CSE 473: Homework Assignment 2

Author: Tejas Vyas
Person #: 5005-5126

(i) Description of the Implementation:

Basic Idea:

We're using K-Means clustering to sort the collections of data. The data in this case is the image of an animal which will be segmented from its original image and layered into a background image.

Initialization and Description:

I used a runfile to run the algorithm multiple times and generate 3 different superimposed images based on the image sets and cluster sizes. Please use the runfile to change K-means constant or images.
Every superimposed image takes in 2 images: One as foreground – The image on which the clustering is performed and the other is background – the image on which the clustered data is superimposed.
I'm using a segmentImg function to initiate segmentation on the foreground image using the K-Means algorithm from Matlab. (The given kmeans file did not work as expected on my MatLab).
Basically, I take in an image and a user-set cluster size (should be small (~[2,5])) in order to get a really nicely segmented image.

- I take the image in, convert it to grayscale (Since the 48 LM Bank of filters are in grayscale)
- Process the image for the K-Means algorithm to be run on it.
- Run the kmeans function
- Resize the resulting image based on the original image's dimensions.
- Use the transferImg function to transfer the segmented image to the background and return the image.

Inputs and Outputs:

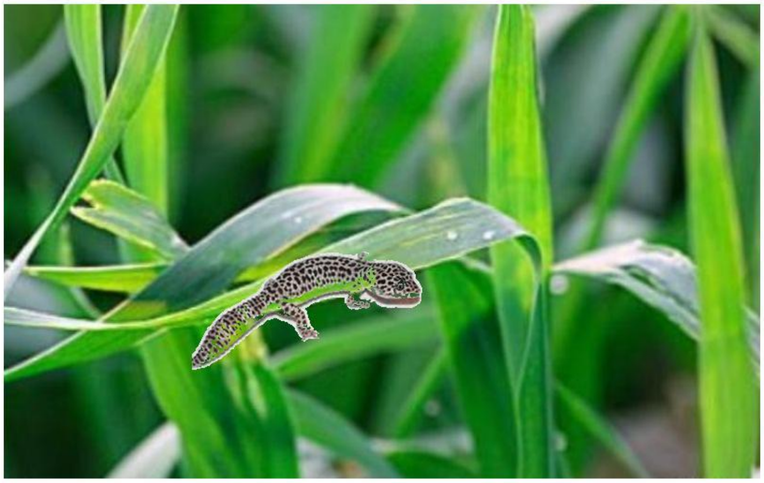
INPUTS

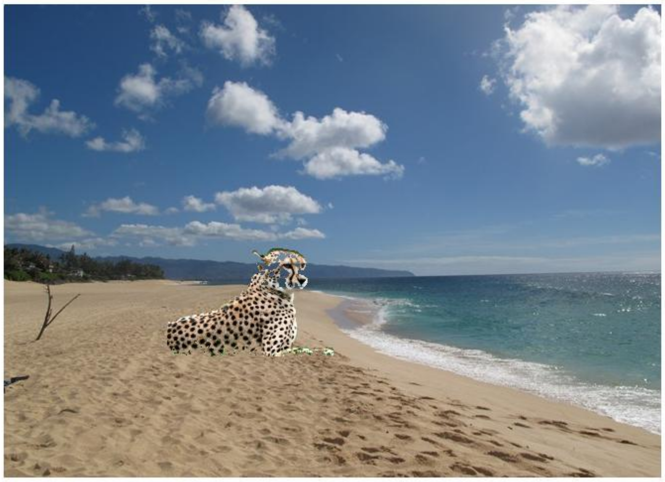
+

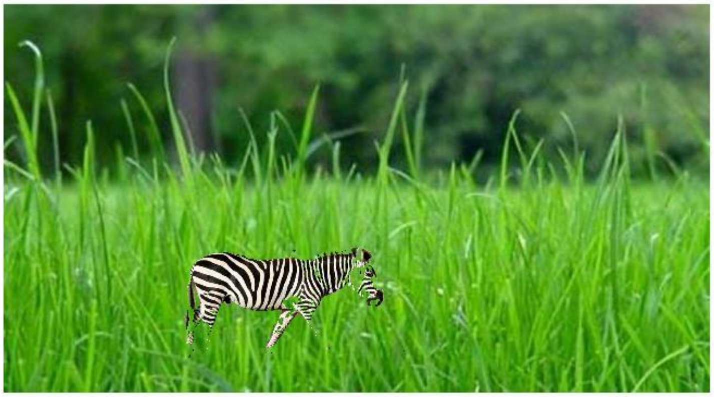
+

+

__OUTPUTS:__







The most interesting is the holes in the images: They are present because the centroid distance of the clusters is too similar to a different part of the image and the image pixel gets sent to a different cluster. This is a limitation in itself and may be removed if we can add a little bit of pattern and or color recognition and adding features to check spatial connectivity to it.

(ii) K-Means cluster algorithm: I didn't implement my own as the one given worked perfectly fine. The Kmeans function however, was taken from Matlab's statistics toolbox instead of the given one.

(iii) The outputs above show 3 different set of superimposed images for your reference. See "Outputs"

(iv) The reasons behind this are:

  a. Texture too similar to other parts of the image
  b. Spatial connectivity not correctly working on the part of the image
  c. Because the image is in grayscale, the differentiation we do based on color can't be performed, and the direct comparison in being made using the distance between the filters.
     i. Using a much larger set of filter maps could improve the segmentation quality
     ii. Using spatial connectivity and texture recognition to pinpoint the distance between the similar components of the image.