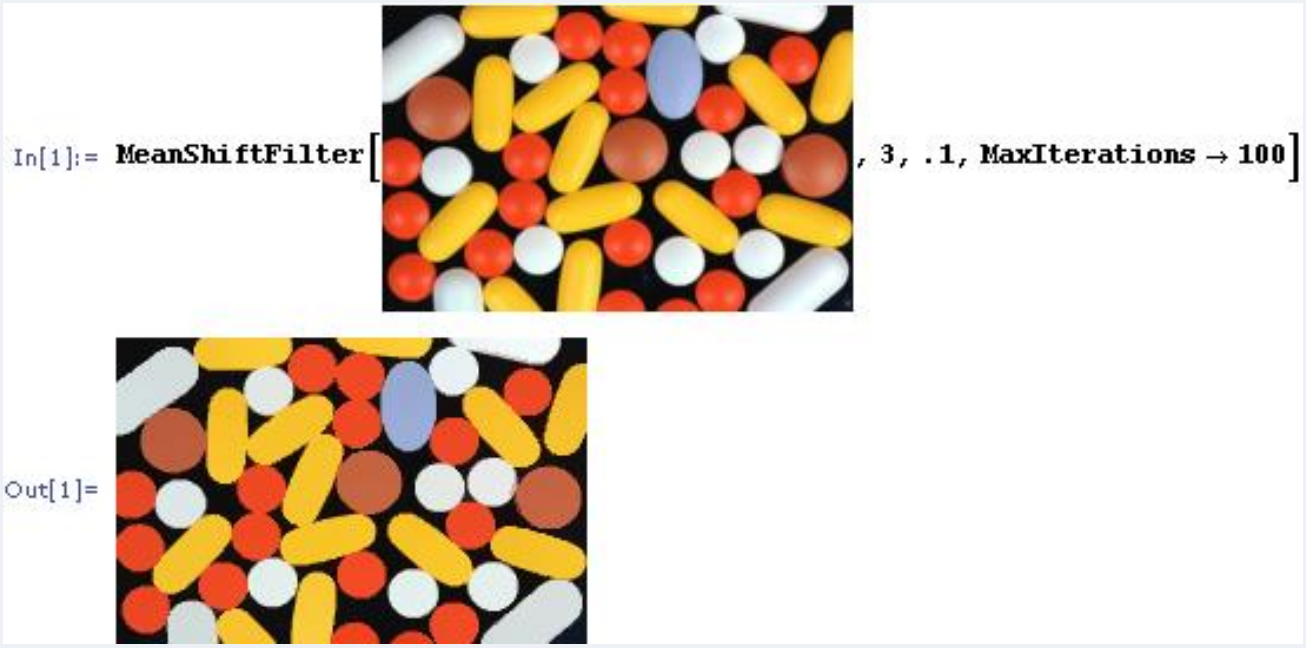
Mean Shift

Mahdi Akbari Zarkesh
9612762638

The basics :

The Mean Shift segmentation is a local homogenization technique that is very useful for damping shading or tonality differences in localized objects. An example is better than many words:

