Image Segmentation

Machine learning Project

ARPIT SINGH

CSE - 5th semester

111601031

Abstract

This report is going to deal with the problem of image segmentation. First, it is going to introduce the problem of image segmentation, what all are the different methods used for image segmentation. Do some analysis on these different approaches and what are the advantages and disadvantages of different methods. Finally the report ends with conclusion and relative merits of each approach.

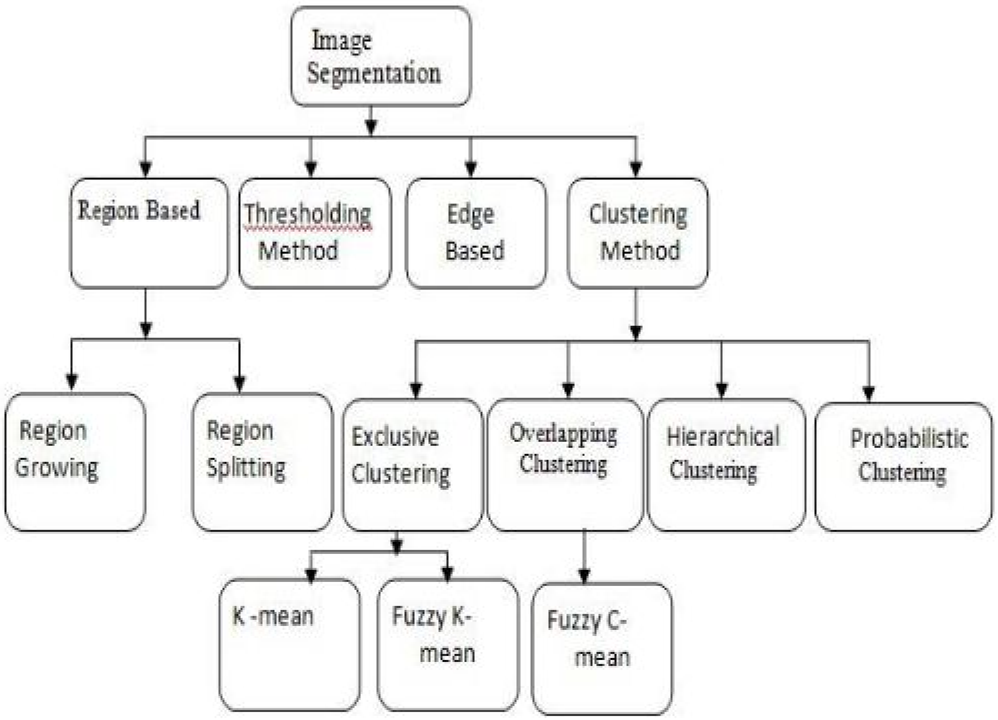
Introduction

Image Segmentation is process of extracting meaningful part from image. Image segmentation filters the important part from rest of image. Image segmentation is process of partitioning the digital image into multiple segments. These segment consist of set of pixels. The goal of segmentation is to simplify and change the representation of image into something meaningful and easier to analyze and process. Use of image segmentation is to locate object and boundaries in image and its process of assigning label to every pixel in image such that pixel with same label share certain characteristics.

Input of image segmentation is any digital form of image. Output is set of segments that collectively cover entire image set of contours extracted of entire image. Properties like gray scale, color, intensity, texture, depth help to recognize similar regions and similarity of such properties is used to construct groups of region. Image segmentation plays important role in human visual perception.

CLASSIFICATION OF IMAGE SEGMENTATION METHODS

Main methods which are used for image segmentation are given in the diagram below. This diagram is followed by the explanation of the basic methods.



Region based image segmentation

Region based methods, partition an image into regions that are similar according to a set of predefined criteria. Region based segmentation are based on following methods

1. **Region growing**

   Region growing is a procedure that group's pixels in whole image into sub regions based on predefined criteria.

   (i) Selection of seed pixels from given image.

   (ii) Selection of similarity criterion such as grey level intensity, shape, size, depth of pixel etc.

(iii) Grow regions by appending to each seed those neighboring pixels that have predefined properties similar to seed pixels.

(iv) Stop region growing when no more pixels met the criterion for inclusion in that region

2. Region splitting

Instead of choosing seed points, user can divide an image into a set of arbitrary unconnected regions and then merge the regions.

Thresholding method

The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T, or a white pixel if the image intensity is greater than that constant. In the project i have used the thresholding method based on the histogram values of the image pixels.

Edge based method

Edge detection is the process of identifying and locate discontinuities in an image.The discontinuities are abrupt changes in pixel intensity which characterize boundaries of object.

The **watershed** transform combines aspects of both **region- based** and **edge-based** approaches to image segmentation. The **regions** are built by pixel grouping (**region-based**), whereas the edges of the **regions** are located **based** on image discontinuities (**edge-based**).

Clustering method

Clustering is the process of organizing objects into groups whose members are similar in some way. This approach uses the clustering algorithms to cluster the pixels of the image depending on their distance in space (3D space in case of RGB values).If the resolution of the image is too high (say 1000 * 1000) then it would take a lot of computational time for the clustering algorithms to form the segments. So instead of using the original image we can use **low resolution image** or we can also use the concept of **superpixels.** Since if we are going to reduce the resolution of the image then we will lose a lot informations. So better would be to use the concept of superpixels.

Concept of super pixel

In order to make the clustering algorithms to find the segments in the image faster (in terms of computational time) we can use the superpixels instead of the original pixels. Super pixels are created by taking a large number of segments (say 5000) from the image. Then these super pixels can be used instead of the original pixels for the clustering algorithms. Spectral clustering with kmeans is one of the algorithms that is used for finding the super pixels in linear mind. Some of the advantages of the super-pixels are listed below -

1. It is **computationally efficient**: it reduces the complexity of images from hundreds of thousands of pixels to only a few hundred super pixels.
2. The super pixels are **perceptually meaningful**: each super pixel is a perceptually consistent unit, i.e. all pixels in a super pixel are most likely uniform in, say, color and texture.
3. It is **near-complete**: because super pixels are results of an over segmentation, **most structures in the image are conserved**.There is very little loss in moving from the pixel-grid to the super pixel map.

Image preprocessing

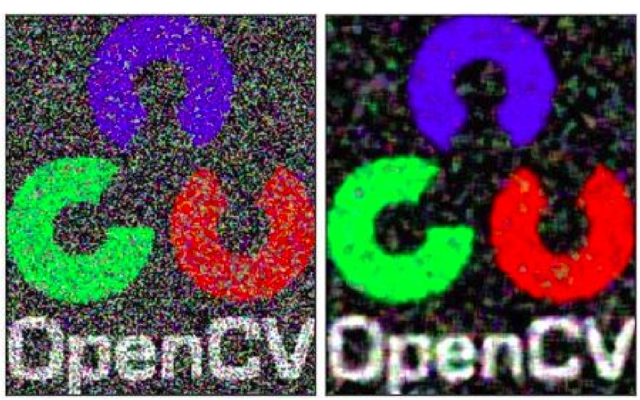
This can be used in the preprocessing image so as to get the better segments of the image. Opencv provide some facilities for smoothening of the image, changing the contrast of the image, removing the noise. Some of the preprocessing steps used in this project are as follows :

1. **Gaussian Blur**

In image processing, a **Gaussian blur** (also known as **Gaussian smoothing**) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.

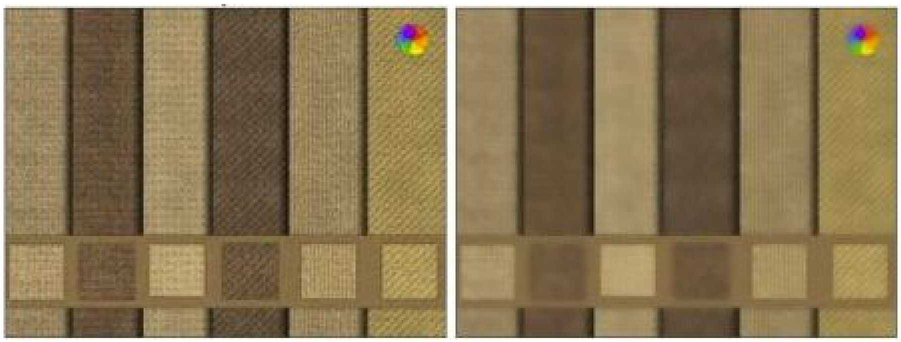
2. **Median Blurring**

It takes median of all the pixels under kernel area and central element is replaced with this median value. This is highly effective against salt-and-pepper noise in the images.But in median blurring, central element is always replaced by some pixel value in the image. It reduces the noise effectively. Its kernel size should be a positive odd integer.



3. Bilateral Filtering

It is highly effective in noise removal while keeping edges sharp. But the operation is slower compared to other filters. Gaussian filter takes the neighbourhood around the pixel and find its gaussian weighted average. This gaussian filter is a function of space alone, that is, nearby pixels are considered while filtering. It doesn't consider whether pixels have almost same intensity. It doesn't consider whether pixel is an edge pixel or not. So it blurs the edges also, which we don't want to do



Approach 1 (kmeans with image preprocessing)

In the first approach use the image pre processing steps to smoothen the image and remove the noise from the image (make use of the filters mentioned in the section above). Then use the kmeans clustering method to find the segments of the image. We have to mention the value of k for the kmeans algorithm. Note that we are using the kmeans step here to cluster the pixels of the image instead of the superpixels. Using kmeans with superpixels is mentioned in the next approach.

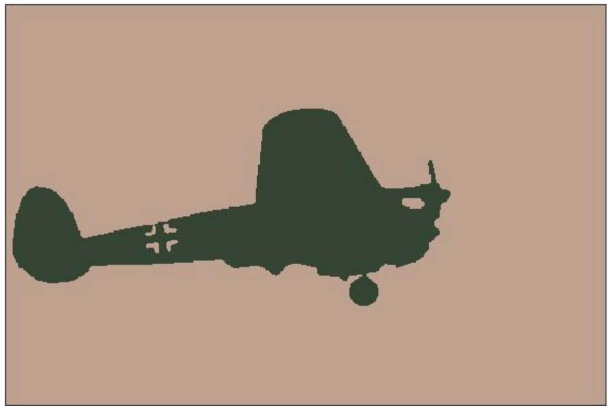
Example (very good result) :-

Original image

After gaussian filtering

Median filtering

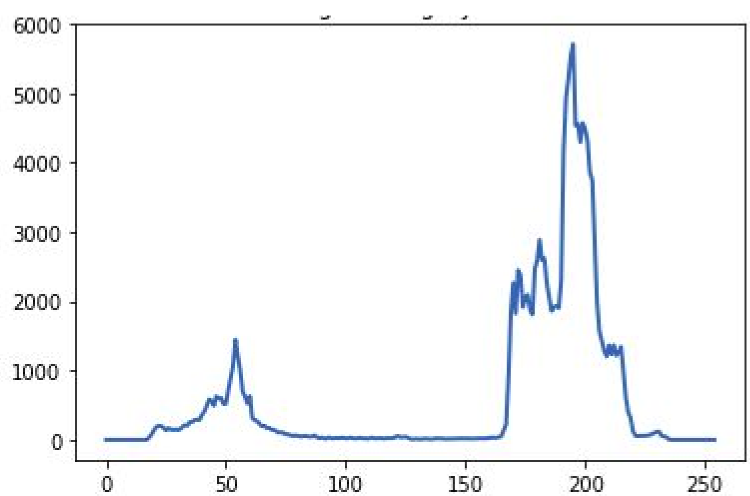
Bilateral filtering



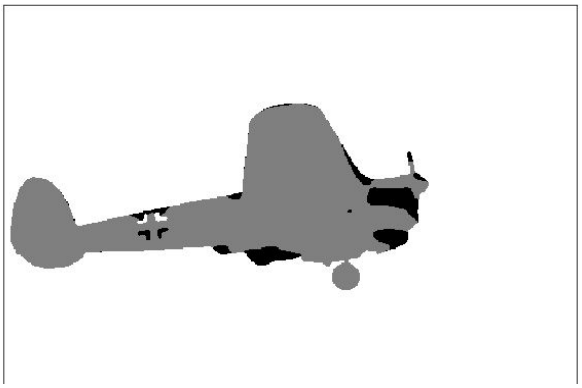
Final output with kmeans using K=2

Approach 2 (Thresholding method)

In this approach we are going to select one of the values from the RGB(either R or G or B). Plot the histogram of the pixel values. Then find the **threshold** from the histogram and then put the threshold on the image to form the foreground and the background segmented image. Once you have the segmented pixels we can construct the segmented image with the foreground pixels as the original pixels and the background pixels as the white pixel. Note this method is going to work very good if there are two distinguishable peaks in the histogram (i.e. the histogram can be separated by putting some threshold).



Histogram of the pixel values



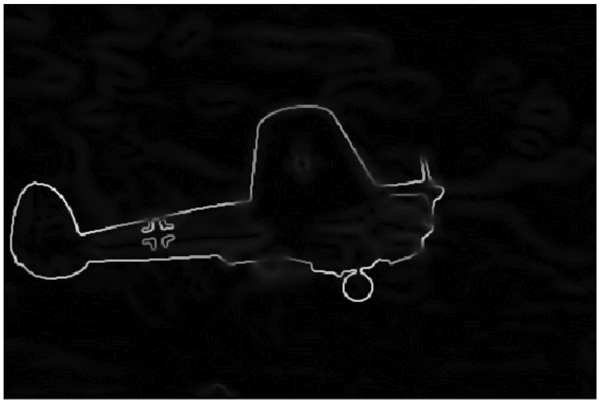
After setting the marker values

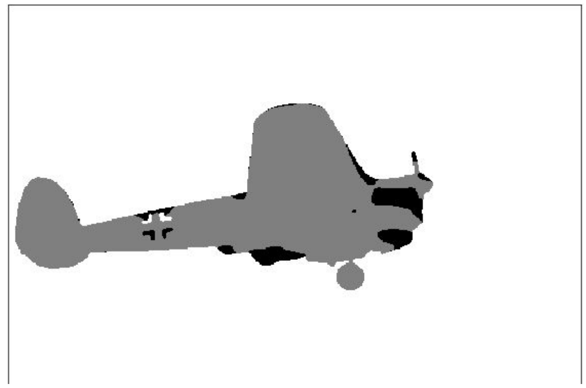
(Thresholded image)

After the reconstruction of the image from the thresholded image

Approach 3 (Watershed algorithm)

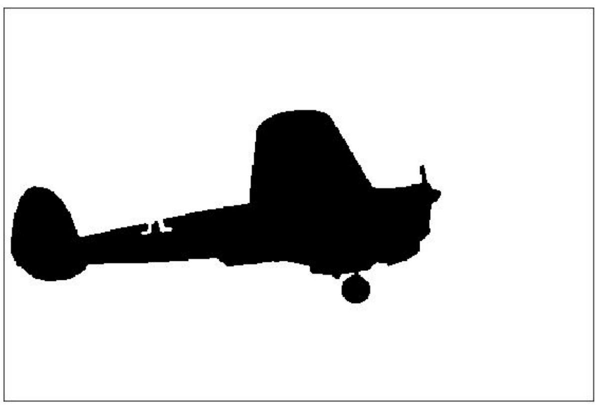
In this approach we are using the watershed algorithm to find the segments of the image.The watershed transformation treats the image it operates upon like a **topographic map**, with the brightness of each point representing its height, and finds the lines that run along the tops of ridges. Starting from user-defined markers, the watershed algorithm treats pixels values as a local topography (elevation). The algorithm **floods basins from the markers**, until basins attributed to different markers meet on watershed lines. In many cases, markers are chosen as local minima of the image, from which basins are flooded. Finally we can reconstruct the segmented image with the foreground part as the original pixel and the background part as the white pixels.





