National Institute of Technology Karnataka, Surathkal

Implementation and Comparative Analysis of Image Segmentation by Non-parametric Color Clustering

Submitted by:

Aditya S Gourishetty (171EE103)

Anusha Misra (171EE107)

Shivam Mahesh Potdar (171EE239)

Supervisor: Dr. Krishnan C M C

A report submitted in fulfilment of Lab Course Project
for the course EE86 (Digital Signal Processing Laboratory)

in the

Department of Electrical and Electronics Engineering

November 3, 2019

Contents

1 Abstract

Image segmentation is an important processing technique required in areas of computer vision as it is precursor to important applications like object tracking. K-Means Clustering is commonly used for image segmentation but it has some disadvantages, which are attempted to be addressed by algorithms like Mean Shift. The present work tries to compare the algorithms and studies the need and scope for improvement in Mean Shift clustering algorithm.

2 Introduction and Literature Survey

Image segmentation is an image processing process widely used in computer vision applications. Image segmentation in simple terms is a process of dividing an image into various parts based on different parameters for different applications which are done to give more meaning to an image for better analysis. Here's an example of Image Segmentation done based on colour. This simplicity makes computation and decision-making easy and hence helps in applying advanced techniques like object detection etc. which have application in autonomous vehicles, humanoid robots, etc.

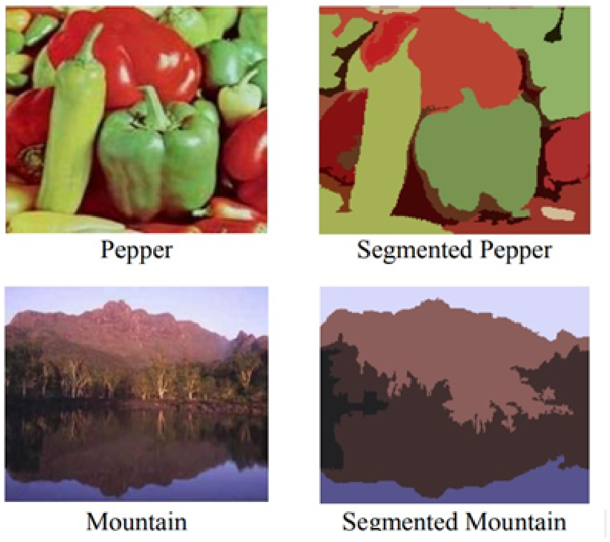

Figure 1: Image Segmentation

With the rise in automation in healthcare industries, Image Segmentation is gaining a lot of importance in various applications such as detecting an organ in a CT image, or a lesion in an MR image. Also, the various computer vision applications like face recognition and self-controlled vehicles all involve image segmentation as a preliminary process. Image Segmentation is done by converting an image into a collection of regions of pixels using various algorithms using techniques like region growing, clustering, and thresholding. Clustering algorithms are unsupervised machine algorithms but are similar to classification algorithms. Some of the commonly used clustering methods are K-means, ISODATA, and Fuzzy C.

K-Mean and Mean shift clustering algorithms work by having different means or centroids in a data set(may or may not be part of the data set) and assigning each data point to the closest mean. Then shift the mean iteratively to a point such that the sum of the distance from all the assigned points is minimum which is also the point with the maximum density of the assigned points(mode). After that reassignment of each of the data points to the updated closest mean takes place and after multiple iterations when no more reassignment takes place, the segmentation is complete and each mean is the mode of its respective cluster or segment.

In K-mean clustering, the number of means has to be manually defined whereas in mean shift clustering the number of means is derived based on the data.

3 Methodology

3.1 KNN

KNN or K Nearest Neighbour Algorithm is one of the very basic Machine learning algorithms used for classification problems. To understand the algorithm, one can consider all the points of a pre classified data set represented in a feature space along the query (the data to be classified) as shown in Figure 3 where all the points labelled 'a' and 'o' are from the data set and 'c' is the query. To classify the point 'c', k nearest (in terms of Euclidean distance) points of the point 'c' are taken into consideration and the class with the maximum points is assigned to the query. The value of k is application dependent. In Figure 3, k=3 where out of its 3 nearest neighbours, 2 of them are classified as 'o' so the point 'c' is classified to belong to the class 'o'.

