

CS 216 Homework 3

Zachary DeStefano, 15247592

Due Date: May 9, 2014 in class

Problem 1

k-Means Clustering results

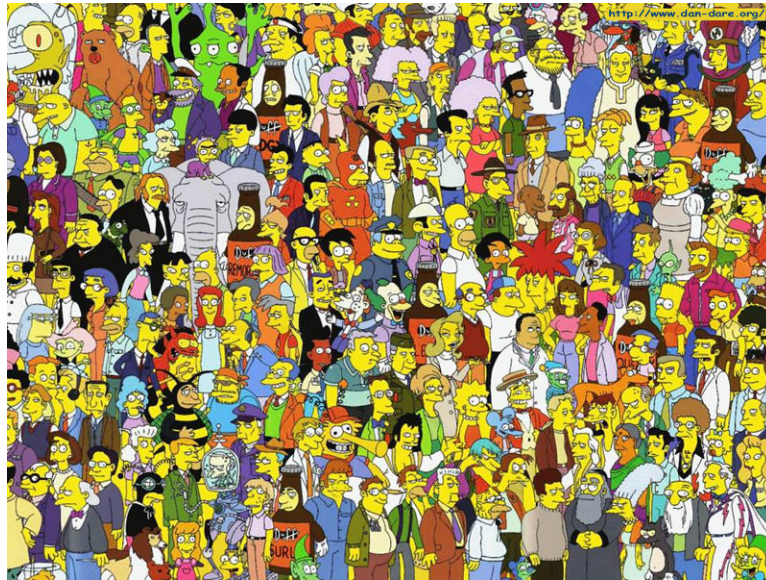


Figure 1: Original Image



Figure 2: The k-Means Image when $k = 2$

Figure 3: The k-Means Image when $k = 5$ Figure 4: The k-Means Image when $k = 10$

Result when skewing the Red Channel

If we multiply the red channel by 100, then the mean will tend more toward the red channel than the other channels. The result will be a red tint on the final result image. That is exactly what happened with the following images which are the same k-means images as above but the red channel was multiplied by 100 before k-Means was done.



Figure 5: The k-Means Image when $k = 2$ and a skewed red channel



Figure 6: The k-Means Image when $k = 5$ and a skewed red channel



Figure 7: The k-Means Image when $k = 10$ and a skewed red channel

Problem 2

Here is my original image:



Figure 8: The image of the zebra

8 filter images



Figure 9: The Horizontal Derivative image when $\sigma = 1$



Figure 10: The Horizontal Derivative image when $\sigma = 2$



Figure 11: The Horizontal Derivative image when $\sigma = 4$



Figure 12: The Vertical Derivative image when $\sigma = 1$



Figure 13: The Vertical Derivative image when $\sigma = 2$



Figure 14: The Vertical Derivative image when $\sigma = 4$

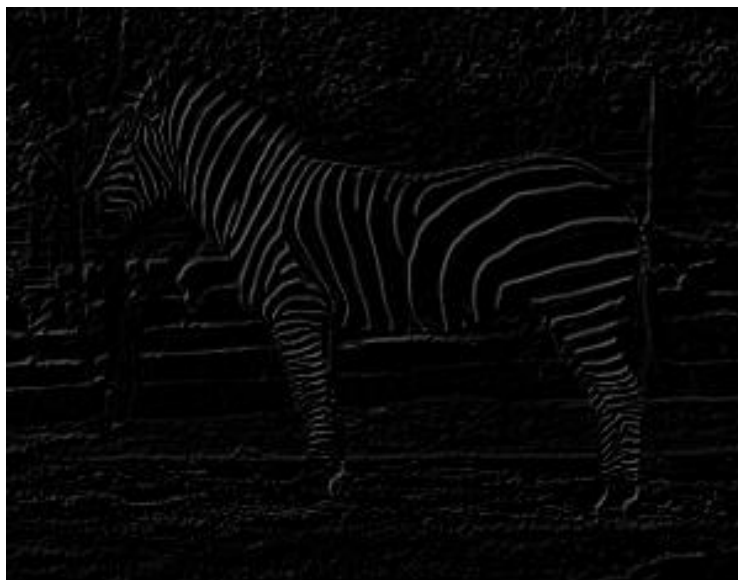


Figure 15: The Gaussian Difference when $\sigma_2 = 2$ and $\sigma_1 = 1$

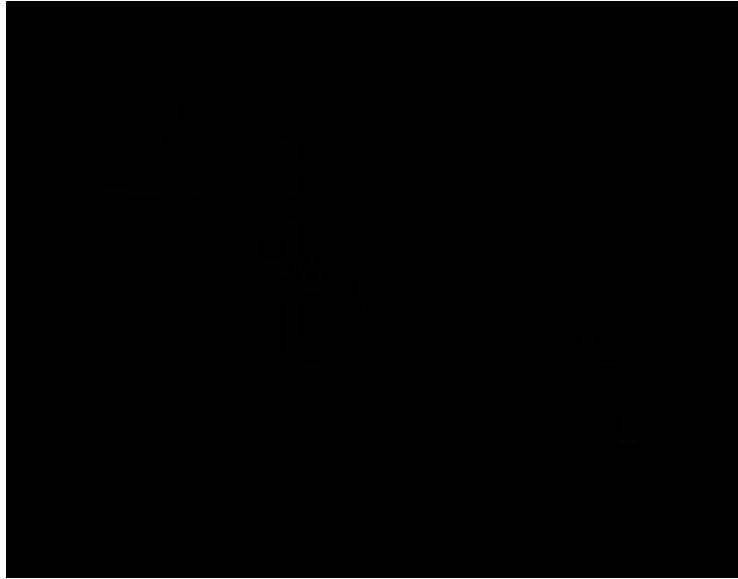


Figure 16: The Gaussian Difference when $\sigma_2 = 4$ and $\sigma_1 = 2$

8 filters in one image

Here are the 8 filters from above put together in one image. The first three images in the first column show the horizontal derivatives. The first three images in the second column show the vertical derivatives. In both cases, $\sigma = 1$ in the first row, $\sigma = 2$ in the second row, and $\sigma = 3$ in the third row. In the last row is the Gaussian difference images with $G_4 - G_2$ as the first image and $G_2 - G_1$ as the second image.



Matlab code

Here is the parent script that calls the function:

```

imname = 'zebra_small.jpg';
imageData = im2double(rgb2gray(imread(imname)));
%
%h1 means horizontal derivative, sigma 1
%h2, h4, v1, v2, v4 follow the same thing
%g42 is gaussian 4-2, g21 is similar
[h1,v1,h2,v2,h4,v4,g42,g21] = get8FilterImages(imageData,imname);
figure
subplot(4,2,1)
imshow(h1)
subplot(4,2,2)
imshow(v1)
subplot(4,2,3)
imshow(h2)
subplot(4,2,4)
imshow(v2)
subplot(4,2,5)
imshow(h4)
subplot(4,2,6)
imshow(v4)
subplot(4,2,7)
imshow(g42)
subplot(4,2,8)
imshow(g21)

```

Here is the main function that gets the information:

```

function [ horizDeriv_Sigma1_imageData,vertDeriv_Sigma1_imageData,...
    horizDeriv_Sigma2_imageData,vertDeriv_Sigma2_imageData,...
    horizDeriv_Sigma4_imageData,vertDeriv_Sigma4_imageData,...
    gaussDiff_4_2_imageData,gaussDiff_2_1_imageData] = get8FilterImages( imageData,
%GET8FILTERIMAGES Summary of this function goes here
% Detailed explanation goes here

```

```
[horizDeriv_Sigma1_imageData,vertDeriv_Sigma1_imageData] =...
    computeDerivImages(imageData,1,imname);
[horizDeriv_Sigma2_imageData,vertDeriv_Sigma2_imageData] =...
    computeDerivImages(imageData,2,imname);
[horizDeriv_Sigma4_imageData,vertDeriv_Sigma4_imageData] =...
    computeDerivImages(imageData,4,imname);
gaussDiff_4_2_imageData = computeGaussDiff(imageData,1,2,imname);
gaussDiff_2_1_imageData = computeGaussDiff(imageData,2,4,imname);
end
```

