Manvi Goel
2019472

Computer Vision
Assignment 2

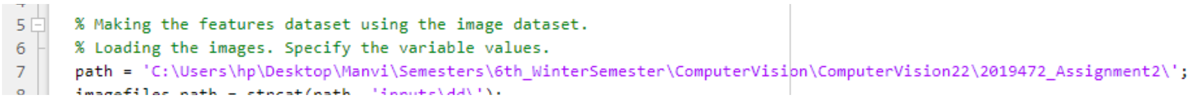
Language: Matlab

Question 1

Filename: m_2019472_q1.m
Write to execute: m_2019472_q1()
Input: None
Sample Input: m_2019472_q1()
Output: A CSV file with the separation measure and concentration measure of all the images in the dataset.

Assumptions.

1. The DL-based and Non-DL-based saliency maps are given in the folders Q1_DL and Q1_NonDL respectively.
2. The number of bins taken for the separation measure is 1000.
3. The path needs to be fixed as required.



```
% Making the features dataset using the image dataset.
% Loading the images. Specify the variable values.
path = 'C:\Users\hp\Desktop\Manvi\Semesters\6th_WinterSemester\ComputerVision\ComputerVision22\2019472_Assignment2\';
imagefiles_path = strcat(path, 'inputs\dd\');
```

Working.

1. Read saliency map and use otsu threshold.
2. Calculate the number of pixels in the two separated regions and select the region with a higher number of pixels as foreground and the other as background.
3. Use the foreground and background masks to get the thresholded grayscale images.
4. Calculate the mean and standard deviation for both the foreground and background.
5. Calculate the Gaussian distribution for the z.
6. Calculate the value of z* and integrate to find the area of intersection
7. Use the area to calculate the separation measure.
8. Use *bwconncomp* to calculate the connected components.
9. Sum the total area of connected components.
10. Use it to calculate the contribution of the largest component and calculate the concentration measure.
11. Print and display the results.

References.

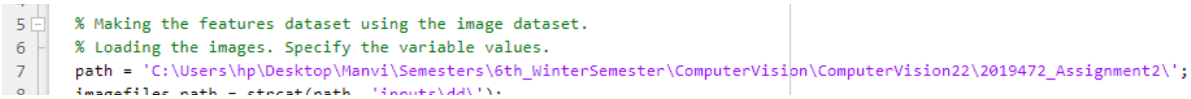
1. Saliency maps.
    - DL based. PoolNet-ResNet50 w/o edge model
    - Non-DL-based. Cv2's *StaticSaliencySpectralResidual_create* method.
    - The maps are stored in the input folder.

Question 2

Filename: m_2019472_q2.m
Write to execute: m_2019472_q2(*k*)
Input: An integer value k (less than 50)
Sample Input: m_2019472_q2(5)
Output: k sub-folders in the output directory

Assumptions.

1. k is less than 50 (size of dataset).
2. The number of neighbors for patches is 8. (hardcoded)
3. The SPP sizes are [16 4 1] (changeable)
4. The number of bins is 10 (changeable).
5. The output directory *Output/2019472_Q2* exists.
6. The images are padded with 0s for calculating the patches for corner pixels.
7. The path needs to be fixed as required.



```
% Making the features dataset using the image dataset.
% Loading the images. Specify the variable values.
path = 'C:\Users\hp\Desktop\Manvi\Semesters\6th_WinterSemester\ComputerVision\ComputerVision22\2019472_Assignment2\';
imagefiles_path = strcat(path, 'inputs\dd\');
```

Working.

1. Read an image and resize the image to 256x256.
2. Calculate the LBP features using 8 neighbors.
3. Calculate the histogram for the specified patch sizes using *imhist.*
4. Concatenate the features to give a spatial pyramid pooled feature vector.
5. Apply *fcm*.
6. Distribute the images to the closest centroid.
7. Save the 50 images into k subfolders.

References.

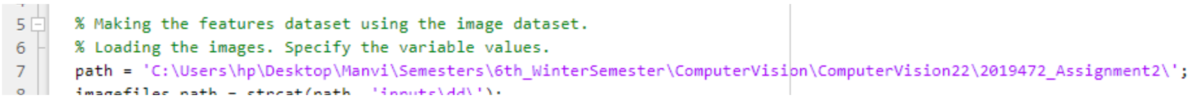
1. LBP adapted from a demo answer.
2. Fuzzy K means using the Fuzzy Logic Toolbox.
3. Dividing the image in patches is done using a demo answer.

Question 3

Filename: m_2019472_q3.m
Write to execute: m_2019472_q3(*Image*, *k*)
Input: An image for search and an integer value k (less than 50)
Sample Input: m_2019472_q3(“15_19_s.jpg”, 10)
Output: A feature vector of length k.

<u>Assumptions.</u>

1. k is less than 50 (size of dataset).
2. The image is resized to 128x64.
3. The number of patches is 16. (can be changed to any even power of 2 = (4, 16, 64, 256) less than 128*64)
4. The number of bins for HOG features is 9.
5. The path needs to be fixed as required.



```
% Making the features dataset using the image dataset.
% Loading the images. Specify the variable values.
path = 'C:\Users\hp\Desktop\Manvi\Semesters\6th_WinterSemester\ComputerVision\ComputerVision22\2019472_Assignment2\';
imagefiles_path = strcat(path, 'inputs\dd\');
```

<u>Working.</u>

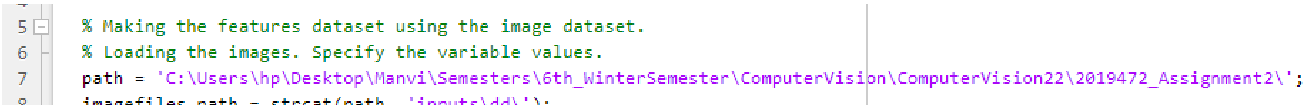
1. Read an image and determine the HOG features for (16) patches using the *extractHOGFeatures* function of Matlab.
2. Using the *bagOfFeatures* function of Matlab extract the k prominent features.
3. *Encode* the image using the model
4. Display the feature vectors. (both unnormalized and L2 normalized)

Question 4

Filename: m_2019472_q4.m
Write to execute: m_2019472_q4(*Search Image*, *k*)
Input: An image for search and an integer value k (less than 50)
Sample Input: m_2019472_q4("15_19_s.jpg", 10)
Output: An array with k indexes, Displayed output of k image (slideshow), and k saved images in the output directory

Assumptions.

4. k is less than 50 (size of dataset).
5. The number of neighbors for patches is 8. (hardcoded)
6. The number of corners selected using the Shi - Tomasi method is 50. (changeable)
7. The size of an individual feature vector from a patch is 256. (hardcoded).
8. The output directory *Output/2019472_Q4* exists.
9. The images are padded with 0s for calculating the patches for corner pixels.
10. The path needs to be fixed as required.



```
% Making the features dataset using the image dataset.
% Loading the images. Specify the variable values.
path = 'C:\Users\hp\Desktop\Manvi\Semesters\6th_WinterSemester\ComputerVision\ComputerVision22\2019472_Assignment2\';
imagefiles_path = strcat(path, 'inputs\dd\');
```

Working.

12. Read an image and calculate the corners using the *detectMinEigenFeatures* function of Matlab.
13. Calculate the LBP features for the top (50) corners patch of 8 neighbors. Represent the calculated values in form of *imhist.*
14. Horizontally concatenate the (50) features to give the final feature vector.
15. Repeat the process for all the images in the dataset and the search image.
16. Find the k nearest neighbors using the *knnsearch* method.
17. Show and save the k images.

References.

1. LBP adapted from a demo answer.