CS676A: COMPUTER VISION AND IMAGE PROCESSING

MEAN SHIFT IMAGE SEGMENTATION

PRANSHU GUPTA 13493, ABHISHEK JAIN 13027 [G13]

==

IMPLEMENTATION DETAILS

We load the image using the 'Image' module of python which gives us a matrix of RGB values. The shape of this matrix is same as the resolution of the image.

To begin the mean shift procedure, we pick seeds as centers of 40x40 grid on the image. Then we run the mean shift algorithm serially for each of these seeds. We iterate at most 15 times at each seed for it to converge, otherwise we move on to the next seed. (We can also pick a seed at each pixel but it costs a lot in speed)

For the mean shift procedure, we take each pixel as a five dimensional vector:

Here, x is the row index and y is the column index of the pixel in the image matrix.

$$[x, y, r, g, b]$$

Where x and y are the spatial co-ordinates of the pixel and r, g and b correspond to the color co-ordinates in RGB space.

In the distance measure we have:

$$m = 1 \qquad S = 5$$

$$D = \sqrt{{d_c}^2 + \left(\frac{d_s}{S}\right)^2 m^2}.$$

We have implemented both Flat and Gaussian Kernel for Mean Shift Algorithm.

After the mean shift procedure, we group together those seeds which have converged to points closer than the bandwidth. That is, we merge their attraction basins by replacing them with their mean vector.

Then we find out the nearest seed for each of the pixels in the image and assign new RGB values to get the segmented image.

NOTE: No python module other than ‘Image’, ‘sys’, ‘numpy’ and ‘time’ has been used in this implementation.

RUNNING THE APPLICATION

To run the mean shift image segmentation script, the user must issue the following command from the terminal:

```
$ python segment.py filename bandwidth Gaussian
```

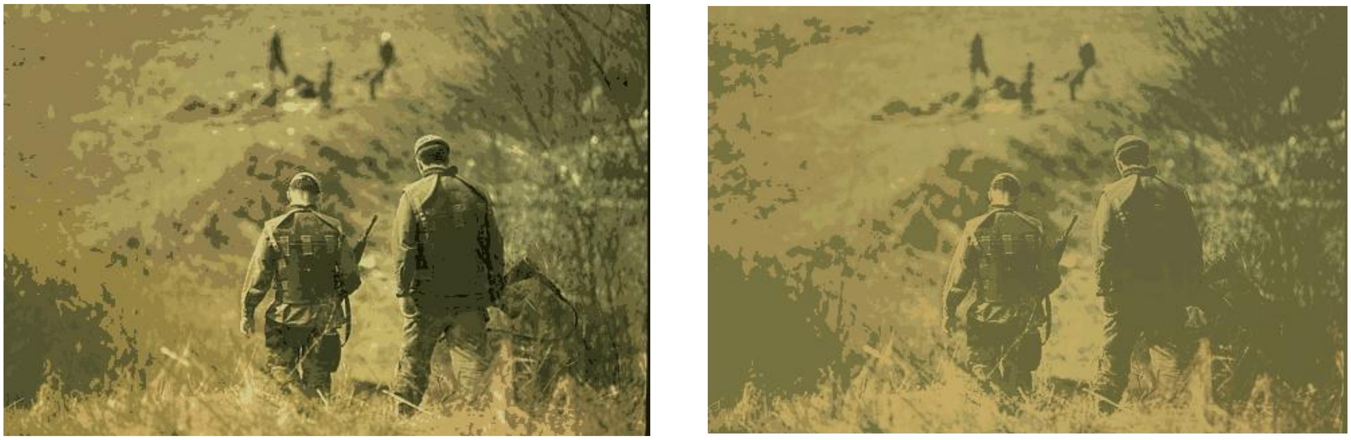
Here filename should be the name of the input image [.jpg] without the extension. Gaussian is 0/1 value telling whether to use a Gaussian kernel or not. If it is 0 then flat kernel will be used.

