Ashish Gupta

50134737

Homework2

---

e)

i) Image Textures generally consist of repeated elements, sometimes called textons. It is difficult to represent a texton accurately. We assume that textons are made of sub elements such as spots, bars, edges etc . These sub elements can be found using filters and then we can represent each point in an image as a vector of filter outputs at that point . Same approach has been applied in the implementation of the given problem. I identified various sub elements using the Leung-Malik bank of filters which consist of multi scale , multi oriented 48 filters.

After I represented each point in an image as a vector of filter outputs, I utilized the fact that, if textons repeat in an image, so does their subelements. I have to now find sets of texton sub elements that occur together. I cannot simply count because each pixel in a patch differs significantly in various ways. I used vector quantization to find sets of texton subelements that occur together. I replace a vector with cluster center closest to that vector. K-means clustering algorithm is based on this idea . We choose k random points to start with and then assign the points to a cluster based on the distance from k chosen random points. A new center is assigned as the mean of the previously found cluster and we continue the process until the cluster centers don't change much.

Each cluster at the end of k-means algorithm consists of the texton subelements that occur together. (forground and background).

The problem after clustering the elements together is to find the foreground and the background segment vector. I examined and found that the cluster assignments and the cluster id that occured most in the neighborhood of the borders corresponded to the background and rejected them as the background clusters. Mostly, the cluster ids that are present most in the center of the image correspond to the foreground segments. Also, we can look at a point at a point in the image and choose the cluster id assigned to it as the foreground segment vector.

Also, I have used random assignment of cluster centers at the start , so each time the program will run, the foreground and background segment vectors in the output will change.

This pixel based segmentation technique suffers from many limitations:

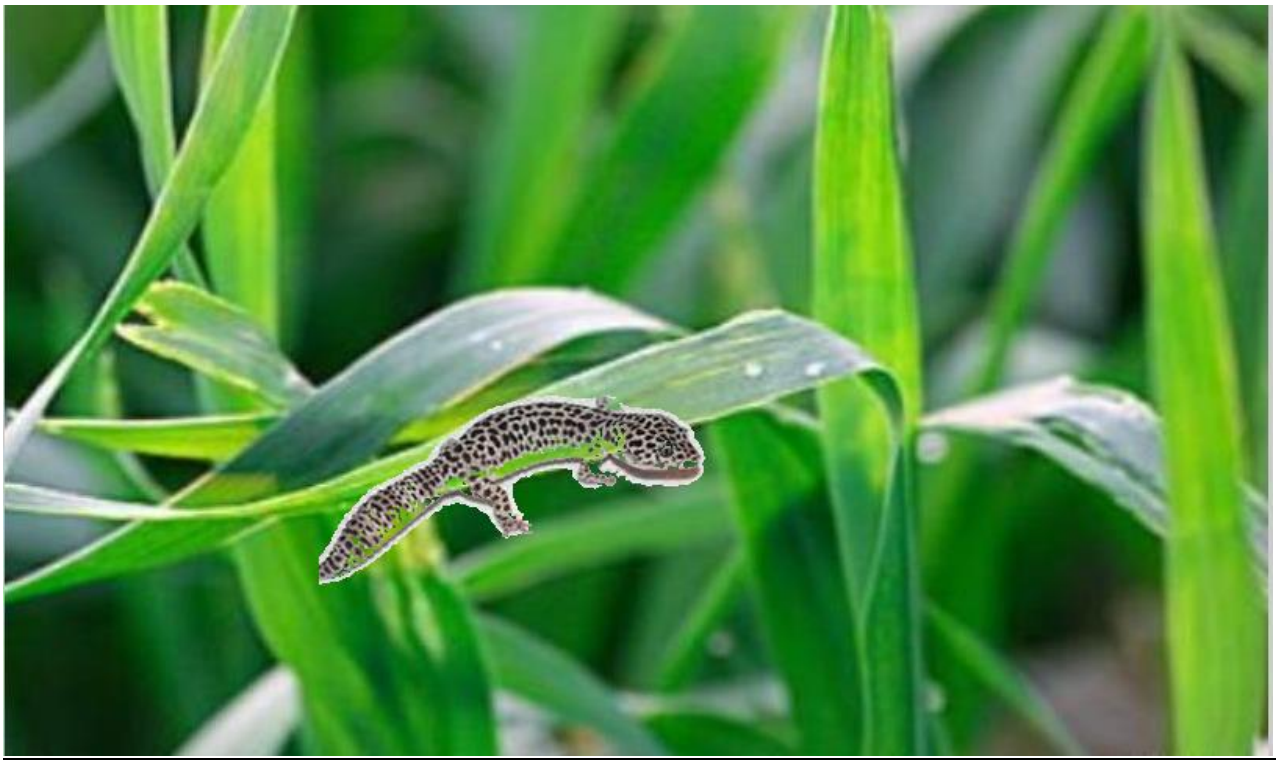
1- **The texture representation at a point must involve a summary of nearby filter outputs** , rather than just the filter outputs at that point.