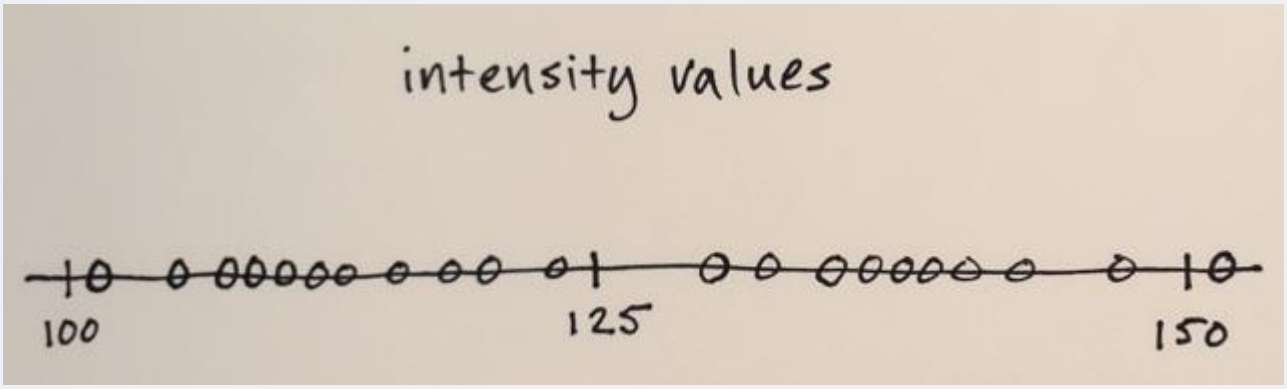


Action : replaces each pixel with the mean of the pixels in a range-r neighborhood and whose value is within a distance d.

The Mean Shift takes usually 3 inputs:

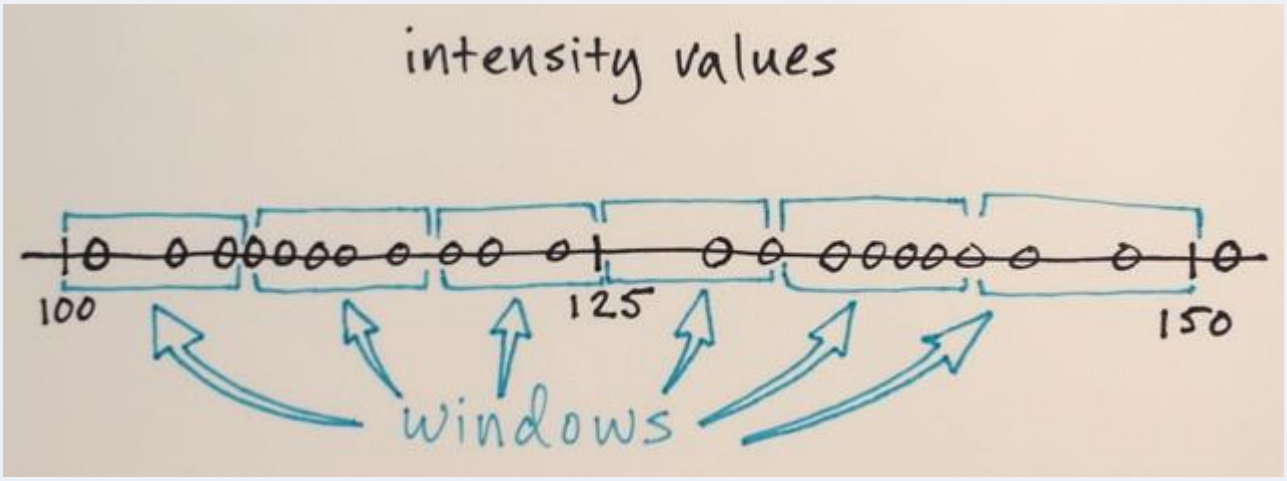
1. *A distance function for measuring distances between pixels. Usually the Euclidean distance, but any other well defined distance function could be used. The Manhattan Distance is another useful choice sometimes.*
2. *A radius. All pixels within this radius (measured according the above distance) will be accounted for the calculation.*
3. *A value difference. From all pixels inside radius r, we will take only those whose values are within this difference for calculating the mean*

A Mean-Shift segmentation works something like this:



In your case, all you have are intensity values, so feature space will only be one-dimensional. (You might compute some texture features, for instance, and then your feature space would be two dimensional – and you'd be segmenting based on intensity and texture)