RESULTS

By plotting the number of converged means vs. Bandwidth for segmentation with flat kernel, we can see that the elbow of most of the curves is located near bandwidth = 40. So, we say that this value of bandwidth should be used to get a good segmentation.

Similarly, in plots for segmentation with the Gaussian kernel, we have the elbows located near bandwidth = 30. So, we say that 30 is the good value of bandwidth.

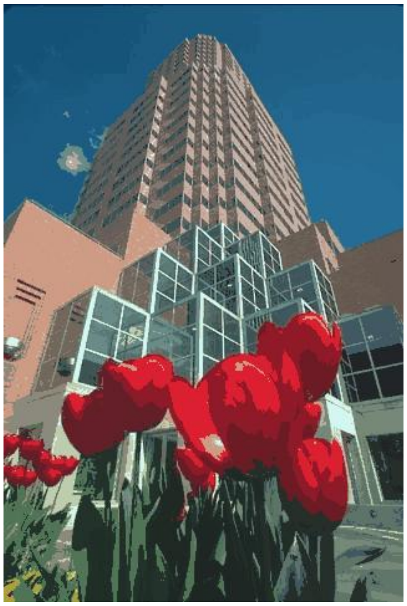
If we keep on increasing the bandwidth, we will lose useful information. This can be seen easily in segmentation result for the image 'input5.jpg', wherein the soldiers and trees all appear to be mixed up.



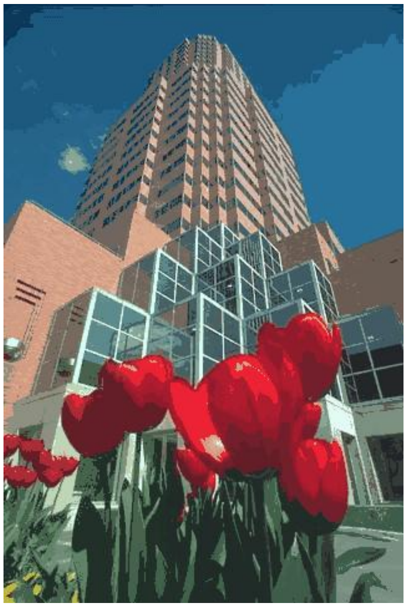
GOOD [bandwidth: 30]

BAD [bandwidth: 80]

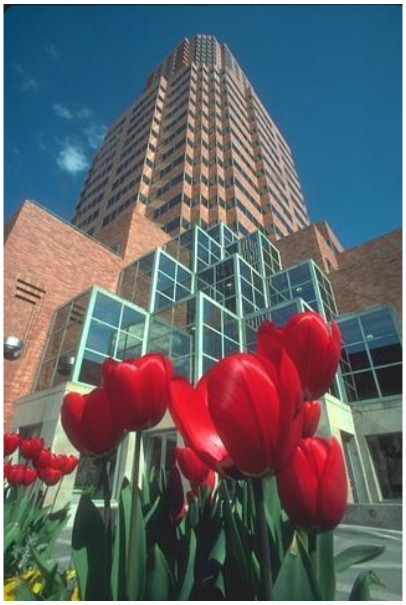
The above example have been generated with Gaussian kernel



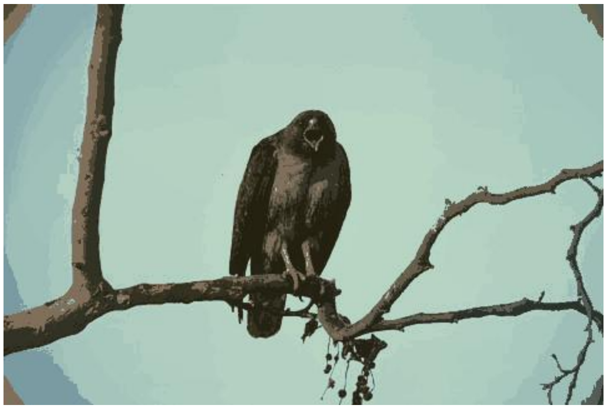
Flat Kernel, Bandwidth = 40



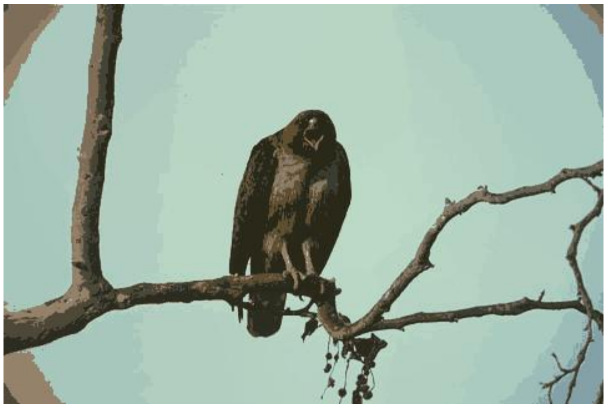
Gaussian Kernel, Bandwidth = 30



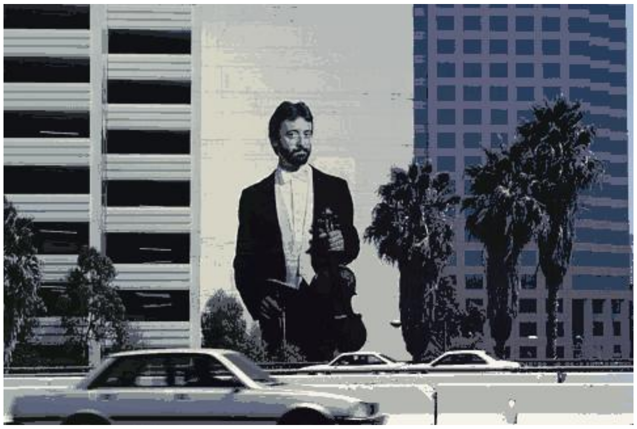
Input Image



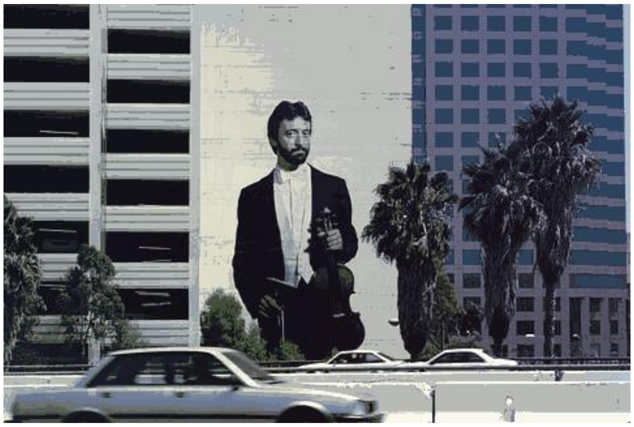
Flat Kernel, Bandwidth = 40



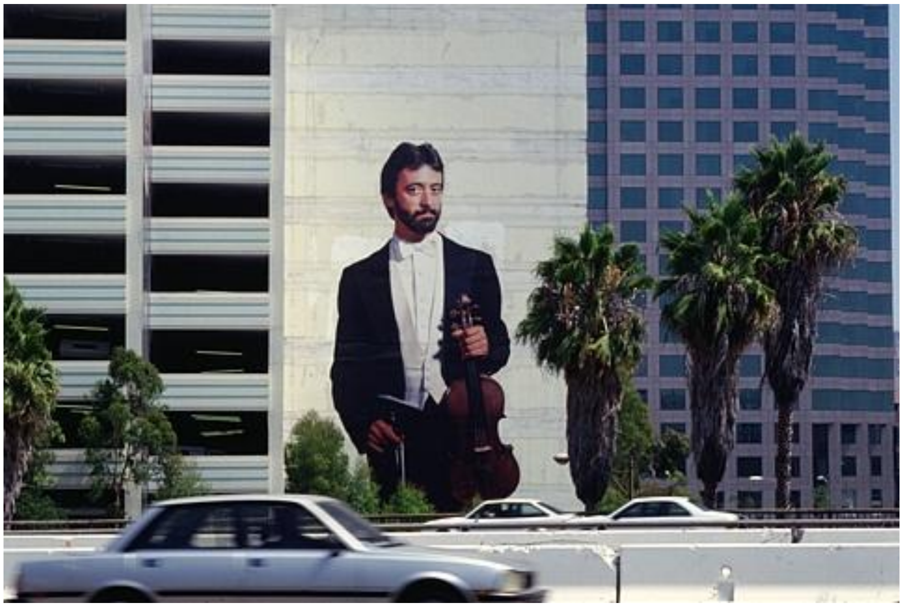
Gaussian Kernel, Bandwidth = 30



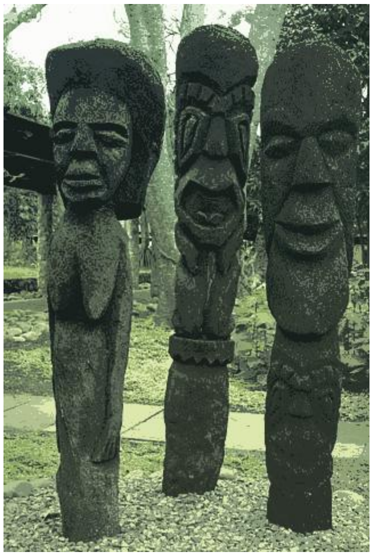