This is the helper function that gets the gaussian derivatives:

```
function [horizDerivImage,vertDerivImage] = computeDerivImages( imageData,sigma,imname )
%COMPUTEDERIVIMAGES Summary of this function goes here
%   Detailed explanation goes here

gaussFilt = fspecial('gaussian',sigma);
filteredImageData = conv2(imageData,gaussFilt,'same');
horizDerivFilter = [1 -1];
horizDerivImage = conv2(filteredImageData,horizDerivFilter,'same');
vertDerivFilter = transpose(horizDerivFilter);
vertDerivImage = conv2(filteredImageData,vertDerivFilter,'same');
horizDerivImageName = strcat('sigma_',num2str(sigma),'_horizDeriv_',imname);
vertDerivImageName = strcat('sigma_',num2str(sigma),'_vertDeriv_',imname);
imwrite(horizDerivImage,horizDerivImageName,'JPEG');
imwrite(vertDerivImage,vertDerivImageName,'JPEG');

end
```

This is the helper function that computes the gaussian differences

```
function [filterDiffImageData] = computeGaussDiff( imageData,sigma1,sigma2,imname )
%COMPUTEDERIVIMAGES Summary of this function goes here
%   Detailed explanation goes here

gaussFilt1 = fspecial('gaussian',sigma1);
gaussFilt2 = fspecial('gaussian',sigma2);
filteredImageData1 = conv2(imageData,gaussFilt1,'same');
filteredImageData2 = conv2(imageData,gaussFilt2,'same');
filterDiffImageData = filteredImageData2-filteredImageData1;
```

```
filterDiffImageName = strcat('sigma2_', num2str(sigma2), '_sigma1_', ...  
    num2str(sigma1), '_gaussDiff_', imname);  
  
imwrite(filterDiffImageData, filterDiffImageName, 'JPEG');  
  
end
```


Problem 3

These are the image patches that were filtered and then analyzed. The left one is the neck of the zebra. The middle one is the picture of the tree leaves above the zebra's back. The right one is the grass in front of the zebra. The radius of the patch was scaled and it was scaled in such a way so that the patch would capture that object and only that object.

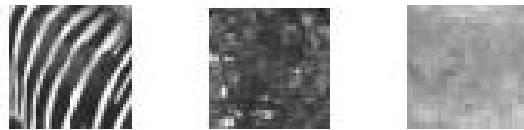


Figure 18: Image Patches that were analyzed

Here is the visualization for each of the mean absolute response vectors.

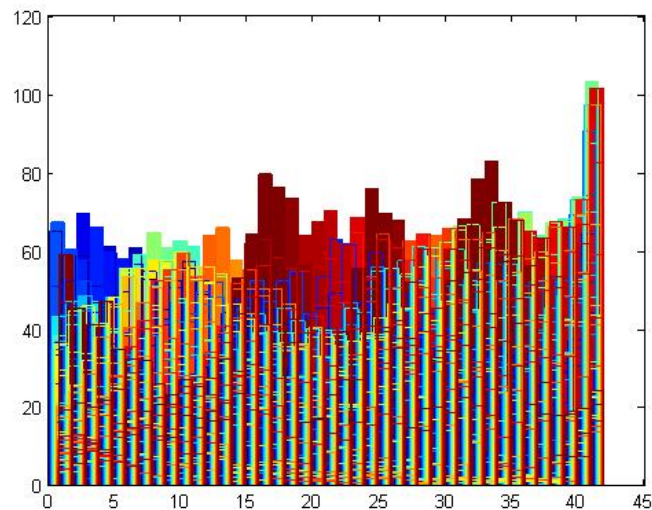


Figure 19: Bar graph visualization for the neck image patch

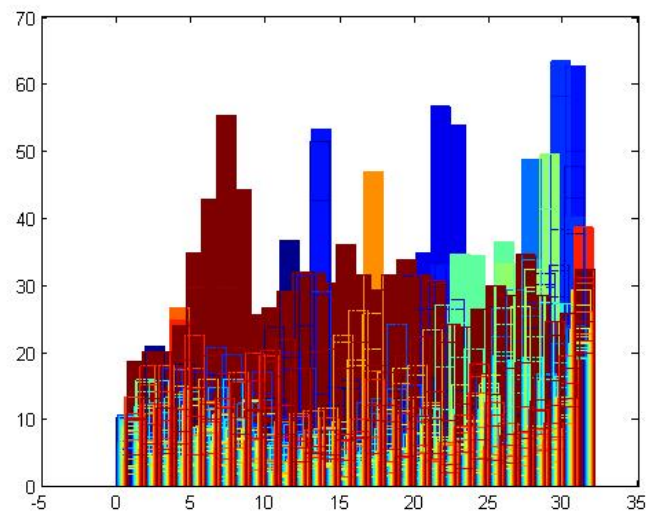


Figure 20: Bar graph visualization for the leaves image patch

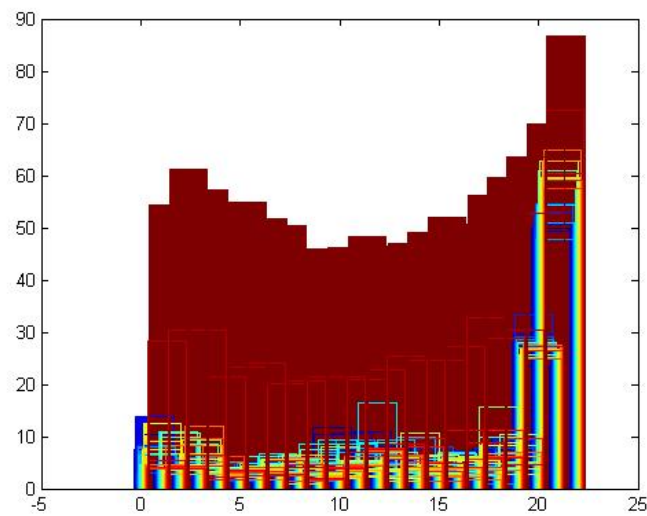


Figure 21: Bar graph visualization for the grass image patch

Problem 4

Here is the color map created by applying the 8 filters to the zebra picture above and then running k-means on the 8-dimensional data created by putting all the images into one large array.

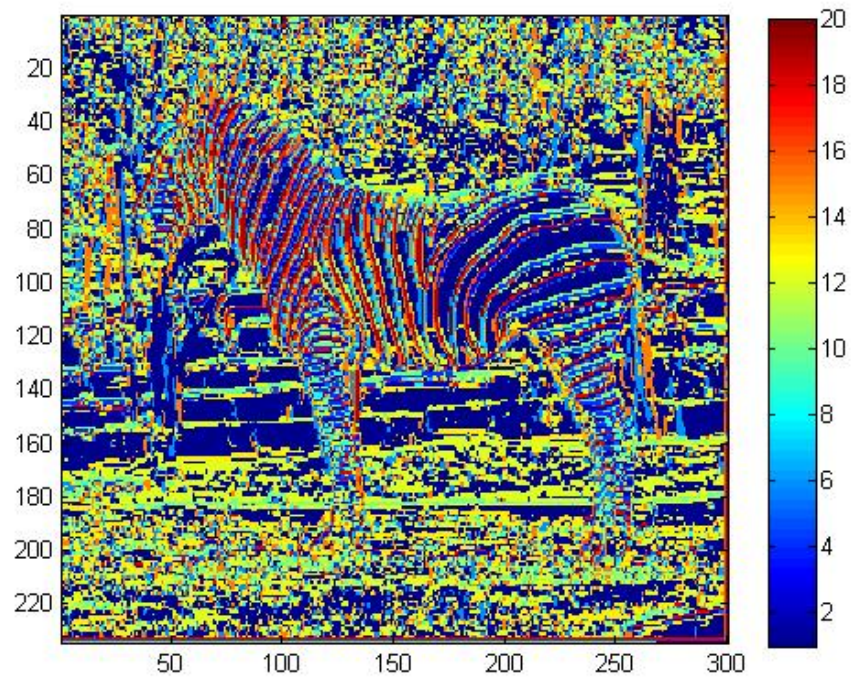


Figure 22: Imagesc visualization of the cluster center labels for the zebra image

Problem 5

Problem 6

In the Department of Ophthalmology at UC Irvine, they have pictures of collagen fibers in the eye and they want to use these pictures as a basis for a 3D reconstruction of the fibers. The first step in the 3D reconstruction will be segmenting the images and putting them together. In the images, the fibers look like lines stacked up on top of each other. These lines can merge together as well as split apart into two lines.

Here is a rough outline of the steps of the project:

1. Segment the images to extract out the line data.
2. Use the line information to line up images side by side
3. Get long images that are formed by a series of images put side by side