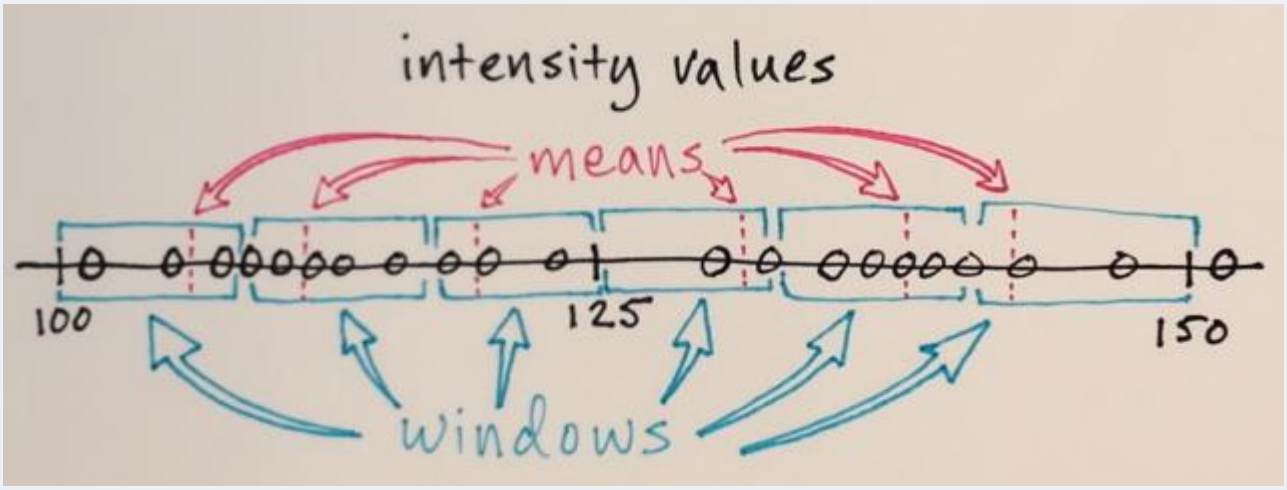
Search windows are distributed over the feature space



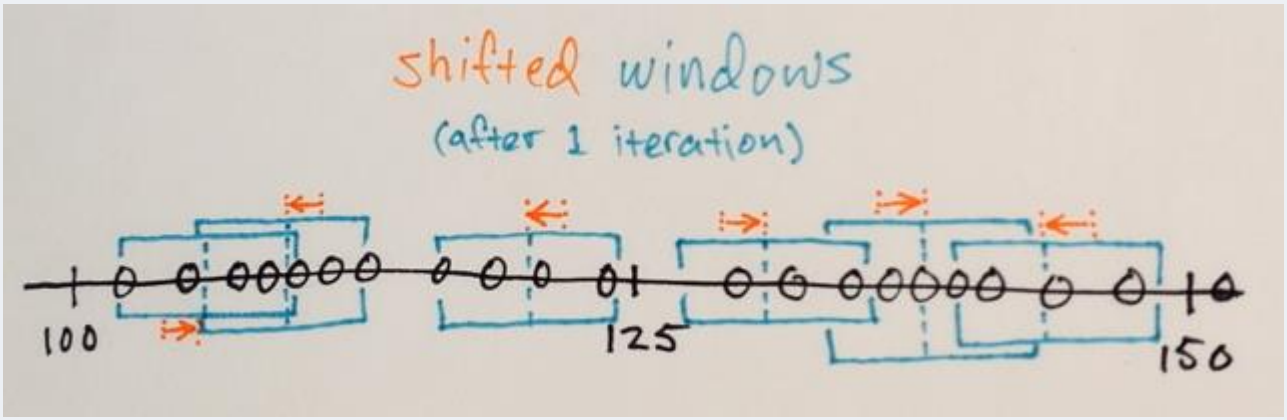
The number of windows, window size, and initial locations are arbitrary for this example – something that can be fine-tuned depending on specific applications

A Mean-Shift segmentation works something like this:

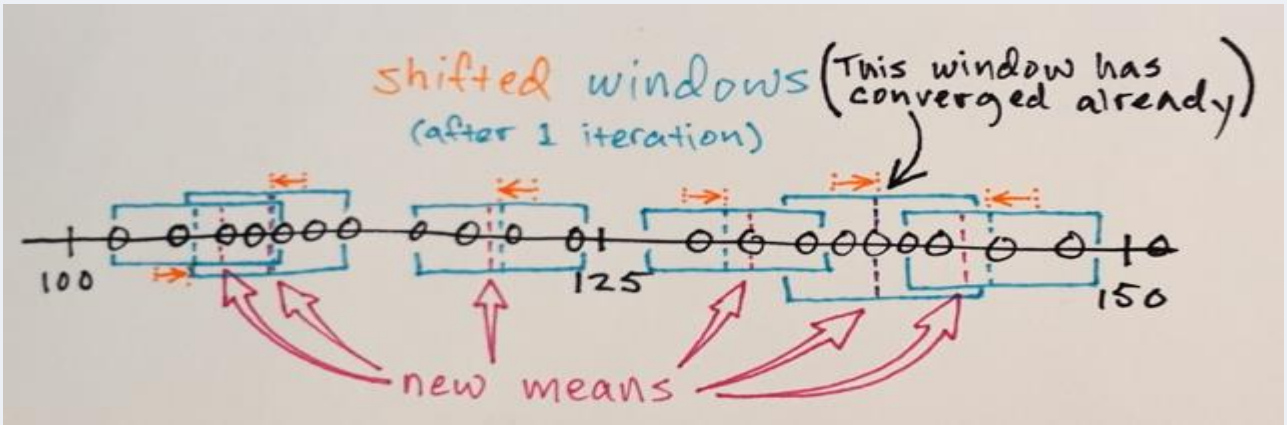
1-The MEANs of the data samples within each window are computed



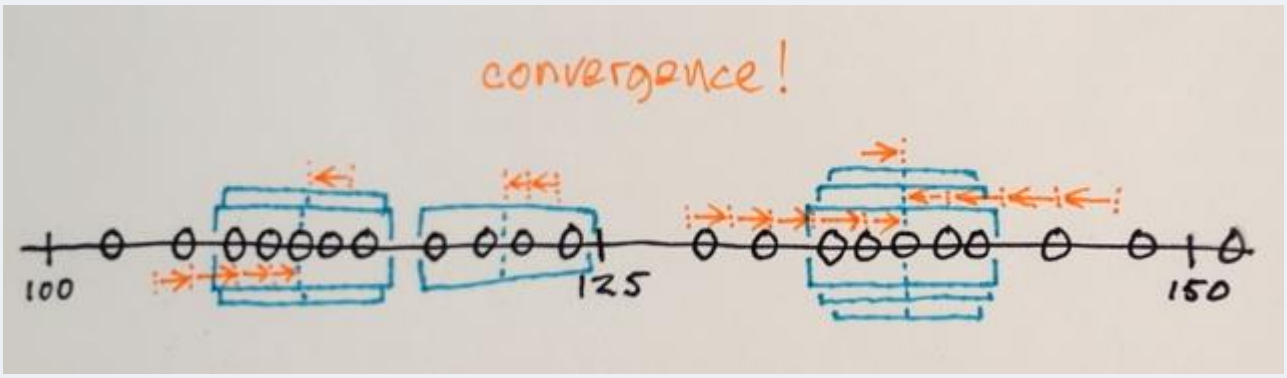
2-The windows are SHIFTed to the locations equal to their previously computed means



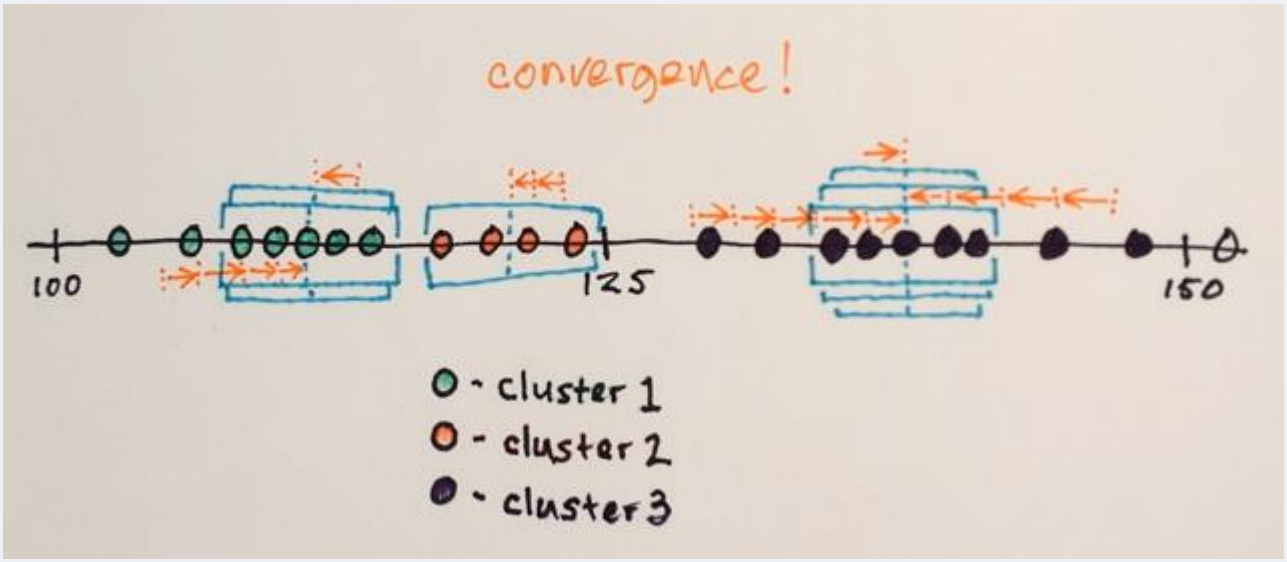
Steps 1.) and 2.) are repeated until convergence, i.e. all windows have settled on final locations



The windows that end up on the same locations are merged.



The data is clustered according to the window traversals



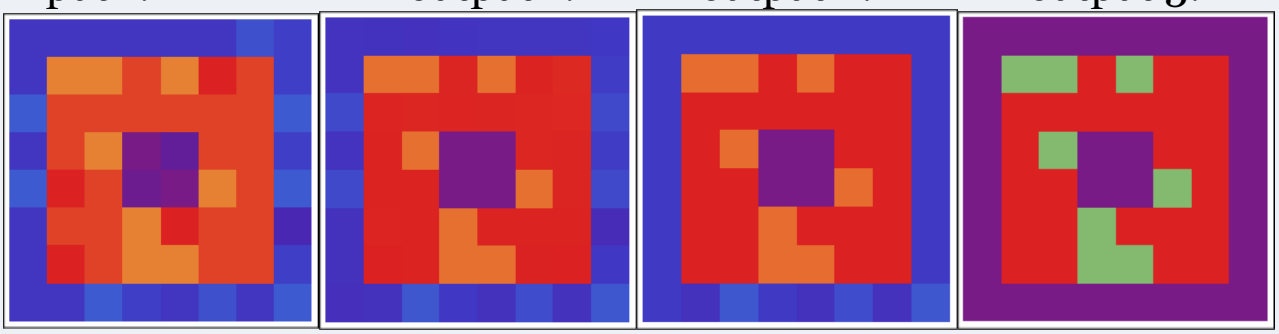
... e.g. all data that was traversed by windows that ended up at, say, location “2”, will form a cluster associated with that location.

So, this segmentation will (coincidentally) produce three groups.

Other sample:

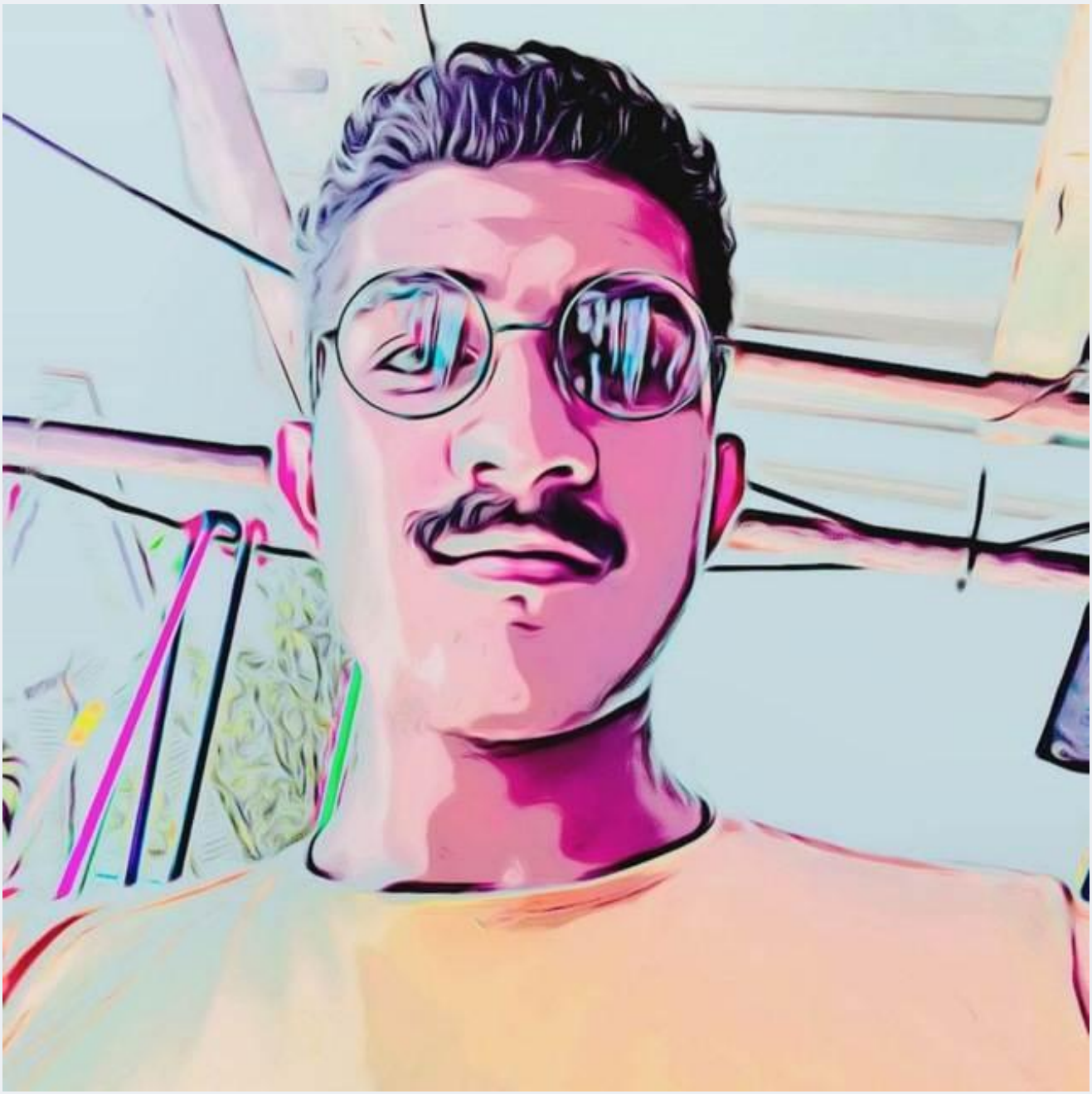
iterate several times with Mean Shift, trying to get a more homogeneous coloring.

Input 1: output 1: output 2: output 3:

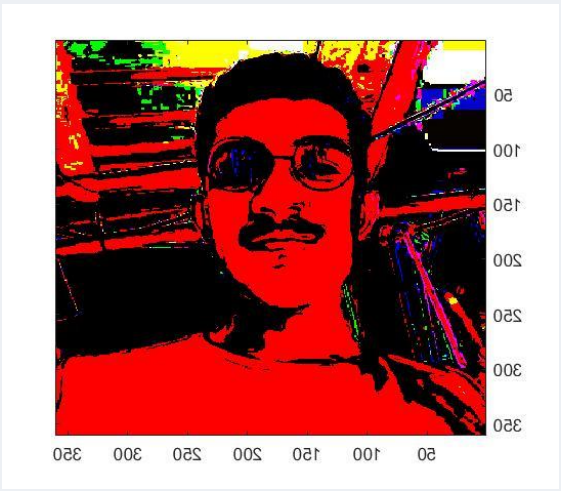


now look at the our result:

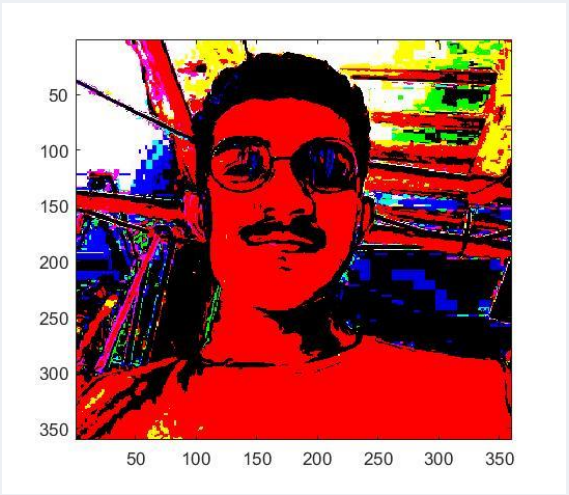
we use photo for data input:



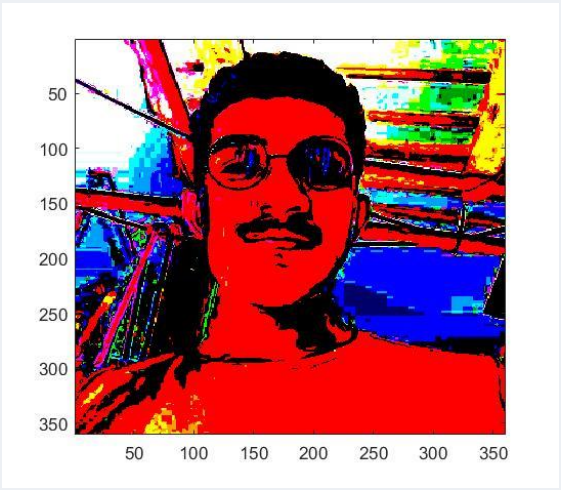
We have a loop that moves over the line of the picture, selects a pixel that has three components, the formula that is explained for a 15 iteration method, and finally we have the width of the image or the same row as our r, Then we select those r that are greater than a threshold.



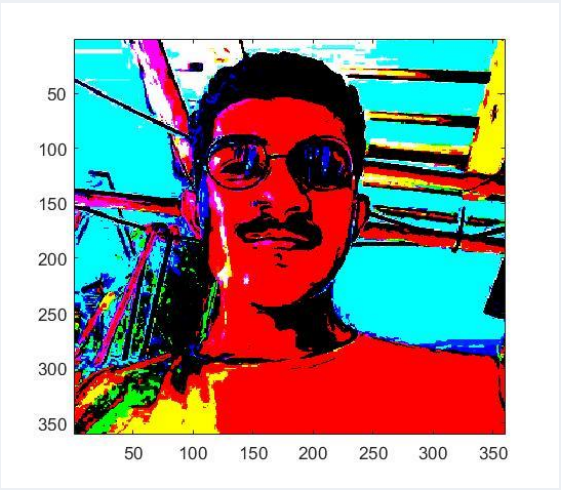
1.5 sigma



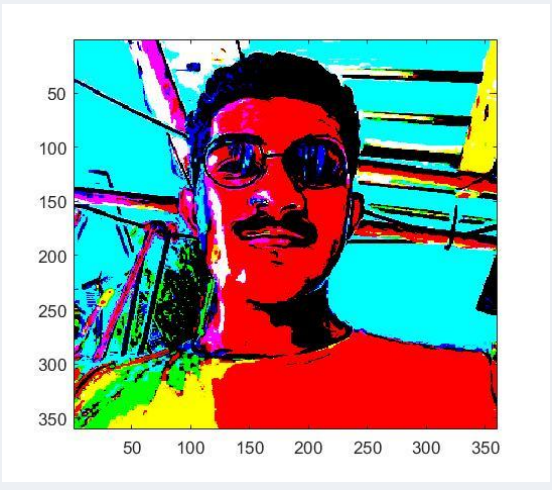
7 sigma



14 sigma



34 sigma



54 sigma

The lower the sigma value, the less cluster the program selects, and the greater the overall content.

Thank you :)