Flat Kernel, Bandwidth = 40



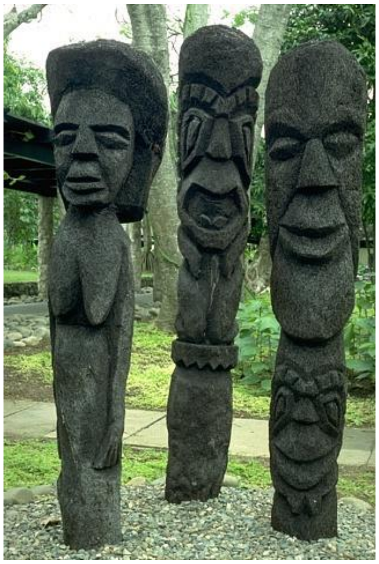
Gaussian Kernel, Bandwidth = 30



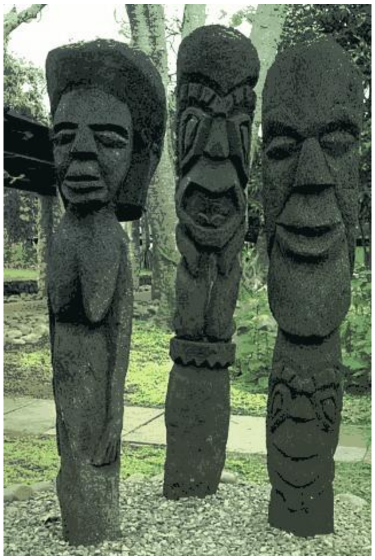
Input Image



Flat Kernel. b = 40

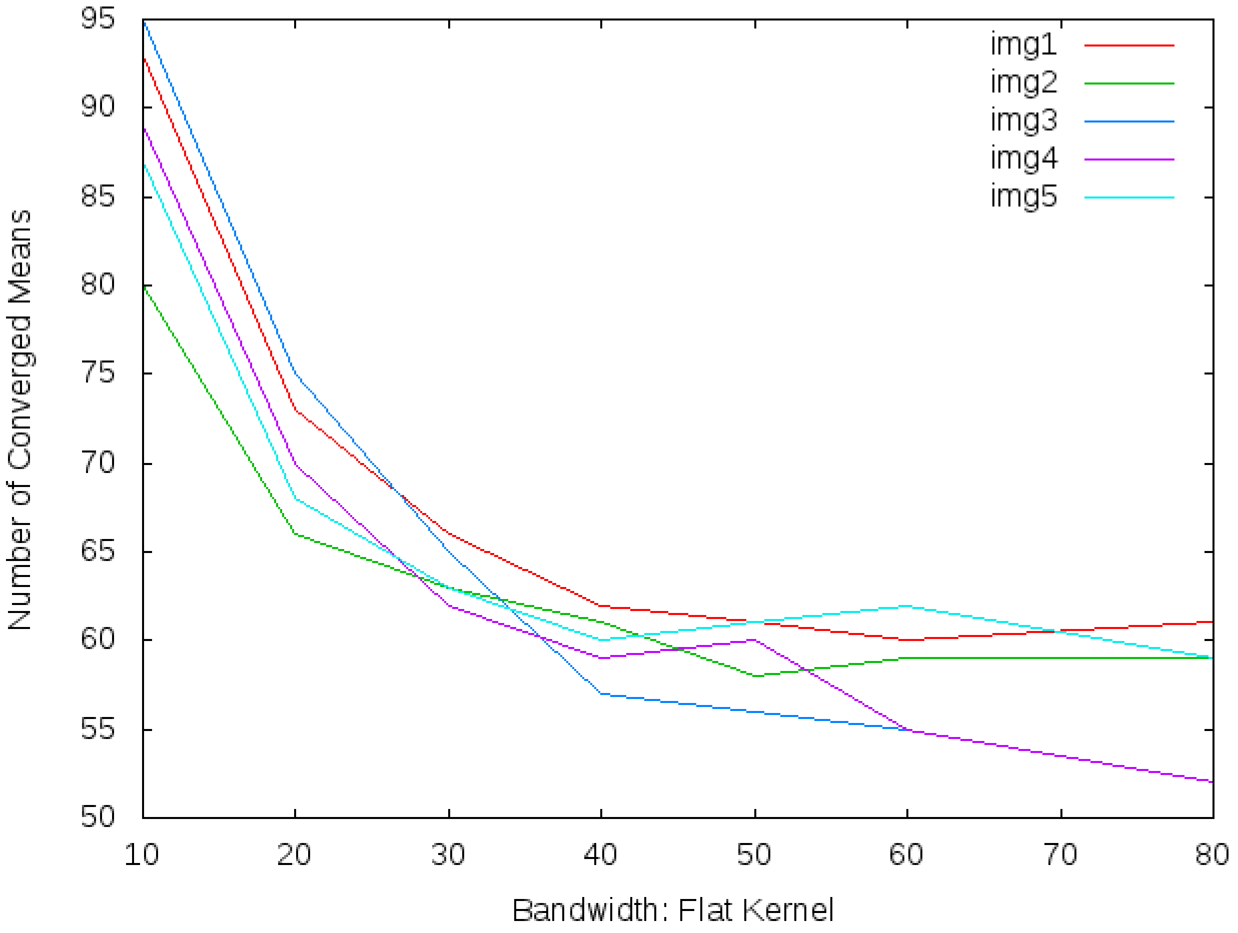


Input Image



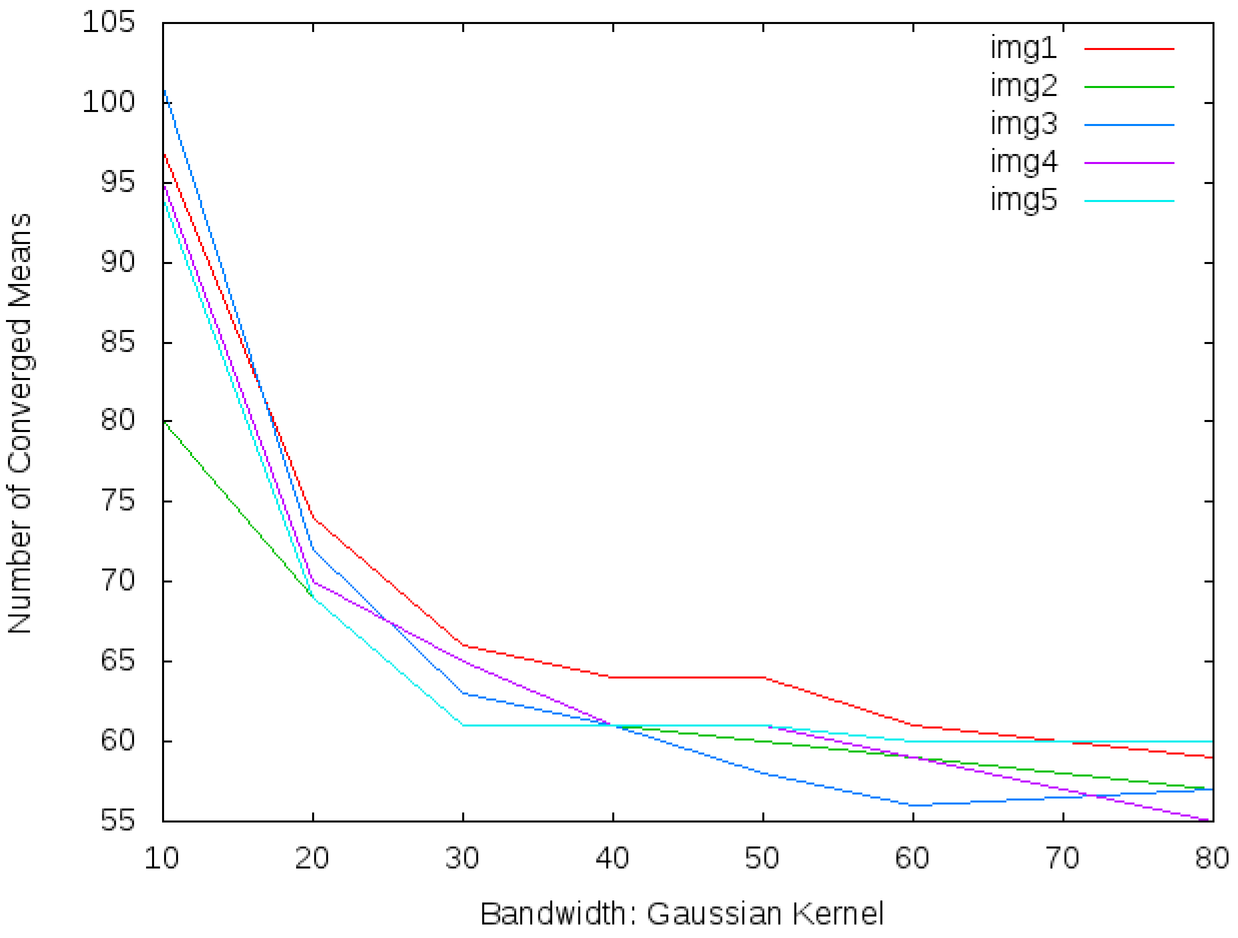
Gaussian Kernel, b = 30

PLOTS OF CONVERGED MEANS vs. BANDWIDTH: FLAT KERNEL



```
                    RESULTS (FLAT KERNEL) - #MEANS VERSUS BANDWIDTH
*************************************************************************************
        IMG/BW                 |  10  |  20  |  30  |  40  |  50  |  60  |  80  |
        ------------------------------------------------------------------------------
        input1                 |  93  |  73  |  66  |  62  |  61  |  60  |  61  |
        input2                 |  80  |  66  |  63  |  61  |  58  |  59  |  59  |
        input3                 |  95  |  75  |  65  |  57  |  56  |  55  |  52  |
        input4                 |  89  |  70  |  62  |  59  |  60  |  55  |  52  |
        input5                 |  87  |  68  |  63  |  60  |  61  |  62  |  59  |
```

PLOTS OF CONVERGED MEANS vs. BANDWIDTH: GAUSSIAN KERNEL



RESULTS (GAUSSIAN KERNEL) - #MEANS VERSUS BANDWIDTH

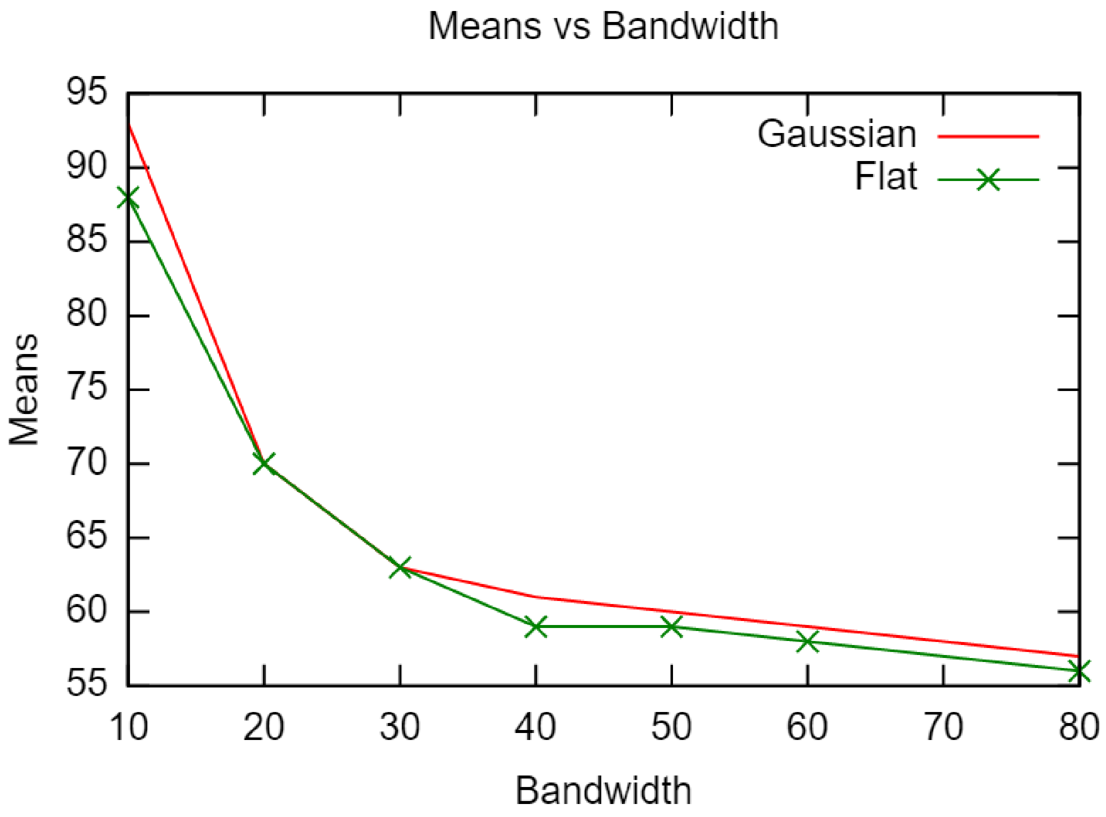
****

IMG/BW	10	20	30	40	50	60	80
input1	97	74	66	64	64	61	59
input2	80	69	61	61	60	59	57
input3	101	72	63	61	58	56	57
input4	95	70	65	61	61	59	55
input5	94	69	61	61	61	60	60

CONCLUSION

If we use very small bandwidth, then will have a peak for each point. This will result in each point being placed into its own cluster. On the other hand, if we use a large bandwidth. We will have one peak that all of the points will climb up to, resulting in only one cluster. So, in order to get good clustering we need to choose somewhere in between.

PLOT OF AVG. NUMBER OF MEANS VS BANDWIDTH



This plot clearly shows the elbows of number of means curve for both kernels. We see that Gaussian kernel (red) reaches its elbow at lower bandwidth (30) than the flat kernel (40). This is because the Gaussian function decays the weight of the points exponentially in their distance from the seed. So, the points that are far away do not add much to the result. However, in flat kernel every point inside the bandwidth has an equal contribution to make. Thus, is it takes a larger bandwidth for better convergence as compared to the Gaussian kernel.

Also, Gaussian kernel is better in retaining features from the input as we can see from the above plot. The Gaussian kernel gives a few more features at the same bandwidth as compared to the flat kernel.