STCV ASSIGNMENT 2 REPORT

Question 1: Implementation of Harris Corner detection:

The pseudo code is:

1. Windows of size 2 * 2 are taken with a stride of 1.
2. For each of the windows a matrix M is prepared. The expression for obtaining matrix M is:

$$M = \sum w(x,y) \begin{bmatrix} I_x{}^2 & I_x I_y \\ I_x I_y & I_y{}^2 \end{bmatrix}$$

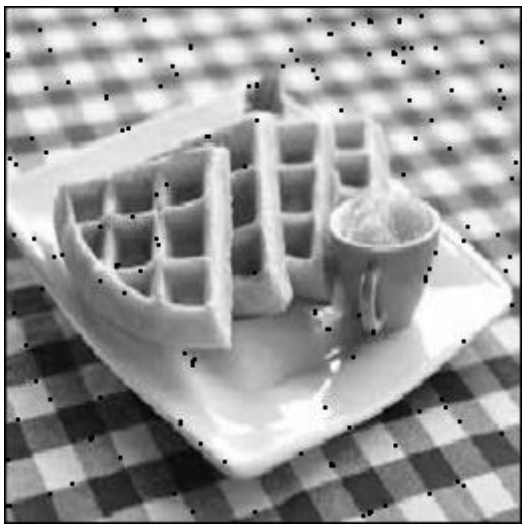
   That is for all points (x, y) belong to the window w, the X-gradient and the Y-gradient is calculated. I_x is the gradient in X-direction and I_y is the gradient in Y-direction. The matrix is obtained by summing up the respective values of all points belonging to the window.
3. Now for each of the windows a score R is calculated as:

$$R = \det M - k(trace\, M)^2$$

   Where M is the matrix obtained in the previous step.
4. Now the list of scores of all the windows is sorted in descending order. The top 'c' windows are now selected as the corners. Here, 'c' is the number of corners that need to be selected.

The input image I have taken is:

Now on selecting the top 150 corners based on R score of the window, the corners detected are shown with black dots in the below image.



Question 2: Implementation of Shi-Tomasi corner detection algorithm:

The pseudo code is:

1. Windows of size 2 * 2 are taken in the image with a stride of 1.
2. For each of the windows a matrix M is prepared. The expression for obtaining matrix M is:

$$M = \sum w(x,y) \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_x I_y & {I_y}^2 \end{bmatrix}$$

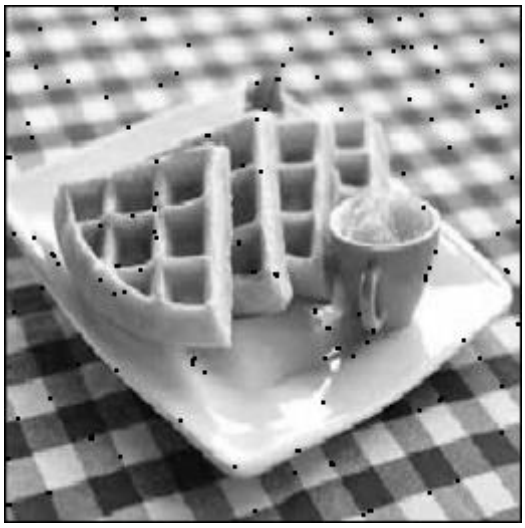
   That is for all points (x, y) belong to the window w, the X-gradient and the Y-gradient is calculated. I_x is the gradient in X-direction and I_y is the gradient in Y-direction. The matrix is obtained by summing up the respective values of all points belonging to the window.
3. Now for each window the eigen values λ_1 and λ_2 are calculated and the score R is obtained as:

$$R = min(\lambda_1, \lambda_2)$$

5. Now the list of scores of all the windows is sorted in descending order. The top ‘c’ windows are now selected as the corners. Here, ‘c’ is the number of corners that need to be selected.

The input image I have taken is:

Now on selecting the top 150 corners based on R score of the window, the corners detected are shown with black dots in the below image.



Question 3: Implementation of SIFT key point detector.

The pseudo code is:

1. The five scales for the image are obtained by applying Gaussian blur by varying the value of sigma. To get the value of sigma for the current scale, the sigma value of previous scale is multiplied by a constant factor.
2. Now the 5 scales are obtained.
3. For these 5 scales, the 4 difference of gaussian are obtained by finding the difference between 2 consecutive scales.

4. Using the difference of gaussian, now we need to obtain the key points. The key points are the points which are a maxima or a minima in their neighborhood.
5. We have got the 4 difference of Gaussian. For each pixel in the 2nd and 3rd DoG we see whether the pixel is a maxima or a minima by checking the value of its 26 neighborhood pixels.
6. The 26 neighborhood points are obtained by seeing the 8 neighboring points in its own DoG and 9 points each in the DoG above and the DoG below.
7. Now out of these obtained key points, we get rid of the low contrast points. This is done by selecting the applying Harris corner detector for the key points obtained.
8. Now the best key points obtained after applying corner detection are selected as the final obtained key points.

The original image I have taken is:

The key points obtained using the algorithm are shown with red dots in the below image:

Question 4: Implementation of mean-shift algorithm:

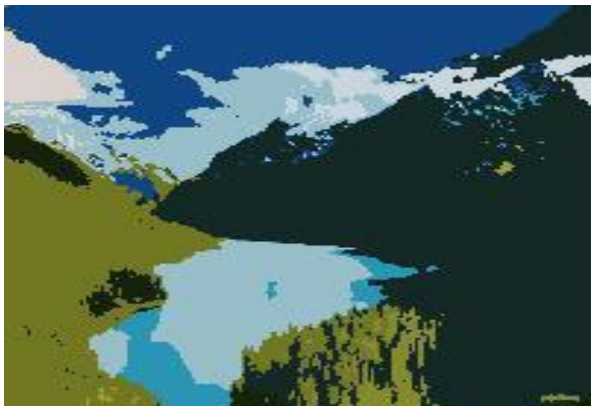
The pseudo code is:

1. The vectors of 5-dimensions are created for all pixels of the image. Vectors are [R, G, B, x, y] where x and y are the pixel co-ordinates.
2. The data is normalized by diving each value by the maximum value in the respective dimension.
3. A random point is picked from all the pixels.
4. All the points that are within a distance of bandwidth from the initial point are selected and their mean is calculated.
5. Now we shift to the new mean that has been calculated.
6. We keep on finding the mean and shift to the new mean until the mean converges to a value.
7. All the datapoints within a distance equal to bandwidth from the converged mean are given the same label as the mean i.e. there RGB values are assigned same as the RGB value of the mean.
8. The steps 3-8 are repeated until almost all datapoints have been labelled. The iteration is quit if some of the datapoints are still not getting labelled.

9. The datapoints that are not labelled are assigned the label of the last mean-shift process in which those datapoints took part.
10. If some datapoints are still unlabeled then a new label is assigned to these datapoints.

The image used for this implementation is:

The segmentation obtained using mean-shift algorithm is:



Question 5: Implementation of SLIC super pixel:

The pseudo code is:

1. Here also 5-dimensional vectors [R, G, B, x, y] are prepared for all the datapoints in the image are prepared.
2. The x and y dimension values are normalized by diving them by the maximum value in the respective dimension.
3. The value of k i.e. the number of super pixels that need to be obtained is decided. Here I have taken the value of k as 125.
4. The image has been resized to 256 * 256.
5. Now, N is the total numbers of pixels in the image.

The approximate size of each super pixel is:
N / k.

6. Now roughly equal sized super pixels are taken initially with cluster centers as uniform distance of S from each other where:
   S = sqrt(N) / k.
7. Now the initial points chosen should not be an edge. So, in the 3 * 3 neighborhood of the chosen pixel we find the lowest gradient position and instead chose that point.
   The gradient is calculated using the expression:

$$G(x,y) = \|\mathbf{I}(x+1,y) - \mathbf{I}(x-1,y)\|^2 + \|\mathbf{I}(x,y+1) - \mathbf{I}(x,y-1)\|^2$$

   Where ||.|| is the L_2 norm value.
8. Now for each cluster center assign the best matching pixels form the 2S * 2S neighborhood around the cluster center according to the distance measure given below:

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$
$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$
$$D_s = d_{lab} + \frac{m}{S} d_{xy} \ ,$$

   D_s is the distance measure used; value of m denotes how much importance needs to be given to the spatial distance. Here, I have taken m as equal to 10.
9. For each of the clusters the new cluster centers are calculated by finding the mean of the vectors of points assigned to the cluster.
10. The residual error for each of the clusters is found using the L1-distnace between the new center and the old center of the cluster. The process is repeated until for all the clusters the residual error is less than a threshold value.
11. Now for each of the clusters after convergence the points in the cluster are assigned the RGB value equal to the RGB value of the cluster center.

Here, I have taken the below image as input:

The boundaries formed for the different super pixels are shown with green color in the below image:

The image segmentation obtained is:

A much better segmentation can be obtained by tuning the parameters of the algorithm.