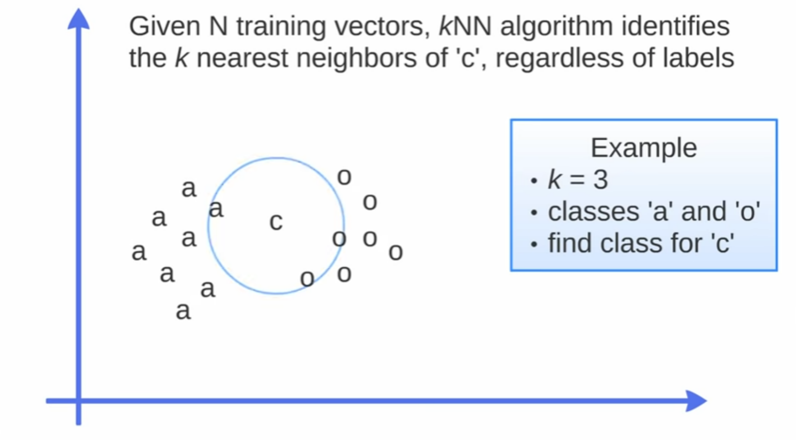
Given N training vectors, kNN algorithm identifies the k nearest neighbors of 'c', regardless of labels
Example
• k = 3
• classes 'a' and 'o'
• find class for 'c'
a
c
o

Figure 2: KNN algorithm.[]

3.2 Mean-Shift algorithm

The mean shift algorithm is very similar to K-Means, where different points in the data set are assigned to different centroids or modes, the mean is minimized but shifting of the modes and reassigning points based on minimum Euclidean distance. But in case of mean shift the points are not assigned to different predefined modes, but a region of interest (ROC) is taken in the distribution of data points in the feature space, and all the data points in the ROC are used to calculate the mean. The centre of the ROC is taken as the mode or centroid which is shifted to the point in the feature space where the mean of all the data points in the ROC is minimum. As the centroid shifts, ROC shifts simultaneously enabling new data points to enter and some of the older points to exit the data the ROC. This process is iterated until there is minimal shift of ROC and the final centroid of ROC is a mode of the data corresponding to a cluster. The assignment of the data points to the different clusters is done by initiating the ROC at each data point and assigning it to the cluster whose mode, the initialized ROC's centroid finally settles at.

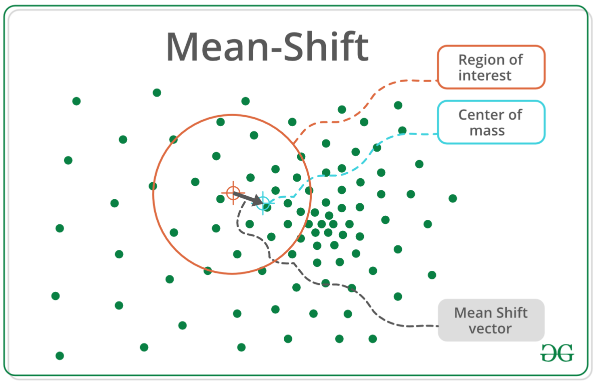

Figure 3: Mean Shift.[8]

3.3 KNN vs Mean-Shift

KNN although very popular has a big disadvantage that the number of clusters has to be pre-specified which is not a idle behavior in applications where the input images might have huge variation in color and other factors. K-means is very sensitive to initializations. A wrong initialization can delay convergence or some times even result in wrong clusters, whereas Mean-shift is comparatively less sensitive.

Mean-shift overcomes this advantage by having only one tunable parameter which is the kernel bandwidth (h), which can be thought of as a measure of how close colors need to be in the RGB color space to be considered related to each other.
Although compared to KNN, one of the disadvantages of Mean-Shift is the execution time is very slow due to N squared complexity. Although it can be optimized with parallelization as theoretically all pixels can be shifted independtly.

3.4 Kernels

The ROC is defined in Mean Shift using Kernels which are functions similar to window functions which are used in Kernel Density Estimation also known as Parzen–Rosenblatt window method. The idea behind using these windows is to give less weights or give less influence to the data points farther from the centroid while calculating mean so that the shift is gradual which can avoid errors like overshooting.
There are various types of Kernel functions such as Epanechnikov, Gaussian, rectangular, triangular etc used in Mean Shift. The weight distribution of a kernel is defined by its bandwidth. An increment in its bandwidth causes it's weight distribution to be more spread out giving more weightage to the points farther from the centroid while calculation of mean. A very good animated illustration of this is shown in [7].

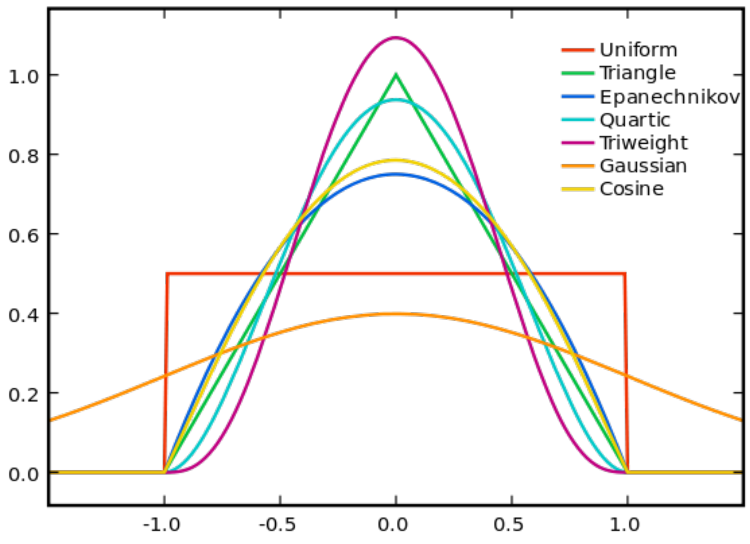

Figure 4: Various Kernels.[1]

3.5 Effect of Bandwidth on Mean Shift

As mentioned previously, an increment in a kernel's bandwidth causes its weight distribution to be more spread out giving more weightage to the points farther from the centroid which is similar to reducing the radius of ROC in Figure1. When the bandwidth is large, the smaller clusters are not identified because, two very close clusters can be included in the ROC simultaneously while calculating the mean and classified into a single cluster. But if the ROC is too small, we might sometimes obtain undesirable results where very small blobs in the feature space being classified as a cluster. Usually the higher the bandwidth use, the lower number of clusters are identified and visa- versa.The results of our code reflect the same and relevant observations have been expressed.

4 Experimentation and Results

With this project we intend to implement an improvised version of the mean-shift clustering algorithm i.e. 'Adaptive Mean-Shift method' mentioned in [10], in which local scale information is involved for calculation of bandwidth.
The entire project is split in two milestones,
Milestone 1:Implementation of the classic mean shift clustering
Milestone 2: Implementation of KNN to make it adaptive
The first milestone has been reached, with the code being successfully implemented, with satisfactory results. The second milestone is yet to be reached.

4.1 Milestone 1: Implementation of Classic Mean Shift

For implementing Mean shift we have used scikit learn python library which essentially used for various machine learning applications, one of which is clustering.

4.2 Main Code Snippets

```
bandwidth = estimate_bandwidth(X, quantile=0.1, n_samples=100)
```

We use sklearn.cluster.estimate_bandwidth for an oppropraite estimation of the bandwidth for the kernel based on the data set such that we get the fairly good results. The larger the value of the bandwidth, the farther the points that fall within the kernel. This is the parameter that changes when KNN is applied.

```
ms = MeanShift(bandwidth=bandwidth, bin_seeding=True)
ms.fit(X)
```

We use sklearn.cluster.MeanShift, which implements mean shift using a flat kernel.The bandwidth parameter is set to obtained value of the bandwidth using estimate_bandwidth. It also optimizes the algorithm by initializing the kernel to discretized version of the of the data points rather than all the data points.'ms.fit' performs clustering.

```
labels = ms.labels_
cluster_centers = ms.cluster_centers_
labels_unique = np.unique(labels)
n_clusters_ = len(labels_unique)
```

The above code is used to find the number of clusters that the algorithm finally ends at. np.unique further helps in choosing clusters that are unique, hence, avoiding redundancy.

```
segmented_image = np.reshape(labels, original_shape[:2])
```

The above code is used to redefine the segmented image using the labels that have been set earlier in the code. This is then plotted to get the final image.

4.3 Initial Results

The results can be seen in the following images. The algorithm works successfully, giving a neat segmentation of the image, based on the colour of the pixels.

Sample 1

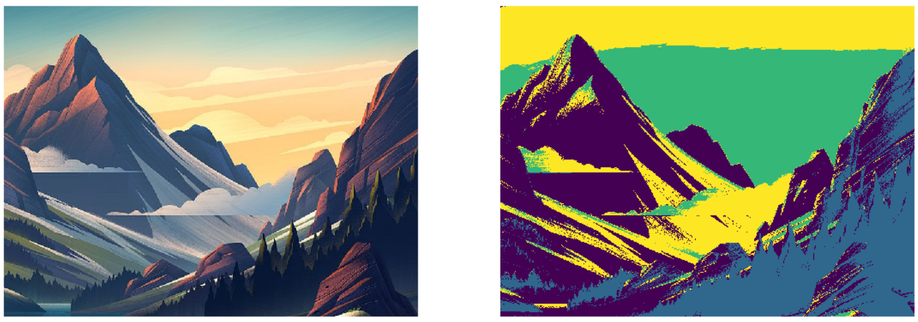

Figure 5: a) Original b) Segmented Image

In Fig.2.1 the Original Image (left) has been segmented into 4 color clusters, producing a color segmented image(right)

Sample 2

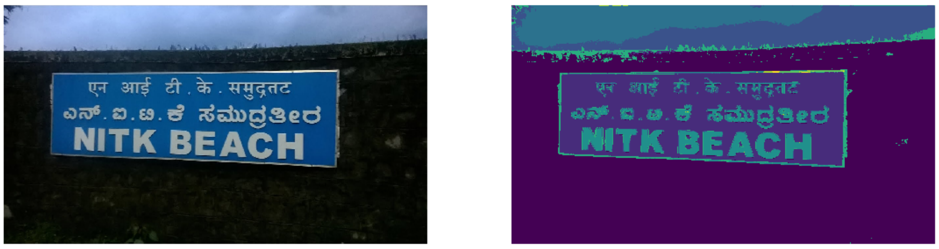

Figure 6: a) Original b) Segmented Image

In Fig.2.2 the Original Image (left) has been segmented into 9 color clusters, producing a color segmented image(right).Here the 9 different clusters are not very clearly visible but once zoomed in,we can see some pixels with shades of yellow and green too along with the clearly visible shades of blue indicating different clusters.

5 Discussions

As seen in the images in the previous subsection, the image segmentation has taken place and is distinctly visible. The segmentation is fairly clear and a clear boundary can be seen between different parts of the image, due to the different colours. The algorithm works efficiently as of now, which can be further improved by incorporating KNNs. This will be visible after Milestone 2 has been achieved.

Effect of bandwidth:

Let us look at the Segmentation of the image in Figure 5 with different bandwidths as shown in Figure 6,7 and 8.The results are in accordance to the theory explained previously where bandwidth of 80 gives the best result with the different parts of the image segmented as preferred.

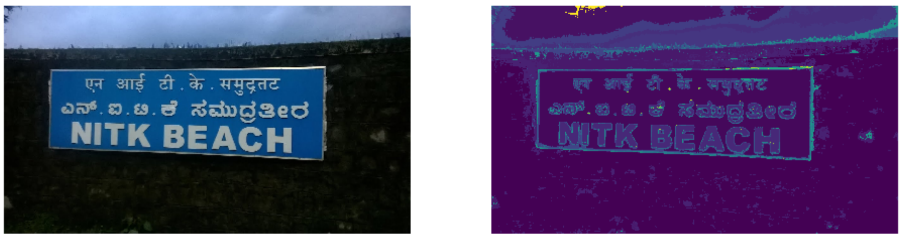

Figure 7: Bandwidth=10, Clusters=45

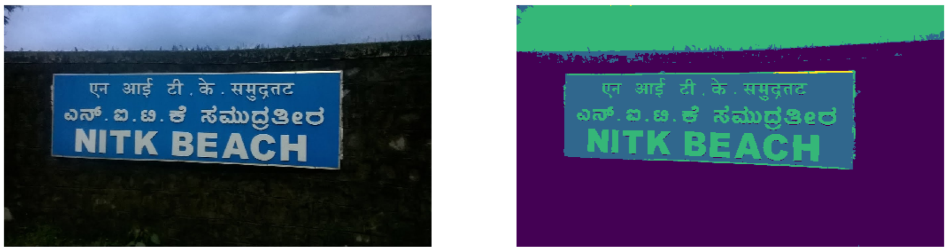

Figure 8: Bandwidth=40, Clusters=4

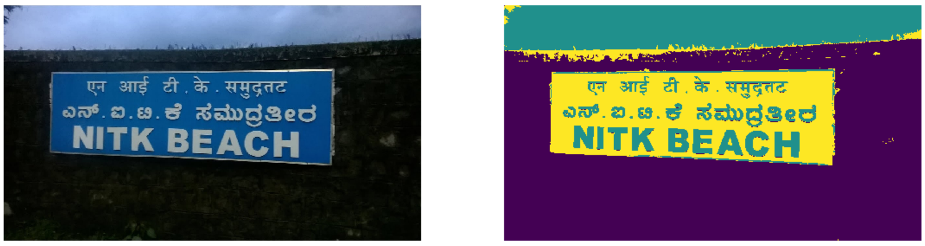

Figure 9: Bandwidth=80, Clusters=3

Adaptive bandwidth:
The mean shift with a constant bandwidth works decently well with pictures like above, but for some pictures like the one shown in Figure 9 and 10 where we want separate clusters for each kind of flower, it is not perfect.When a lower bandwidth is taken like 30 the different shades of some colors are being clustered too and we get 10 clusters.To avoid that if we increase the bandwidth to 60 we do get desired number but the boundaries are not uniform and there is some wrong clustering too due to bigger window size.To avoid that one can use adaptive mean shift clustering as mentioned in[2] where due to local scale information the window size can adjust accordingly.As mentioned in [2] KNN can be used to reflect upon the density distribution of the feature space and adjust bandwidth accordingly which is like including local scale information to choose bandwidth. Mean Shift can be used in combination along with KNN to get optimized results.

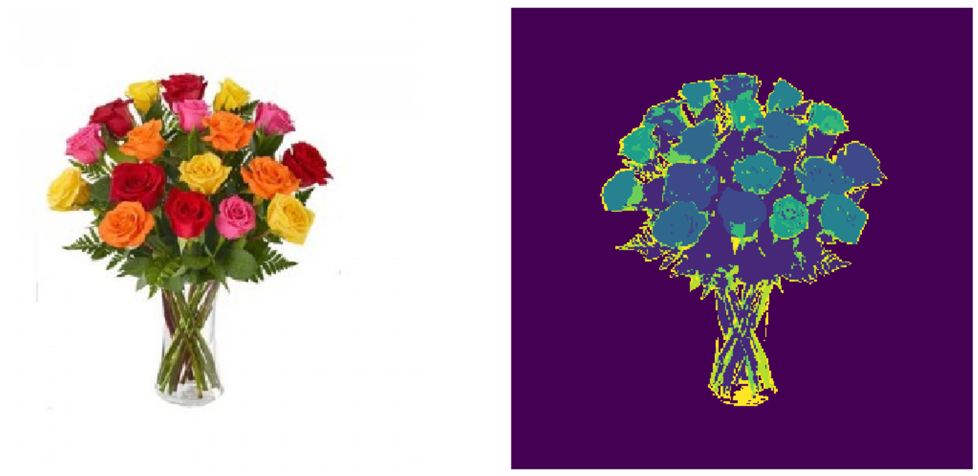

Figure 10: Bandwidth=30, Clusters=10

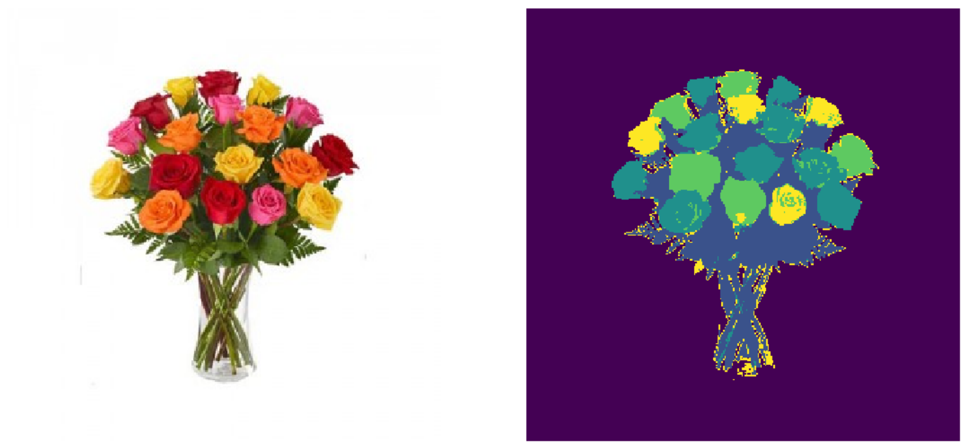

Figure 11: Bandwidth=60, Clusters=5

6 Conclusion and Future Scope

It is seen that the bandwidth is a very important parameter and tuning it correctly for a given application can have very significant impact on the clustering efficiency.
Appropriately varying it for every point can further increase the efficiency and even better results as compared to finding it manually.

7 References

Literature references

1. Cheng, H.D., Jiang, X.H., Sun, Y., Wang, J.: Color image segmentation: Advances and prospects. Pattern Recognit. (2001). doi:10.1016/S0031-3203(00)00149-7
2. Bo, S., Ma, Y., Zhu, C.: Image Segmentation by Nonparametric Color Clustering. In: Shi, Y., van Albada, G.D., Dongarra, J., and Sloot, P.M.A. (eds.) Computational Science – ICCS 2007. pp. 898–901. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
3. S. Bo, L. Ding and Y. Jing, "On Combining Region-Growing with Non-Parametric Clustering for Color Image Segmentation," 2008 Congress on Image and Signal Processing, Sanya, Hainan, 2008, pp. 715-719. doi: 10.1109/CISP.2008.275
4. S. Bo, L. Ding and Y. Jing, "On Combining Region-Growing with Non-Parametric Clustering for Color Image Segmentation," 2008 Congress on Image and Signal Processing, Sanya, Hainan, 2008, pp. 715-719. doi: 10.1109/CISP.2008.275
5. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011
6. Bogdan Georgescu, Ilan Shimshoni, and Peter Meer. 2003. Mean Shift Based Clustering in High Dimensions: A Texture Classification Example. In Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2 (ICCV '03), Vol. 2. IEEE Computer Society, Washington, DC, USA, 456-.
7. D. Comaniciu, V. Ramesh and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada, 2001, pp. 438-445 vol.1. doi: 10.1109/ICCV.2001.937550

Online references

1. https://www.quora.com/What-is-kernel-density-estimation
2. http://www.chioka.in/meanshift-algorithm-for-the-rest-of-us-python/
3. https://courses.csail.mit.edu/6.869/handouts/PAMIMeanshift.pdf
4. https://www.sciencedirect.com/science/article/pii/S0898122108005798
5. https://pdfs.semanticscholar.org/d8a4/d6c60d0b833a82cd92059891a6980ff54526.pdf
6. https://pythonprogramming.net/mean-shift-from-scratch-python-machine-learning-tutorial/
7. https://mathisonian.github.io/kde/
8. https://media.geeksforgeeks.org/wp-content/uploads/20190429213154/1354.png
9. https://www.youtube.com/watch?v=UqYde-LULfs
10. https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/