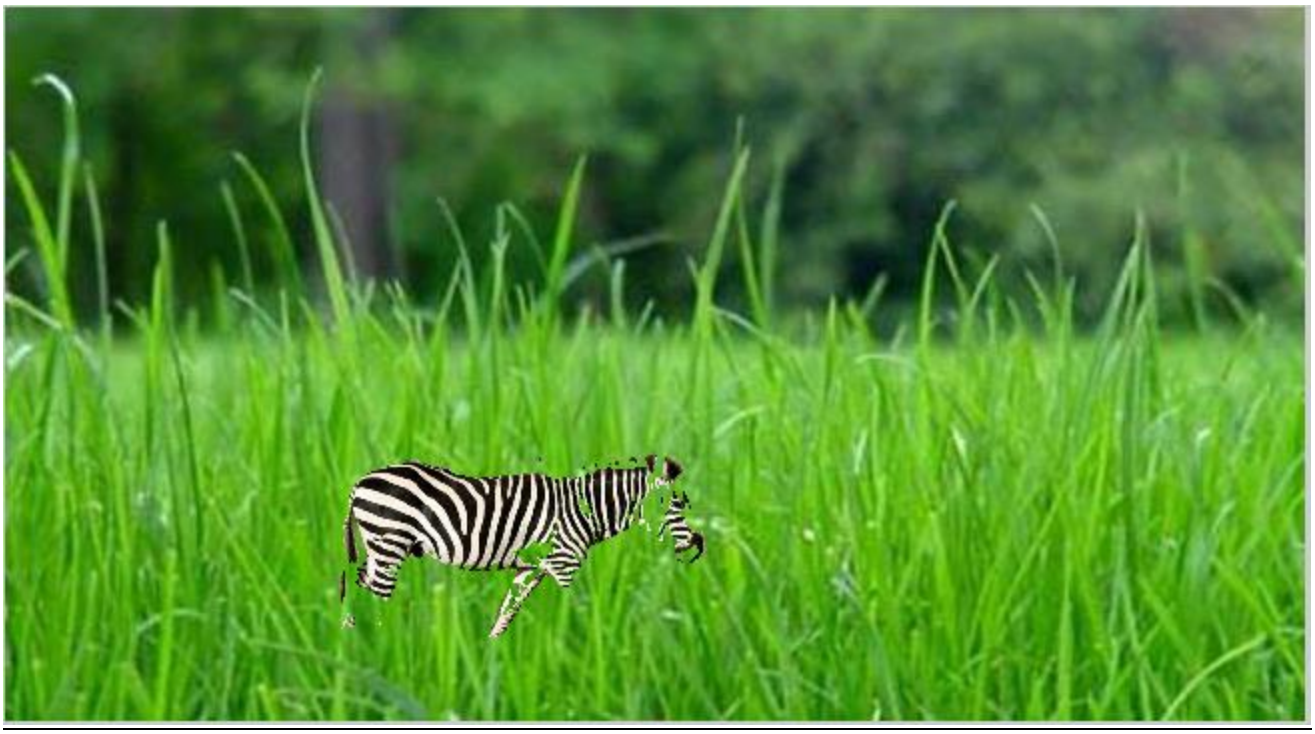
2- **This implementation cannot distinguish between lights spots on a dark background and dark spots on a light background** - We cannot just take the absolute value of the filter output maps. Performing half wave rectification and summarizing the neighborhood by computing a Gaussian weighted average will produce better results.

ii) I have **implemented my own version** of k-means clustering algorithm.

iii)Output Images

k- means clustering – k value of 3 has been used for all images.





iv) As mentioned before directly taking the absolute value of the filter output maps does not distinguish between light spots in the dark background and dark spots in the light background. Clustering of such elements get affected and they get clustered with a wrong cluster.

