HACETTEPE UNIVERSITY

AIN432-IMAGE PROCESSING

K-Means Clustering for Image Segmentation

IZZET
AHMET
2210765003

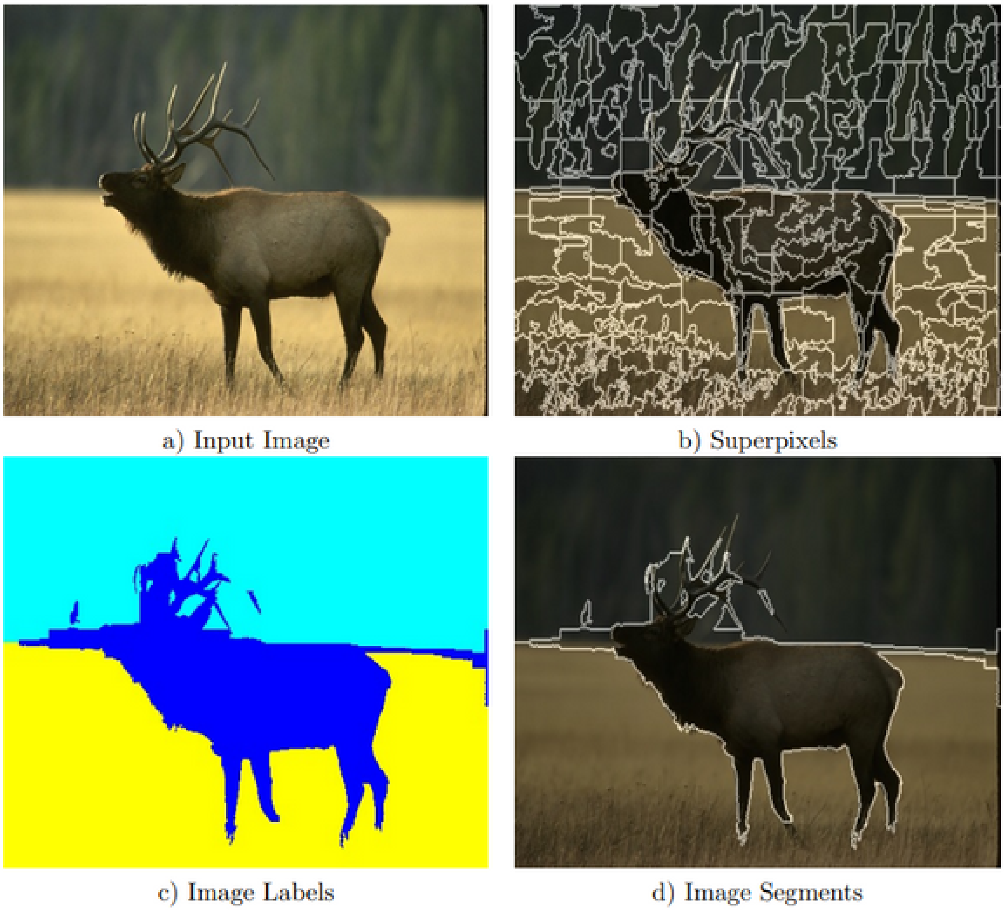
Table Of Contents

- Background Of Project
- Overview
- Different Methods On Different Images
- Conclusion
- Result
- References
- Additional Method

Background Of Project

Clustering is a process that groups data with respect to data similarity so that similar data take part same cluster. In image domain clustering is used for various types of problem e.g. image quantization, image segmentation. A good clustering algorithm must group data to homogeneous subsets as possible. Similarity is most critical step in a clustering algorithm that determine how the clustering algorithm groups data. Kmeans clustering is one of the most popular clustering algorithm that groups data to k dissimilar clusters. It is an unsupervised learning algorithm for clustering problem and the main idea is to define k centroids one of each cluster. These centroids is randomly assigned to data space for first iteration. In the next steps, for each data point, distance to these centroids is calculated and data points are assigned to nearest centroids as cluster elements. Then for each cluster, new k centroids are calculated from k clusters. This steps go on until clusters centroids unchanged.

Overview



a) Input Image b) Superpixels c) Image Labels d) Image Segments

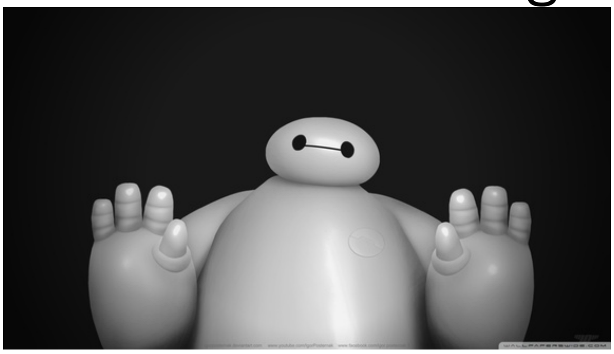
Figure 1: K-Means Image Segmentation using Superpixels

The goal of this project is to implement a K-means clustering algorithm for image segmentation by using pixel-level and superpixel representation of an input image and applying it to Pixel-Level Features (RGB and RGB with location) and Superpixel-Level Features (Mean of RGB color values, RGB color histogram and Mean of Gabor filter responses).

Overview

In this project, the models were tested and compared with different parameters on 5 selected different images. These pictures are selected by their certain features:

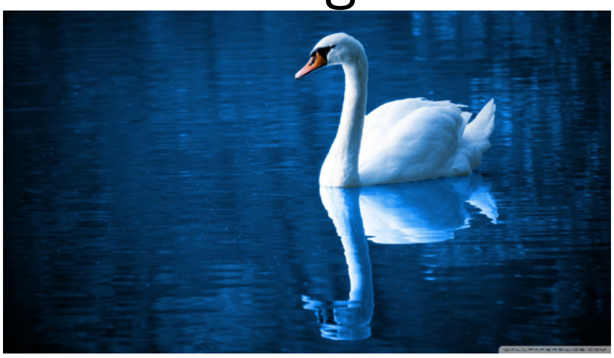
Black & White Image



One Character with Multiple Backgrounds

More Character With Same Colors

One Character with One Backgrounds



More Character With DifferentColors

Image Segmentation With Pixel-Level Features

RGB Color Feature

First, normalization was applied to the images, and then KMeans was applied with different parameters. Here are the results for different images and comments about them:

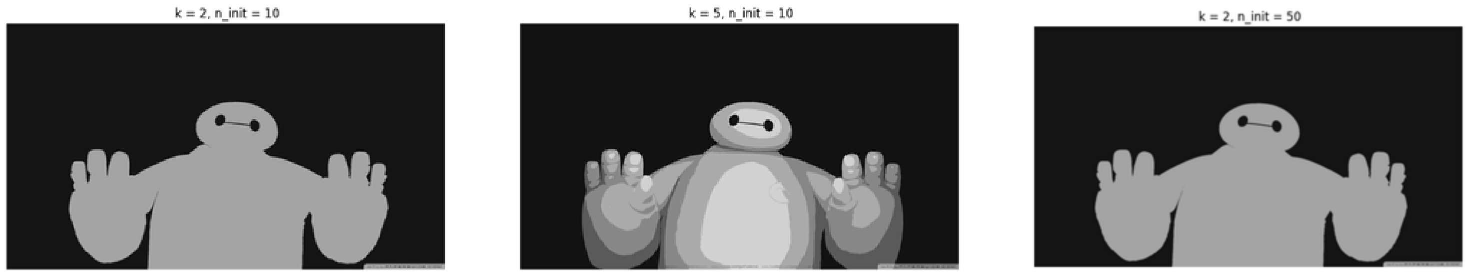


Figure 1

The image includes only one white character with a black background. So we need to select k = 2, otherwise our segmentation will not be so perfect. On the other hand, if we select the number of iterations larger, there will be no differences at all. So the best segmentation is k = 2 and n_init = 10.

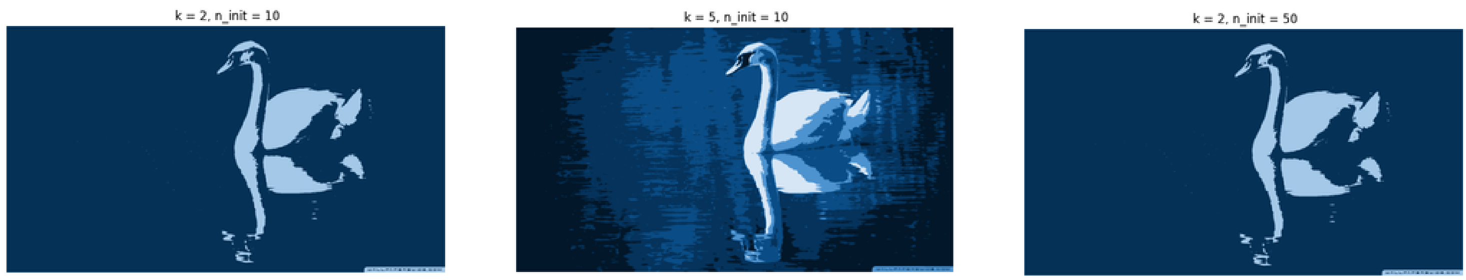


Figure 2

The image includes only one white character with a blue background but the results are the same. The only difference is, if we increase number of cluster, it tries to captures unnecessary details

Figure 3

The image includes only one white character with two backgrounds but the results are same with second image

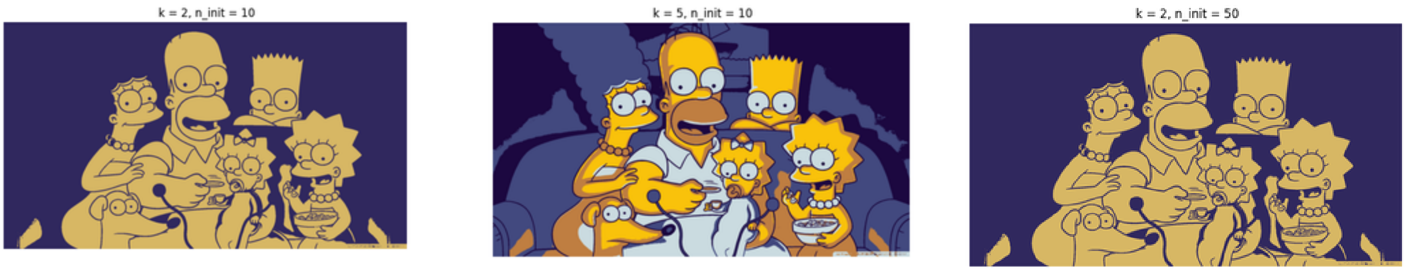


Figure 4

We could expect different results when we have more character with same color but segmentation still pretty good

Figure 5

From comparing Figure 5 - Figure 4 color variety doesn't affect the result so much.

By looking at all the figures above we can conclude that, if the image has little diversity it gets better segmentation.

To make an easier comparison between models and to keep the report short, we will continue by taking k = 2 and the number of iterations = 10 to

RGB Color & Spatial Location Features

First, normalization was applied to the images (divide r,g,b with 255 and x with max width and y with max height), and then KMeans was applied with k = 2 and n_init = 10. Here are the results for different images and comments about them:

RGB RGBXY

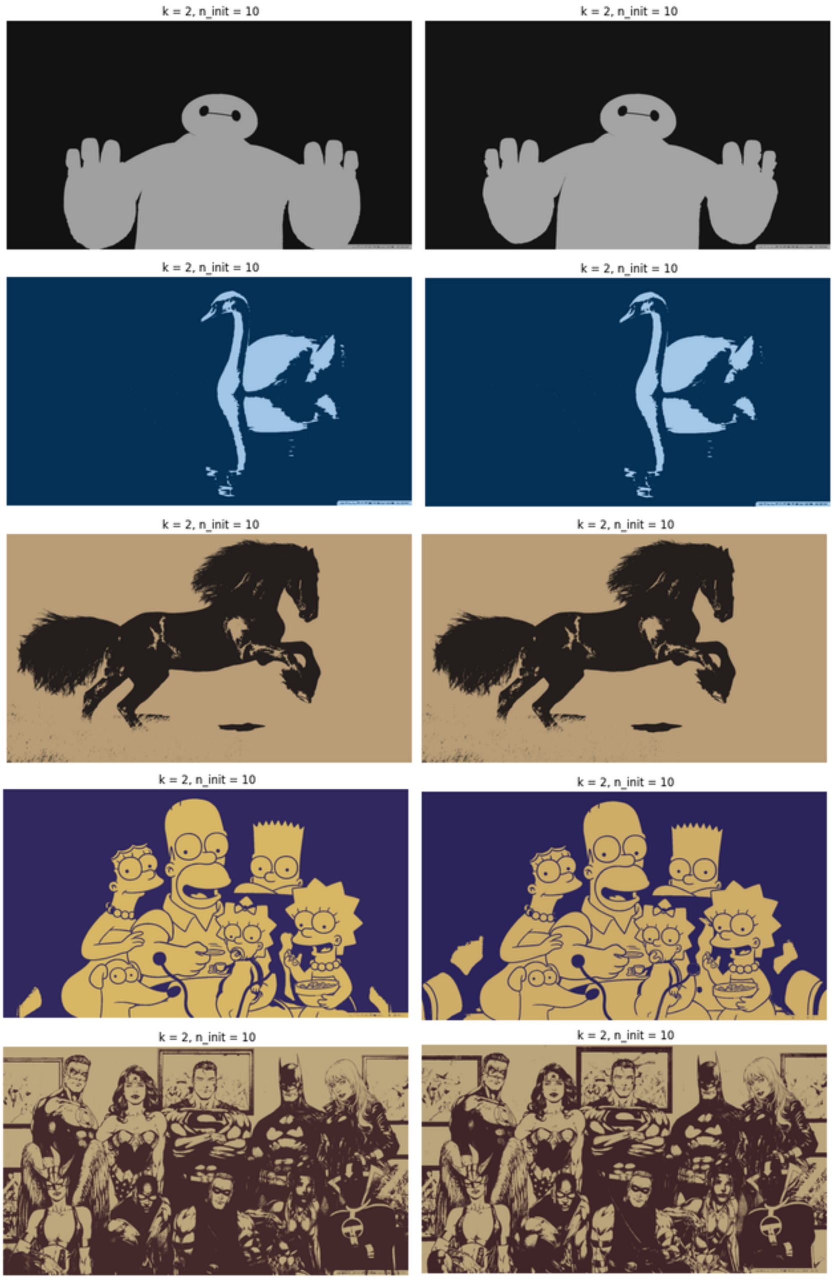


Figure 6

As we can see, there see approcismately no differences between rgb and rgb with location used segmentation for k = 2. To see differences between these two, lets try with increasing number of cluster

RGB RGBXY

Figure 7

By looking at Figure 7, it is obvious that using color features with spatial locations is better than using just color features.

Looking at figures 6 and 7, we can say that as the number of clusters increases, the rgb with location algorithm provides a better segmentation compared to using only rgb.

Segmentation With Superpixel-Level Features

Superpixels are perceptually meaningful and compact image regions obtained through a grouping process that groups pixels with similar properties, such as color or texture, into a single entity. Unlike individual pixels, superpixels represent higher-level structures in an image. The goal is to simplify the image representation while preserving important details.

Mean of RGB Color Values in Superpixel

Lets apply this method and compare result with different parameters

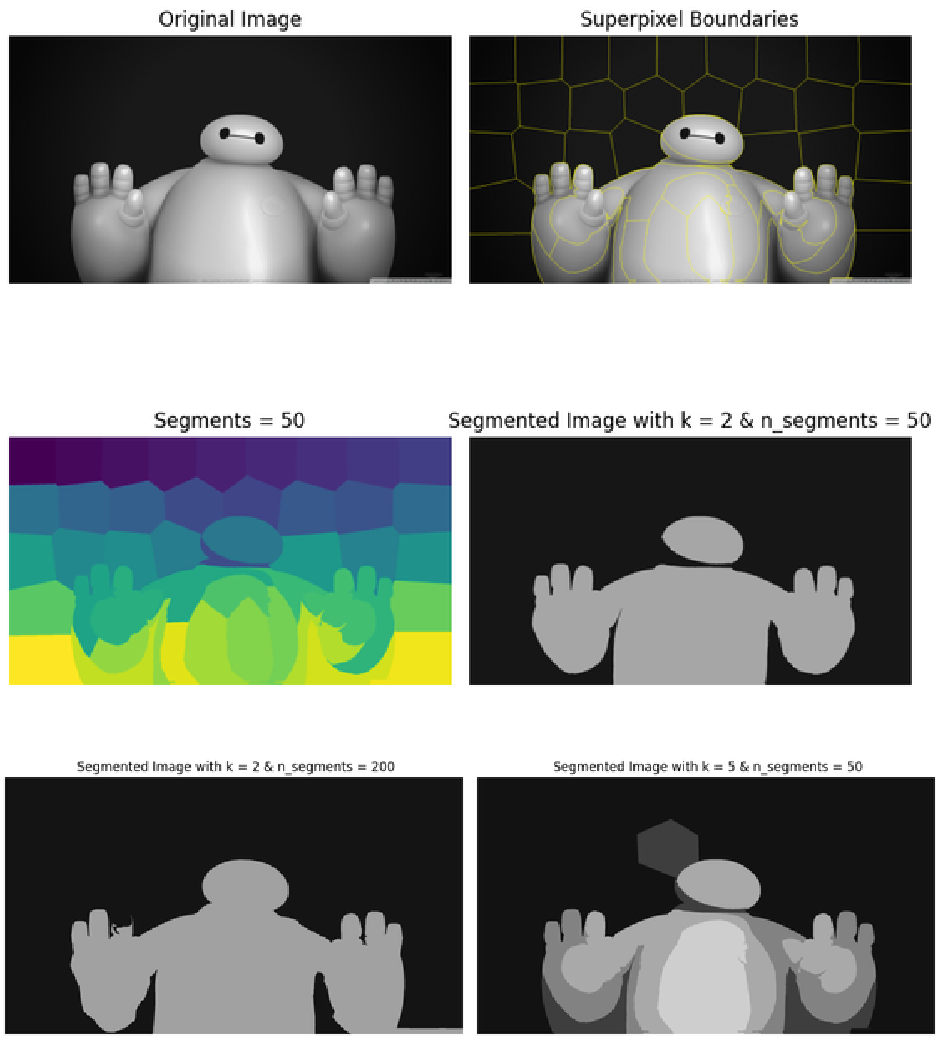


Figure 8

The image consists of black and white and only one character. By looking at the three responses of the method we can say increasing the number of clusters is not a good choice for this image because this image only has 2 objects and the increasing number of segmentation also not a good choice because this image doesn’t have much detail

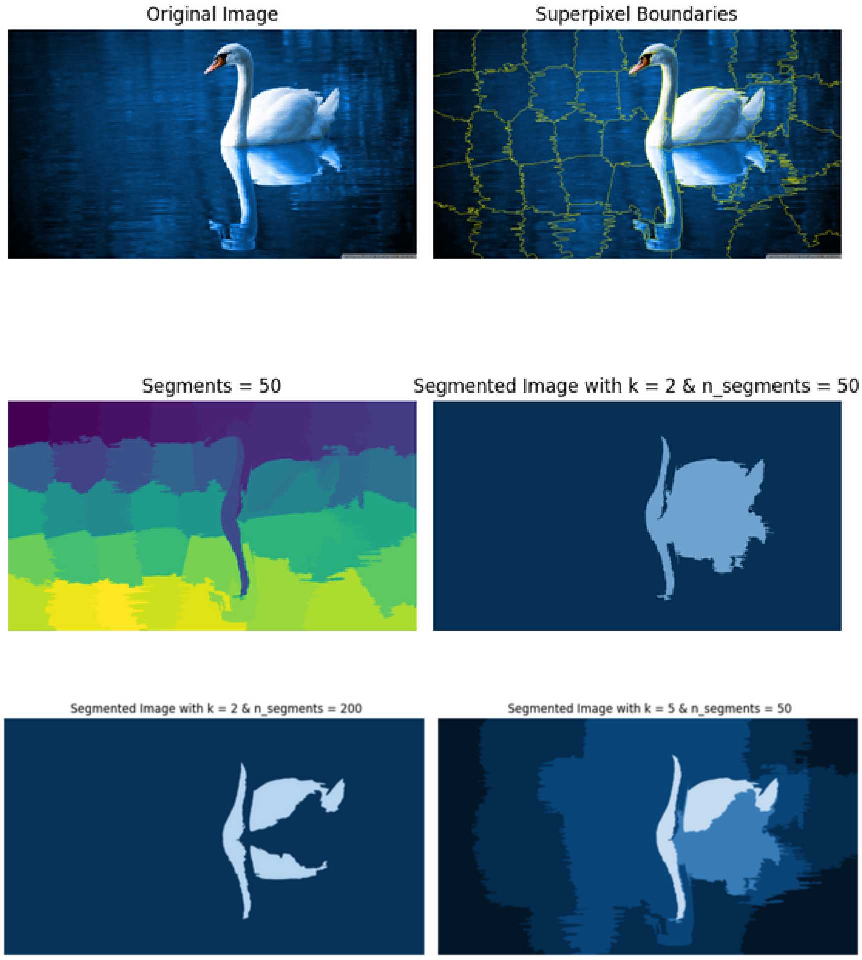


Figure 9

The image consists of mostly blue and white and only one character. By looking at the three responses of the method we can say increasing the number of clusters is not a good choice for this image because this image only has 2 objects and the increasing number of segmentations also not a good choice because this image doesn’t have much detail

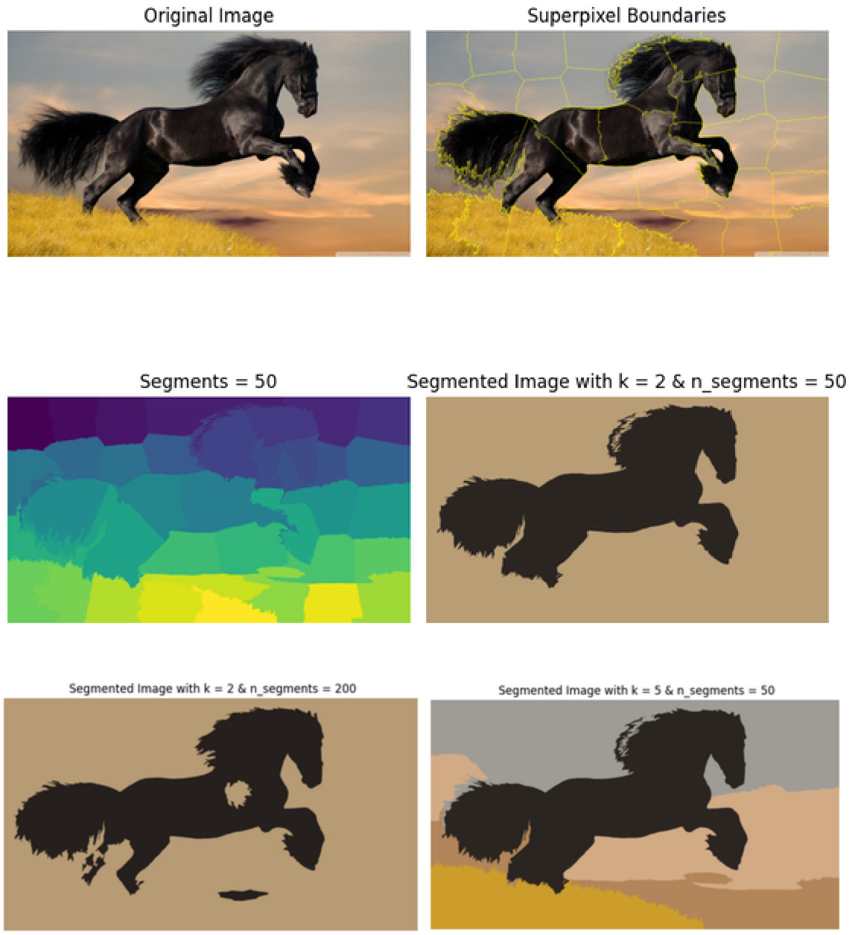


Figure 10

The image consists of one black horse and two different backgrounds. So increasing the number of clusters will make our segmentation better as we can see from comparing the forth and sixth images. But increasing the number of segmentations is not a good option too because we don't have much detail in this image too

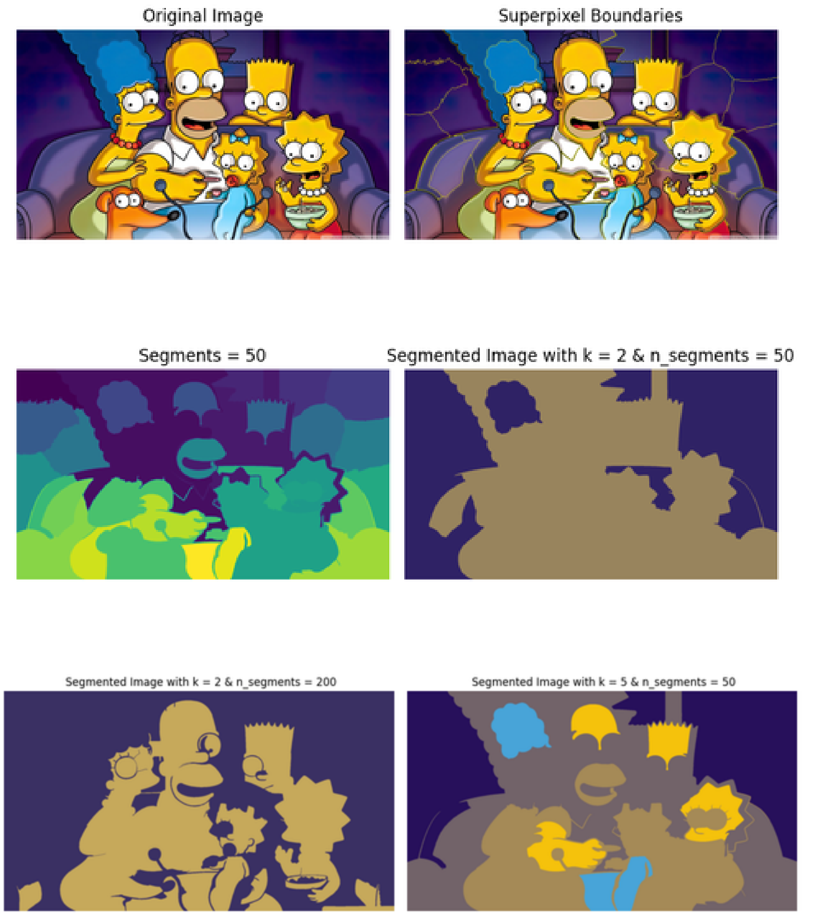


Figure 11

This image is more detailed and more colorful than the images placed above, so increasing the number of segments must affect our segmentations in a better way. Also, increasing the number of clusters affects in a good way for segmentation. We can see the differences by referring fourth image and comparing it with the fifth and sixth images

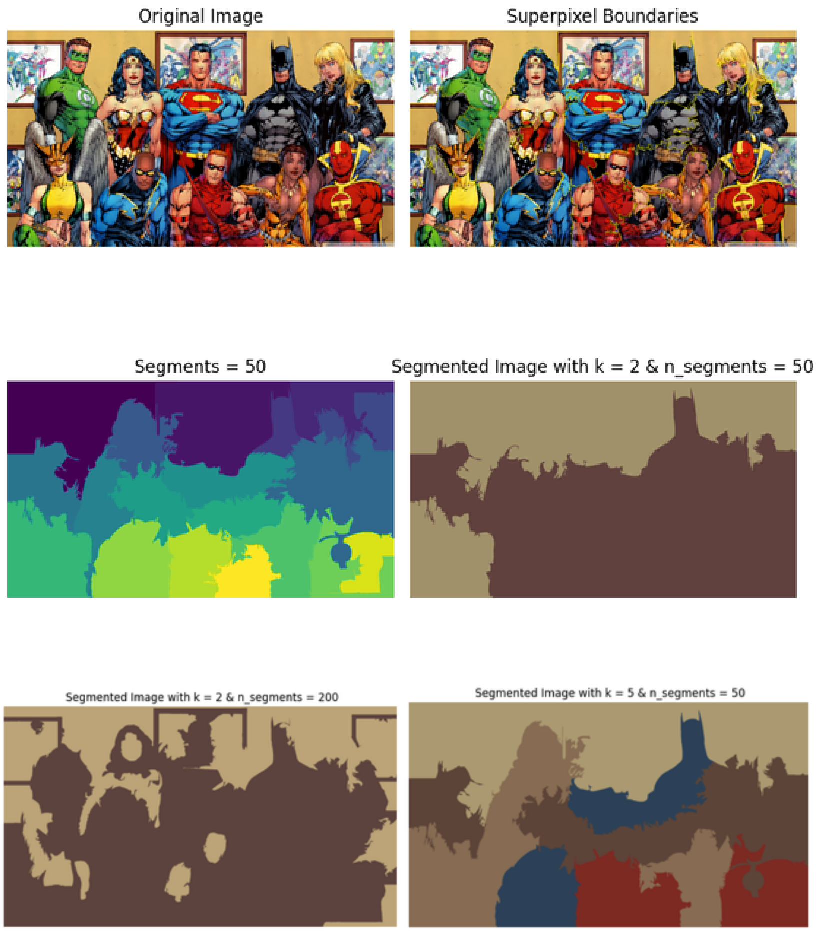


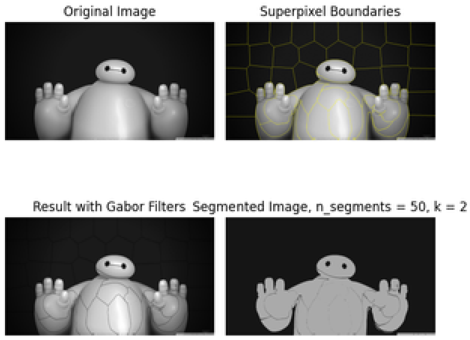
Figure 12

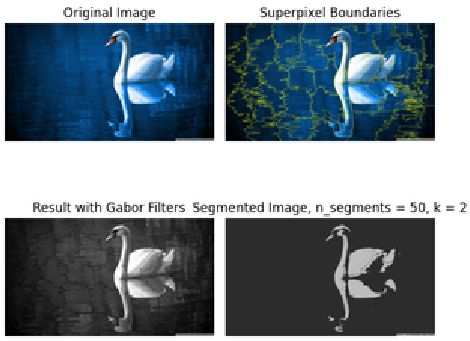
This image is more detailed and more colorful than the image in Figure 11. But it also means the segmentation is harder. But increasing the number of segmentations will make our method better because this image has much detail but increasing the number of clusters is not a good way because the image is very divers

Mean of Gabor Filter Responses

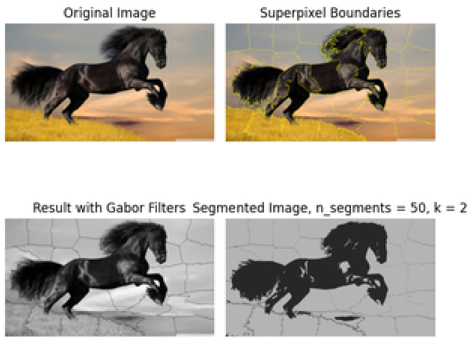
The mean of Gabor filter responses in superpixels involves averaging the Gabor filter outputs within each compact and perceptually meaningful image region, known as a superpixel, thereby providing an effective method for texture analysis.

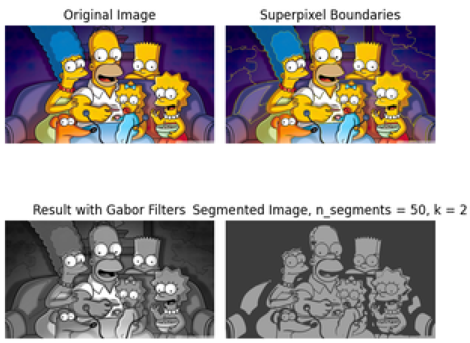
Try with number of segments = 50 and number of clusters = 2



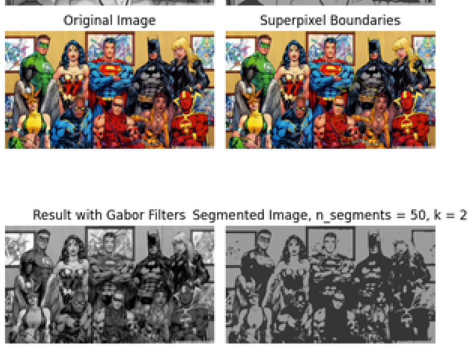


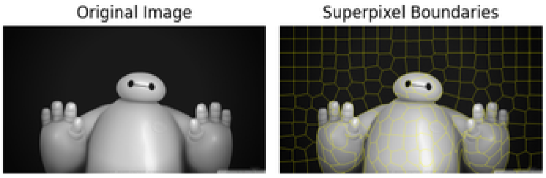
The Gabor bank (16 different gabor filter) is formed by utilizing each different parameter with each other.
ksize = 9
sigma = [1.0, 5.0]
theta = [0, np.pi/4]
lambd = [1.0, 5.0]
gamma = [0.1, 0.5]

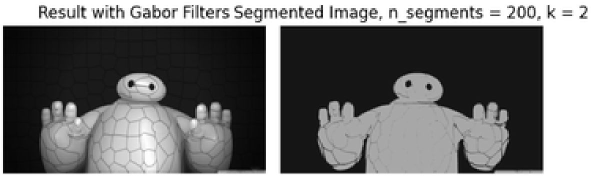


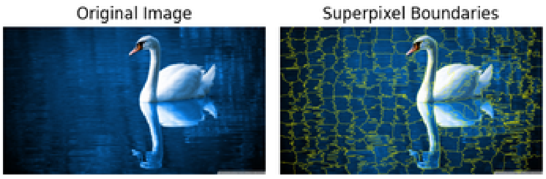


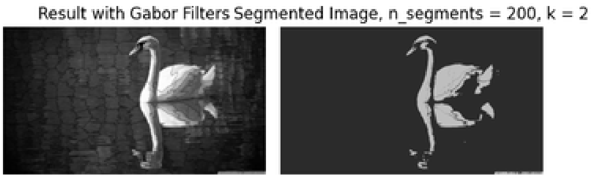
The results are way better than methods performed before. It even segments first images eyes, segments only horse (not bushes) for third image, segments all family very well in forth image and also our superheros segmentation is good enough to detect characters. Lets try with different parameters and finally compare methods among them

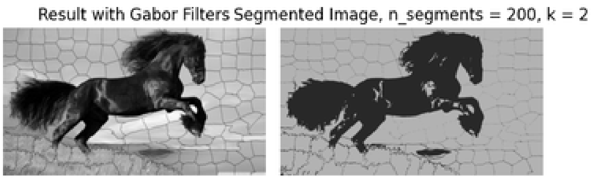


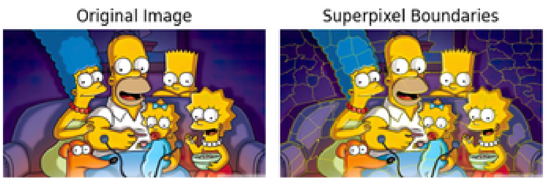


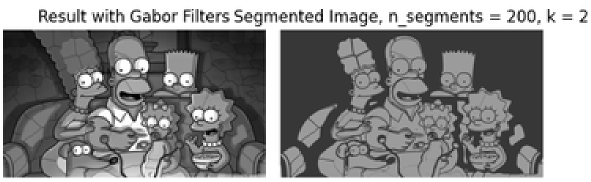


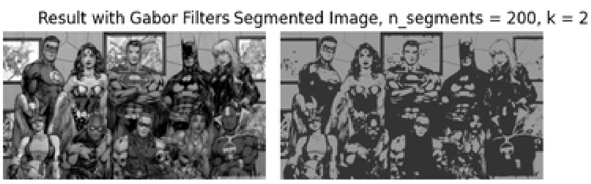




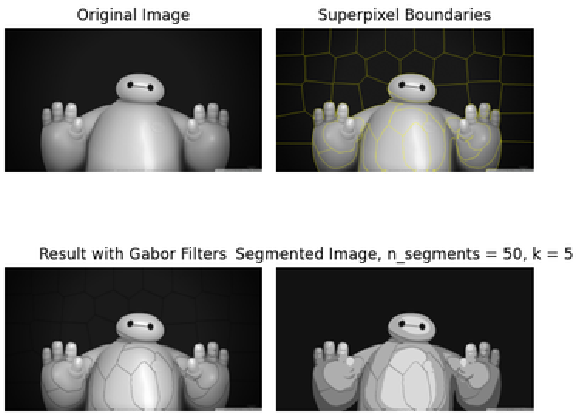


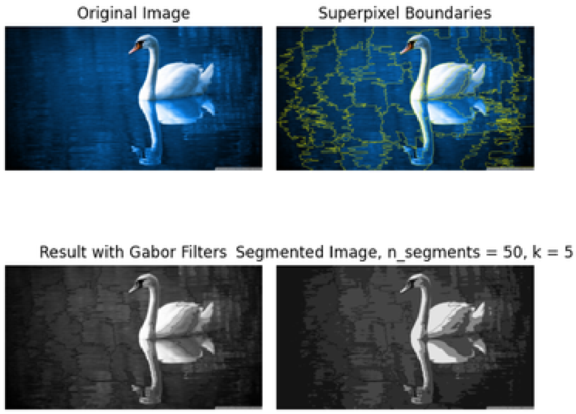


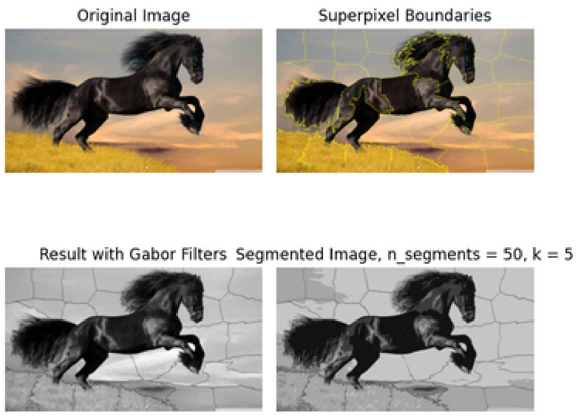


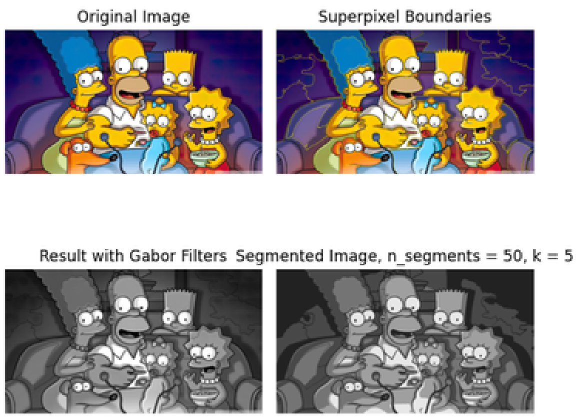


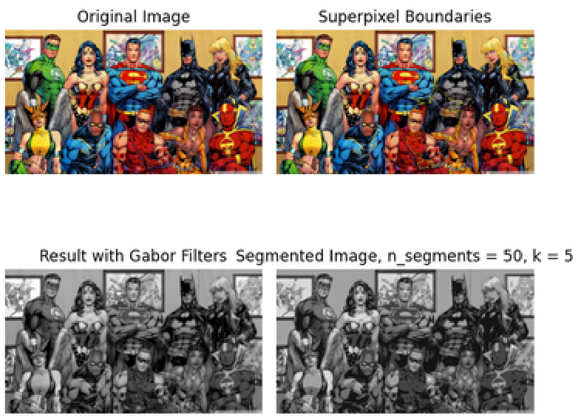
When increasing the number of segmentations from 50 to 200, there wasn't much change except for details. When the number of segmentations increases, also we can detect details better but it isn't such import thing. The disadvantages of increasing number of segmentation is we get more lines in final image and it shows our image worse. We should keep our number of segmentation 50







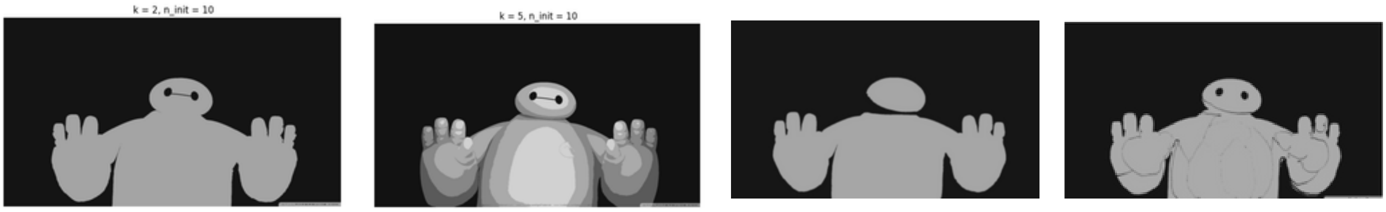




When incereasing number of clusters from 2 to 5, algorithms gets more detailed and captures even S letter on the superman. It gets more detailed segmentation but it is up to your purpose to use which number of clusters you wan to use. k = 2 is enough for first three images. For fair comparison, we will go with number of clusters = 2 and number of segmentation = 50.

Comparisons of Methods

I will share the results in the order of RGB, RGBXY, superpixel with mean, and superpixel with Gabor filter. (I couldn't convert image histograms to images, so I will not mention about that method)



For the black & white image, RGB color features are the best segmentation (It even sees the lines between the eyes but Gabor bank couldn't do that) and the fastest method amongst others. So if we want to make segmentation a one-character image that consists with black & white pixels, we can use just RGB color features and KMeans clustering algorithm

For the blue & white image, RGB color features and Gabor bank are the best segmentation and but we should choose RGB color features because it is way faster than Gabor bank.

If we only want to detect the horse, algorithms other than the RGB with spatial location algorithm perform quite well. However, if our goal is to also distinguish the boundary between grass and the sky, the only algorithm capable of doing this is the RGB with spatial locations algorithm

According to our goal, and although it may vary, if our aim is to distinguish multiple characters with the same colors in front of a single background, RGB color features and superpixels with the mean of Gabor filter bank algorithms perform this task quite effectively. The Mean of RGB color values on superpixels algorithm, on the other hand, is notably poor in this regard. This is because the algorithm initially segments the image into superpixels and then attempts to merge these superpixels. Given that the characters have the same colors, it exhibits a significantly unsuccessful performance in this aspect.

This image resembles the one above. We can once again observe how ineffective the superpixels with mean algorithm is in dealing with various characters of different colors and shapes in front of a diverse background. The RGB color features algorithm, on the other hand, has demonstrated a highly successful performance in extracting characters and shapes. The RGB with spatial locations algorithm has attempted to cluster objects with the closest colors together, especially focusing on similar clothing colors, but its success is limited, and its output might not be perceived as a proper clustering. In contrast, the superpixels with the mean of Gabor filter bank algorithm has excelled in segmentation, effectively distinguishing characters and tables.

Conclusion

Firstly, I must say that RGB color features and RGB with spatial locations algorithms exhibit very similar performances when k = 2. However, as the number of clusters increases, the RGB with spatial locations algorithm, incorporating spatial information, shows much better segmentation. But for comparison purposes, when k = 2, both are expected to yield similar results. Therefore, I chose k = 5 for the RGB with spatial locations algorithm.

The strength of using just RGB color features is that the method is effective in scenarios where objects have distinct color differences, making it well-suited for situations where color is a prominent feature for segmentation. However, it may struggle in cases where color alone is not sufficient for accurate segmentation, especially in scenes where objects have similar colors but different spatial locations.

The strength of using spatial locations is that the method is capable of grouping pixels with similar colors that are spatially close, which helps in cases where the spatial arrangement is crucial for accurate segmentation. However, it may face challenges in situations where objects have similar colors but are spatially separated. The effectiveness depends on the distribution and arrangement of colors in the image.

Comparing the superpixel methods, the "apply mean of RGB color values to superpixel" algorithm is directly related to the number of segmentations in the image. The algorithm divides the image into segments and assigns the average of their RGB values to the corresponding values. When applying KMeans clustering to this method, its effectiveness diminishes in images with high detail. Adjusting the segmentation or cluster numbers based on the image can enhance the algorithm's performance.

Looking at the "apply mean of gabor filter responses" algorithm, (firstly, I must mention that normally, this algorithm does not produce black and white transformations, but I chose a cluster number of 2, so I didn't want to implement this again on the image). This algorithm provides valid results for almost all types of images. I used approximately 16 different Gabor filters and kept the segmentation number at 50, yet it demonstrated quite successful performance. Adjusting parameters can lead to even better segmentation, but this algorithm, utilizing the average of various responses, is slow but performs well.

Result

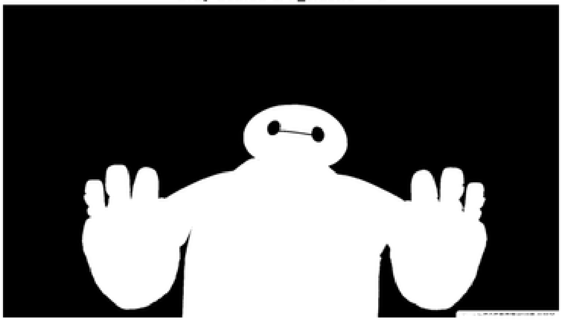
Superpixel-based segmentation methods offer various advantages and disadvantages compared to traditional pixel-based approaches. These methods summarize information by reducing the number of pixels in the image, resulting in faster processing and lower computational costs. Additionally, they create larger and visually meaningful regions by merging pixels with similar features, enhancing information integrity. However, they may be less precise in capturing detailed boundaries compared to pixel-based methods. Furthermore, superpixel algorithms often require parameter tuning and may encounter challenges with scale variability. Nevertheless, depending on the problem characteristics and requirements, superpixel-based methods are generally preferred for achieving more meaningful and coherent segmentation results.

References

- https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_segmentations.html
- https://docs.opencv.org/3.4/df/d6c/group__ximgproc__superpixel.html
- https://towardsdatascience.com/create-your-own-k-means-clustering-algorithm-in-python-d7d4c9077670
- https://pub.towardsai.net/k-means-clustering-from-scratch-4e357ac4716f
- https://scikit-image.org/docs/stable/auto_examples/features_detection/plot_gabor.html
- https://www.mathworks.com/help/images/ref/gabor.html
- https://ieeexplore.ieee.org/document/8321272
- https://www.analyticsvidhya.com/blog/2021/09/image-segmentation-algorithms-with-implementation-in-python/

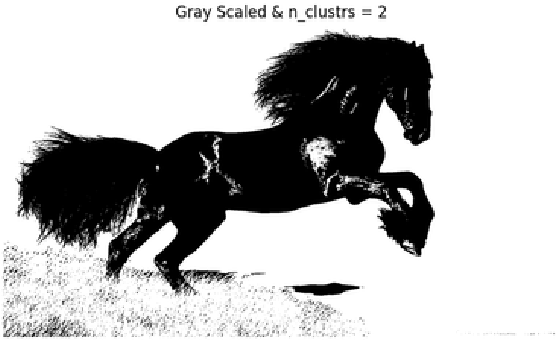
Additional Clustering Method

Gray Scaled & n_clustrs = 2

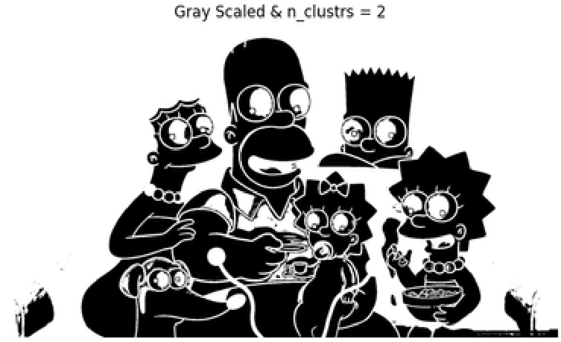


Gray Scaled & n_clustrs = 2

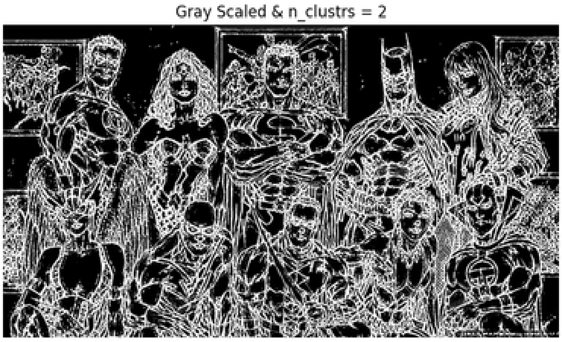
Gray Scaled & n_clustrs = 2



Gray Scaled & n_clustrs = 2



Gray Scaled & n_clustrs = 2



When creating Gabor filters, I wanted to experiment without applying KMeans clustering. Doing this, I noticed that even without clustering, it could be a segmentation algorithm that works well based on the image features. As seen in the examples next to it, it operates quite effectively, especially when there is low color diversity in the image.