Elevation map (this is to find the ridges, edge magnitude)

Marker image



Watershed algorithm (it uses the evaluation map and marker image given above)

Reconstruction of the segmented image from the watershed image.

Below is the comparison of the above three mention methods on different images.

Original image

Kmeans (K = 2)

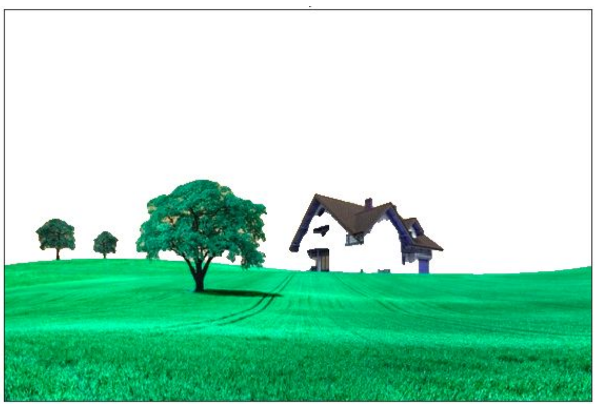
Watershed algorithm

Thresholded image

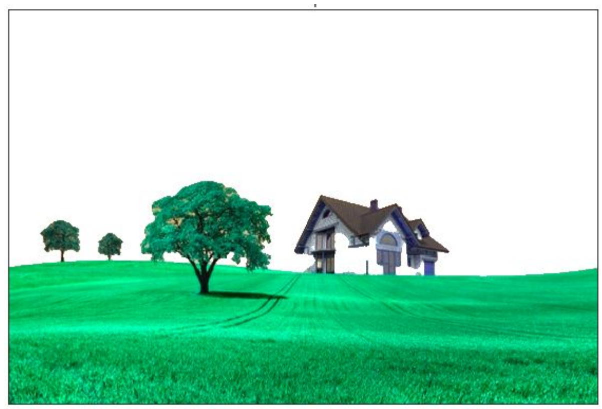
Performing Okay (examples)

Original image

Kmeans (k =2)



Watershed algorithm

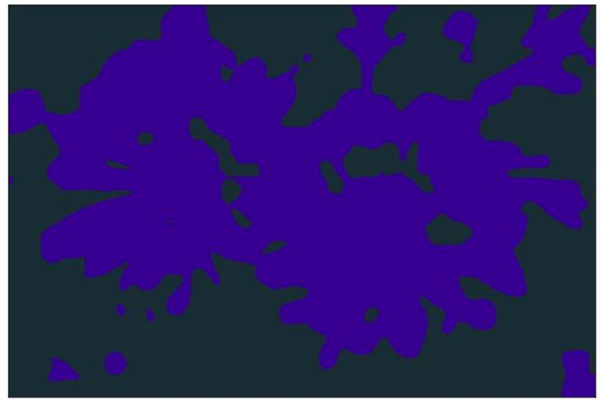


Threshold segmentation

Performing very badly

This is one example of the output where all the above mentioned methods are not performing good. For the watershed and threshold segmentation since there is no proper threshold , that is why the results are very bad.

Original image



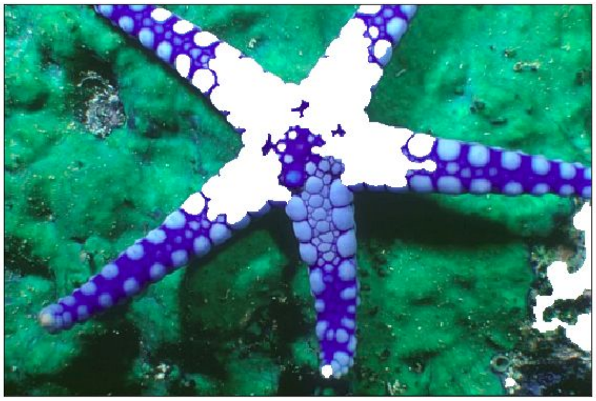
Kmeans (K = 2)

Watershed segmentation

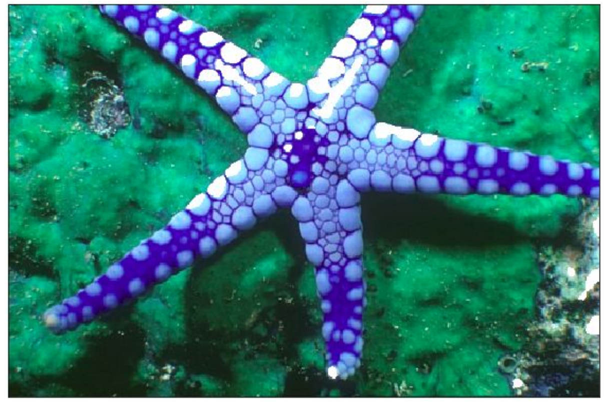
Threshold segmentation

Original image

Kmeans (K = 2)



Watershed segmentation



Threshold segmentation

Need for other methods

All the above mentioned methods are not able to segment the images where there are more than two segments with different colors and textures.Moreover, watershed and threshold segmentation are used to separate the foreground from the background, that is why, they work well where there is a clear distinction between the background and the foreground.

Approach 4 (Superpixels and clustering algorithms)

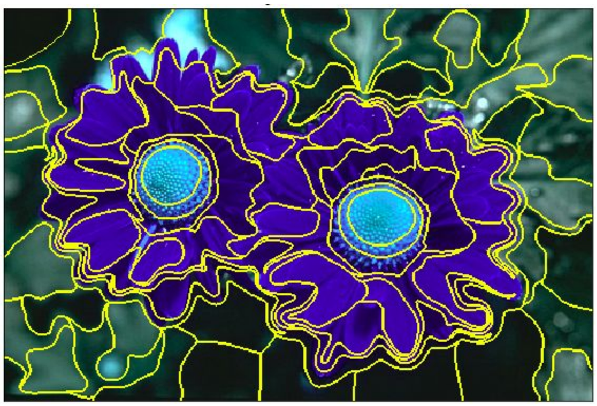
We can use different clustering algorithms for image segmentation but since the resolution of the image is very high so it will be computationally very expensive for algorithms like **DBSCAN** , **agglomerative clustering** and all to find the clusters of pixels. So we can create **superpixels** by segmenting the original image into large number of segments (say 5000) and then replace the segments with the mean of the segments. Now we can use the clustering algorithms to cluster the super pixels and then reconstruct the segmented image from the super pixel segments.

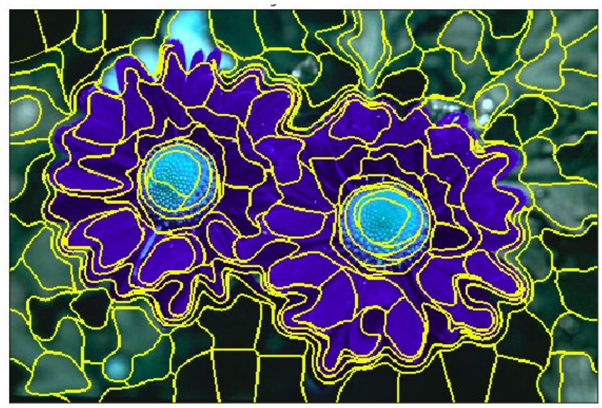
Creating the superpixels

Below we are going to see what is the effect of segmenting an image into different number of segments. Different segments are separated by the boundary. Note in this segmentation we are ensuring that regionally connectivity is maintained in between the different segments.

Segmented image with 5 segments

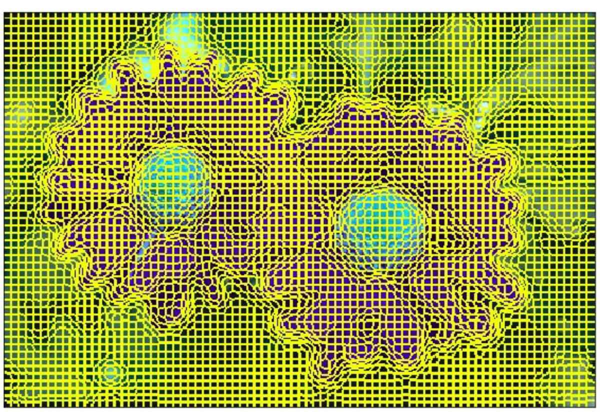
Original image





Segmented image (200 segments)

Segmented image (100 segments)



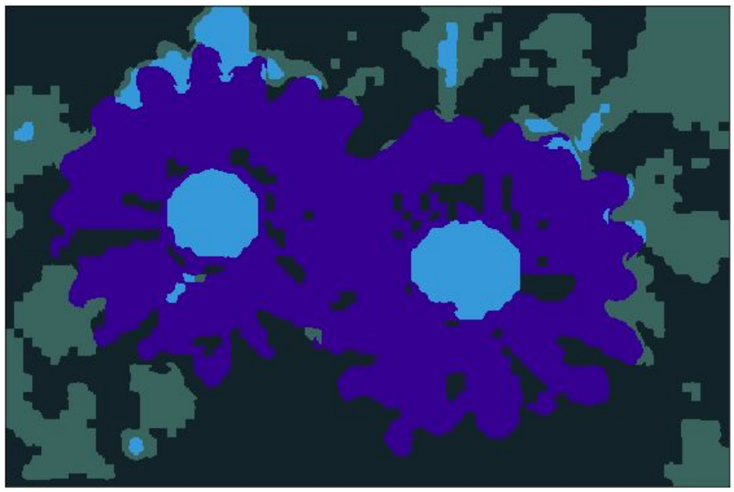
Segmented image (5000 segments)

Reconstruct of the original image by replacing each superpixel (segments) with the mean of the pixels in the segments. Number of segments here are 5000.

Using the clustering algorithms

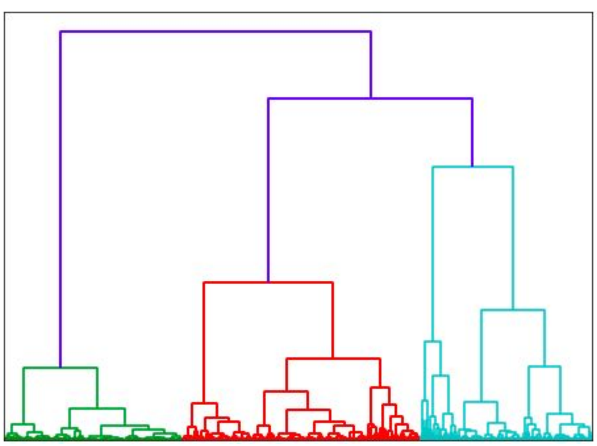
Now we can use the clustering algorithms to find the cluster in between the superpixels and then reconstruct the segmented image from the clustered super pixels segments.

Kmeans

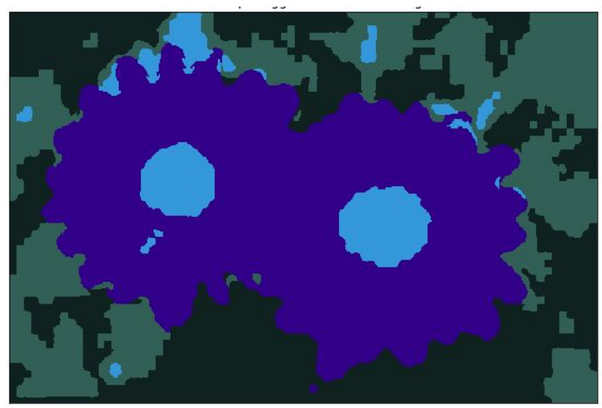


Kmeans with k = 4 on the superpixels.

Agglomerative clustering

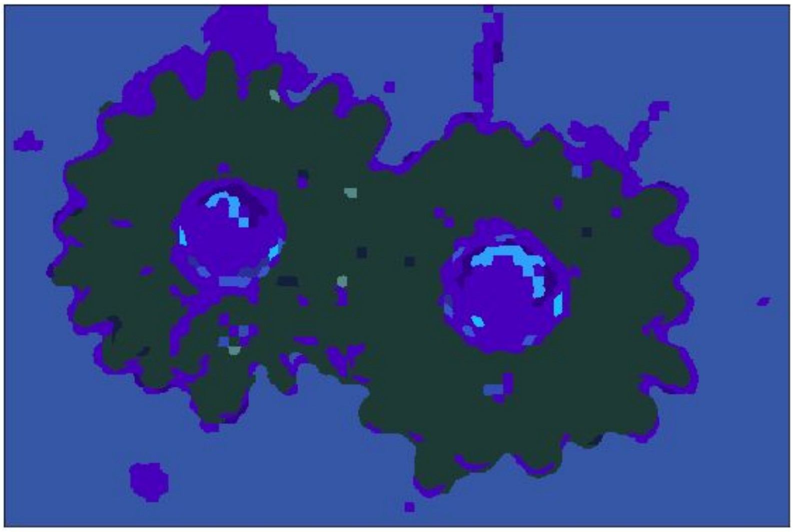


Dendogram for agglomerative cluster.This can be useful to find the appropriate number of clusters in the image



Clustering the image with into three segments using the agglomerative clustering (3 segments and ward linkage)

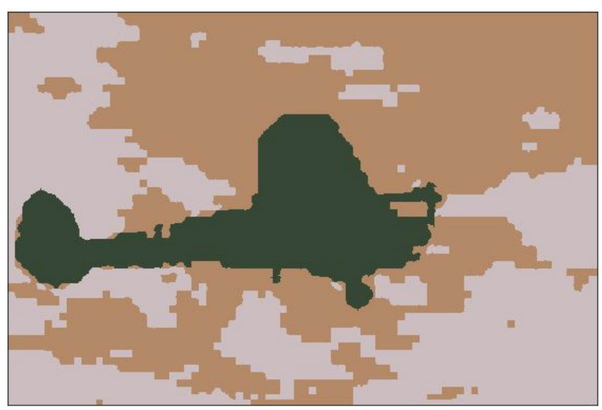
DBSCAN



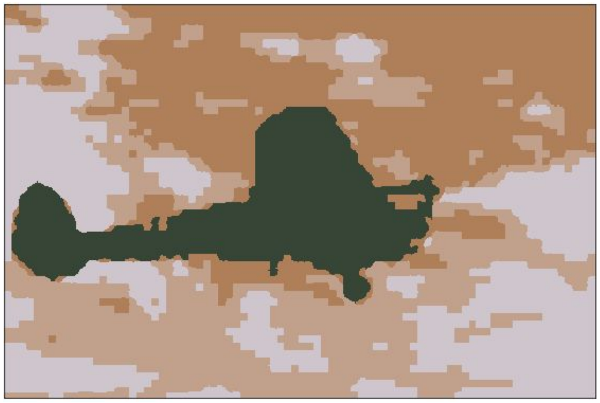
With DBSCAN clustering (eps = 5). Number of clusters it has given is 13 with 3 major clusters and other with very few elements. That is the reason for viewing some noise in the image.

Some examples of the above method

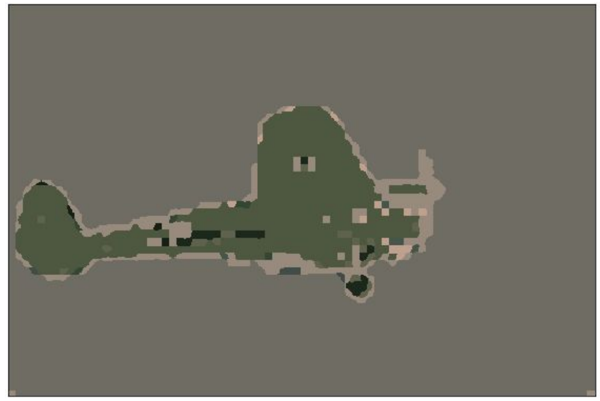
Original image



Kmeans (K = 3)
Trying to separate the cloud, plane and sky



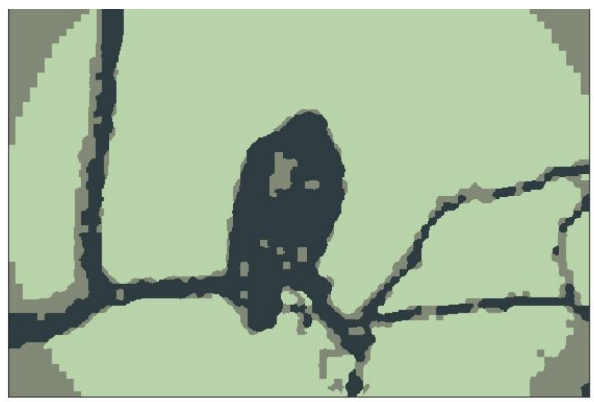
Agglomerative clustering (K = 3)



DBSCAN (eps = 4.5)
Small noises but able to segment the plane from the background.

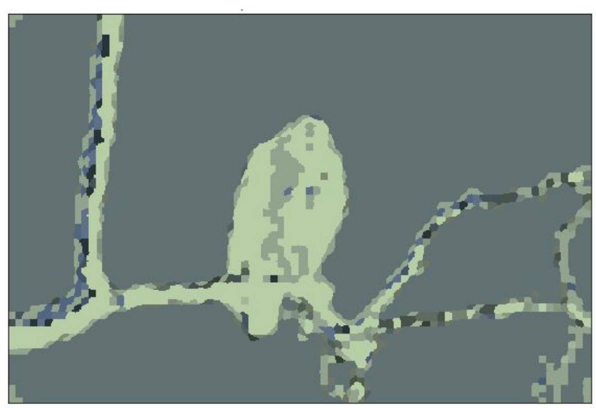
Example 2

Original image



Kmeans (K = 3)

Agglomerative clustering (k = 3)



DBSCAN (eps = 4.5)

Examples (performing okay)

Original image

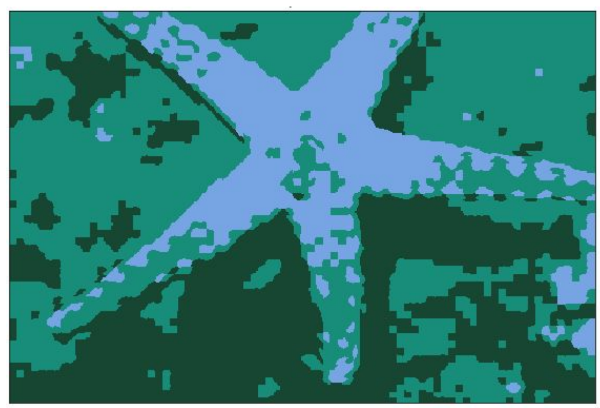
Kmeans (k = 3)

Agglomerative clustering (k = 4)

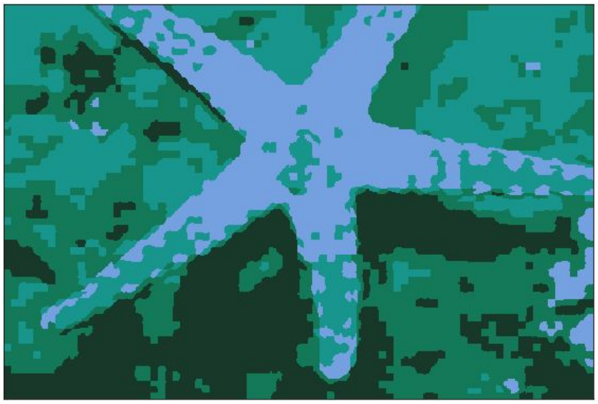
DBSCAN (eps = 5.5)

Example (okay performance)

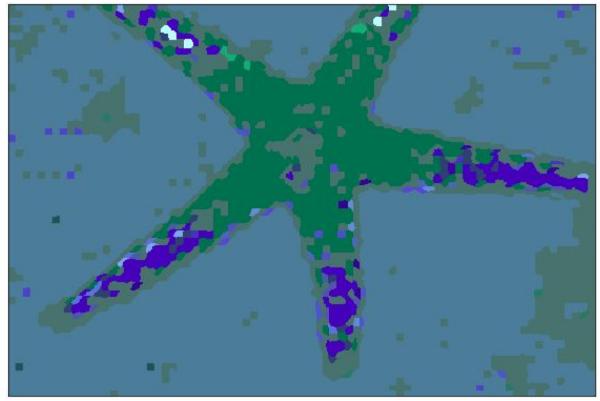
Original image



Kmeans (k = 3)



Agglomerative clustering



DBSCAN (eps = 5.5)
Note all other approaches mentioned above are performing very badly on this image but super pixels with DBSCAN is doing a decent job of segmenting the starfish from the background.

Analysis

Here i will analyse the different approaches used so far and what are the merits and demerits of these approaches.

1. **Kmeans (with image preprocessing , without superpixel)**

   Advantages

   - It is one of the easiest method used in order to segment the image.
   - Computationally faster when the k value is not very large.
   - Reconstruction of the segmented image from the cluster centers is very simple.
   - Works very well for images where there is clear distinction in between the different segments.

   Disadvantages

   - Computationally very slow when the value of k is large (more than 30).
   - We have to mention the value of k in advance which is not necessary that we always know in advance.
   - Do not take into account the **regional connectivity** in between the segments.

Note that we cannot use this kind of clustering algorithms with the high resolution image always as the number of pixels will be very large and so the computational time and the memory required is going to be very large. That is the reason why we cannot use algorithms like DBSCAN, agglomerative clustering directly.

2. **Threshold segmentation**

Advantages

- Computationally faster and memory efficient.
- Segments the image with clear distinction of the background and the foreground segments.

Disadvantages

- We need to tune the value of T (threshold value in the histogram) for finding the correct segments.
- Works very poorly for image where there are large number of colors, textures.

3. **Watershed algorithm**

Advantages

- It combines the effect of **edge based** and **region based** segmentation.
- Computationally faster and memory efficient.
- The resulting boundaries form closed and connected regions. Traditional edge based techniques most often form disconnected boundaries that need post-processing to produce closed regions.
- It is best recommended to use if the foreground and the background are having the contrast colors.

Disadvantages

- Works poorly if there are segments that are not connected.
- Works poorly for images if there is no particular boundary separating the two segments.
- Works poorly for the most natural images. (as the segments are not that easily separable).

4 Kmeans using superpixels

Advantages

- Since we are using superpixels which are very small in number as compared to that of the total number of pixels, so it is computationally much faster as compared to the original kmeans.
- Works well even for larger values of k.
- It preserves all the properties of kmeans on the original images.

Disadvantages

- Since we are using the superpixels, so edges of the segments may not be that smooth. It may look like it is made up of small blocks.
- Do not take regional connectivity in the segments into account.
- Need to mention value of k in advance.But since the number of superpixels are not that high **elbow method** and other methods can be used to find the appropriate value of k.

5. Agglomerative clustering with superpixels

Advantages

- We cannot use agglomerative clustering directly with the original pixels of the high resolution image, but this method can be used to cluster lesser number of superpixels, So indirectly approximating the use of agglomerative clustering on the original image.
- We can plot the **dendogram** and this can be used to get the appropriate number of clusters. This can also be used to analyse how closely the segments are related to each other.
- We can use different linkage methods to get different form of segmentation.

Disadvantages

- Need to mention the number of segment while using this.
- It is very sensitive to the noise (outliers).

6. DBSCAN clustering with superpixels

Advantages

- This is by far one of the best methods for segmenting the natural images. Just need to tune the parameters correctly.
- With original pixels, it is very computationally expensive to use the DBSCAN but with superpixels it is possible to use it efficiently.
- We no need to mention the number of clusters. Setting the appropriate value of the **min_samples** and **eps** will give the desired result.
- It also take into account the **regional connectivity.**

Disadvantages

- Since we are not mentioning the number of segments so it may produce large number of segments with most of the segments very small. This may look like **noise in the segmented image**.
- Need to tune the parameters for all the images. No way to generalize the parameters.
- Since we are using the superpixels, so the edges of the segments may not be very smooth (as compared to normal kmeans or watershed algorithms).
- Images with clear background and the foreground perform better with watershed algorithm.

Conclusion

In this report I have addressed the problem of image segmentation and explored some of the standard methods that can be used for segmenting the image. There is no particular evaluation metric that can address weather the segment formed from the image is good or not. The goodness of the segment depends on the application. So sometimes methods that are appropriate for segmenting the image in one case may not be that good in other cases. So this report presented the vast range of methods that are available to segment the image. Still there is a room for lot of improvement and other advance methods like **Deep learning for image segmentation, Spectral clustering** methods can be used for better results.