This can be improved by using more pixel features such as color . A neighborhood based segmentation can also help in reducing holes in the images . Instead of clustering pixels , we can cluster based on neighbourhood summarization in the image.

f)

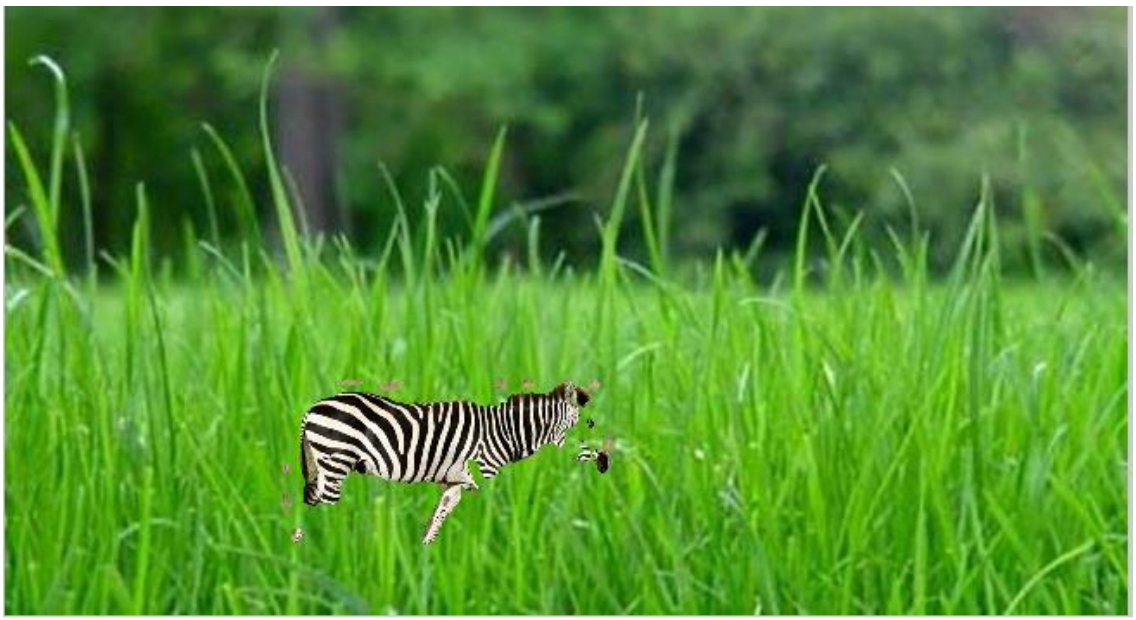
Segmentation by Half Wave Rectification and Summarization of neighbourhood

The requirements in the problem state to take the absolute value of the output filter maps. However taking the absolute values hamper the output as stated above.

I have tried to perform the half wave rectification on the filter outputs. After performing the half wave rectification of the filter output maps, I tried to summarize the neighborhood by taking a Gaussian weighted average by convolving with a Gaussian filter . This does help reduce the number of holes in the output . The output also depends on the size of kernel chosen. Taking a big kernel affects the output as some elements which do not belong to a cluster , get clustered together. This is because the representation does not change much as we move across the image because the neighborhoods overlap.

Performing the segmentation using half wave rectification on the dog image reduces the number of holes. K value of 3 has been used in all images for k-means algorithm.

Here is another example of segmentation by neighborhood summarization.



The code for Segmentation by Half wave Rectification and Summarization of neighborhood is present in **segmentImgHWR.m** and can be used in the same way as segmentImg .

I tried to augment the pixel features with the normalized RGB value at each pixel but that did not help in reducing the holes in the image. The code for augmenting (mentioned in a comment) is part of segmentImg.m.

I also tried the color based segmentation. It does produce significantly better results by reducing the holes in the cluster for the provided input images. However, if the foreground and the background have same color, color based segmentation will fail. The code for color based segmentation is present in colorBased.m and has been referenced from http://www.mathworks.com/

Output of color based clustering: