CO 543

Jayawardena H.D.N.S.

Lab 05 Report

E/19/167

1. Read the image texture.tif. Display the image. How many textures are there in the image? Describe them.

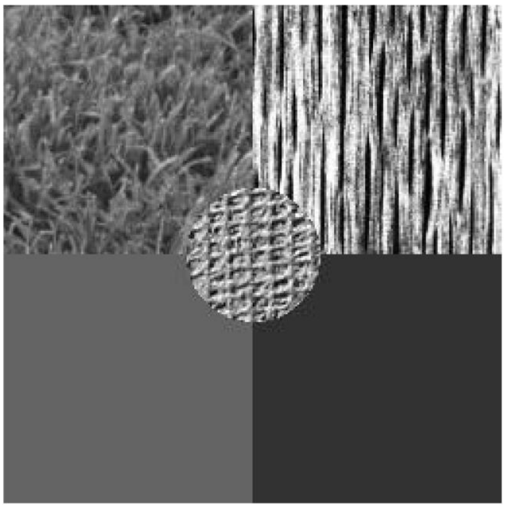
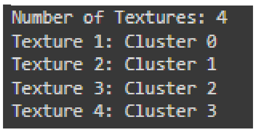
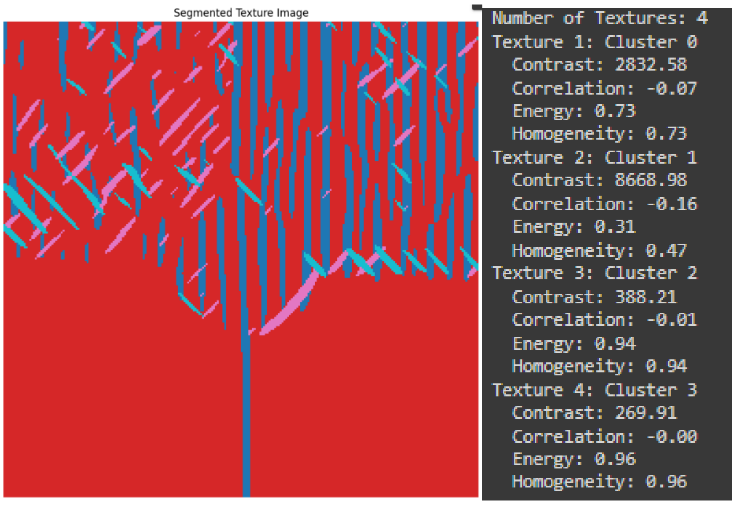


Fig 01: texture.tif

There were 4 textures,



```
Number of Textures: 4
Texture 1: Cluster 0
Texture 2: Cluster 1
Texture 3: Cluster 2
Texture 4: Cluster 3
```



```
Number of Textures: 4
Texture 1: Cluster 0
  Contrast: 2832.58
  Correlation: -0.07
  Energy: 0.73
  Homogeneity: 0.73
Texture 2: Cluster 1
  Contrast: 8668.98
  Correlation: -0.16
  Energy: 0.31
  Homogeneity: 0.47
Texture 3: Cluster 2
  Contrast: 388.21
  Correlation: -0.01
  Energy: 0.94
  Homogeneity: 0.94
Texture 4: Cluster 3
  Contrast: 269.91
  Correlation: -0.00
  Energy: 0.96
  Homogeneity: 0.96
```

Fig 02: Segmented texture and texture description

- **Top Left Texture**: This appears to be a rough or grassy texture.
- **Top Right Texture**: This has a linear, wood-like grain pattern.
- **Bottom Left Texture**: This is a smooth, solid color with no apparent texture.
- **Bottom Right Texture**: This is another smooth, solid color with no apparent texture, different from the bottom left in shade.
- **Center Circular Texture**: This has a patterned, woven appearance.

2. Select several features and calculate them on blocks of size of 12 × 12 using Gabor filter. Display the calculated features and estimate which ones can be used to segment given structure. For the selected images apply the K-means method and comment on the results.

   Input Image

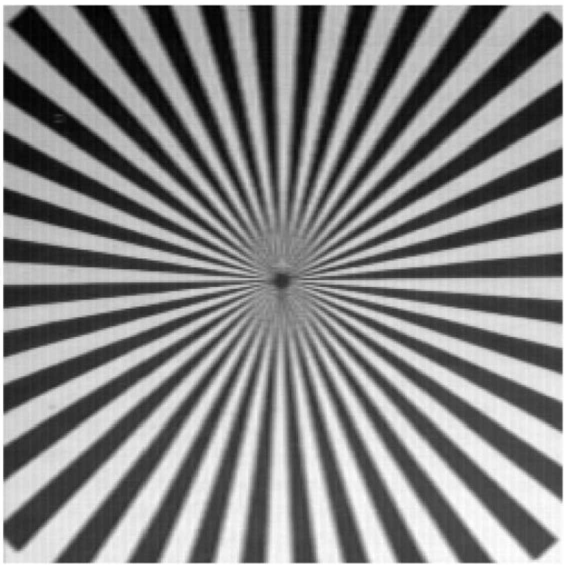
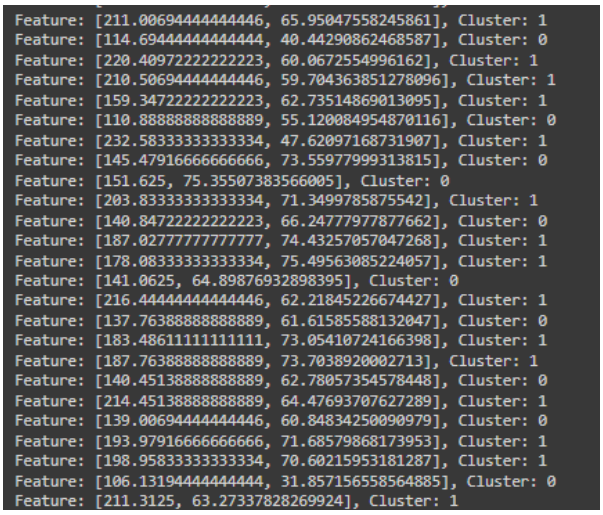


Fig. 03: Input image(texture.tif)

Steps:

- Define and apply Gabor filters to the image
- Apply filters to the image
- Split the image int 12x12 blocks and calculate features. Here as features mean intensity value and standard deviation is selected.
- Apply K-means clustering
- Print features and K-means results for each block
- Display the clusters here I have used 2 clusters.

Output



```
Feature: [211.00694444444446, 65.95047558245861], Cluster: 1
Feature: [114.69444444444444, 40.44290862468587], Cluster: 0
Feature: [220.40972222222223, 60.0672554996162], Cluster: 1
Feature: [210.50694444444446, 59.704363851278096], Cluster: 1
Feature: [159.34722222222223, 62.73514869013095], Cluster: 1
Feature: [110.88888888888889, 55.120084954870116], Cluster: 0
Feature: [232.58333333333334, 47.62097168731907], Cluster: 1
Feature: [145.47916666666666, 73.55977999313815], Cluster: 0
Feature: [151.625, 75.35507383566005], Cluster: 0
Feature: [203.83333333333334, 71.3499785875542], Cluster: 1
Feature: [140.84722222222223, 66.24777977877662], Cluster: 0
Feature: [187.02777777777777, 74.43257057047268], Cluster: 1
Feature: [178.08333333333334, 75.49563085224057], Cluster: 1
Feature: [141.0625, 64.89876932898395], Cluster: 0
Feature: [216.44444444444446, 62.21845226674427], Cluster: 1
Feature: [137.76388888888889, 61.61585588132047], Cluster: 0
Feature: [183.48611111111111, 73.05410724166398], Cluster: 1
Feature: [187.76388888888889, 73.7038920002713], Cluster: 1
Feature: [140.45138888888889, 62.78057354578448], Cluster: 0
Feature: [214.45138888888889, 64.47693707627289], Cluster: 1
Feature: [139.00694444444446, 60.84834250090979], Cluster: 0
Feature: [193.97916666666666, 71.68579868173953], Cluster: 1
Feature: [198.95833333333334, 70.60215953181287], Cluster: 1
Feature: [106.13194444444444, 31.857156558564885], Cluster: 0
Feature: [211.3125, 63.27337828269924], Cluster: 1
```

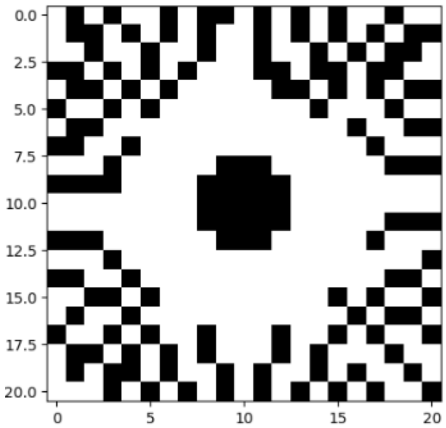


Fig. 04: Feature values for 12x12 blocks

Fig. 05: Output

3. Calculate the spectra energy (without the DC component) feature on the texture.tif image, on the blocks of size 12×12. Is this feature good for the segmentation of the textures on this image? Segment the energy image using the K-means method and comment on the results.

Input image

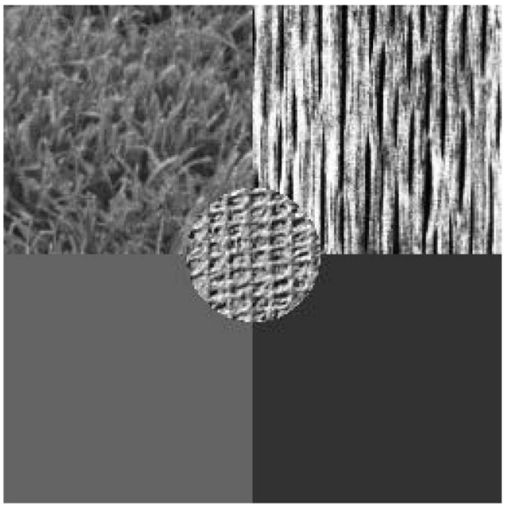
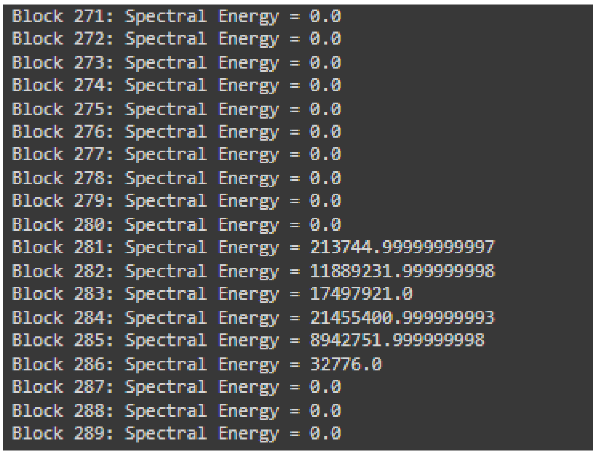


Fig. 06: Input image(texture.tif)

Output



```
Block 271: Spectral Energy = 0.0
Block 272: Spectral Energy = 0.0
Block 273: Spectral Energy = 0.0
Block 274: Spectral Energy = 0.0
Block 275: Spectral Energy = 0.0
Block 276: Spectral Energy = 0.0
Block 277: Spectral Energy = 0.0
Block 278: Spectral Energy = 0.0
Block 279: Spectral Energy = 0.0
Block 280: Spectral Energy = 0.0
Block 281: Spectral Energy = 213744.99999999997
Block 282: Spectral Energy = 11889231.999999998
Block 283: Spectral Energy = 17497921.0
Block 284: Spectral Energy = 21455400.999999993
Block 285: Spectral Energy = 8942751.999999998
Block 286: Spectral Energy = 32776.0
Block 287: Spectral Energy = 0.0
Block 288: Spectral Energy = 0.0
Block 289: Spectral Energy = 0.0
```

Fig. 07: Spectral energy for some blocks

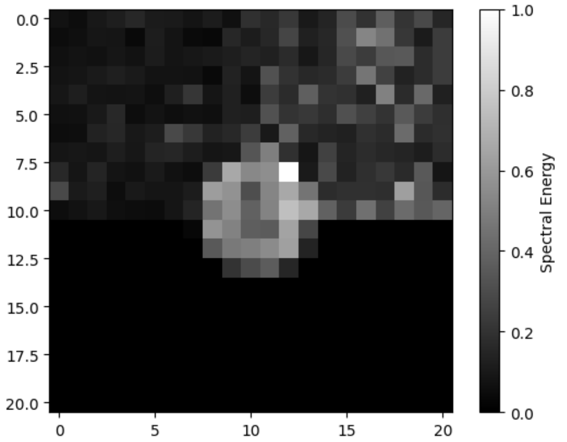


Fig. 08: Energy map

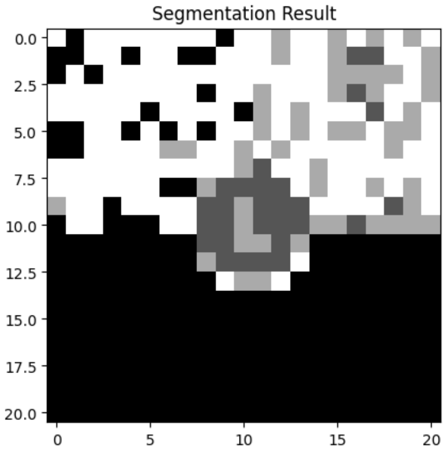


Fig. 09: Output after k means clustering

(4 clusters)

the energy map shows noticeable differences between textures, it means that the spectral energy feature is capturing meaningful information about the textures. Therefore, it can be used as a feature for segmentation.

4. By using segmentation and cv2.inpaint restore the “Efac.jpg” image.

Input

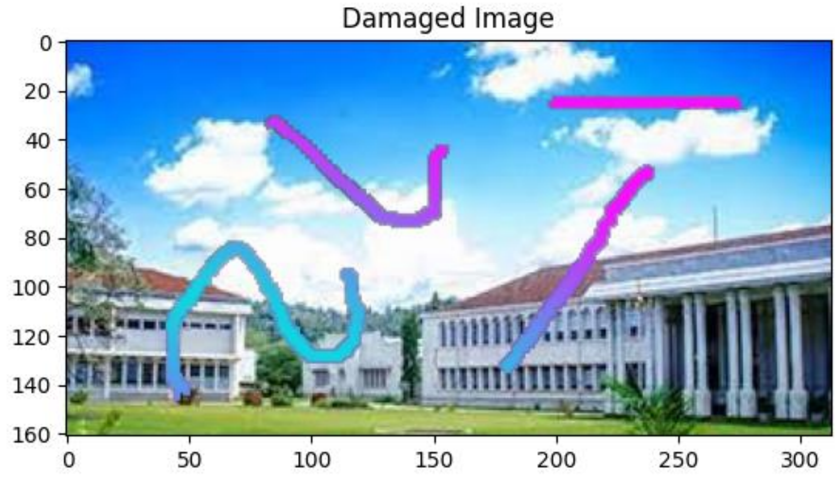
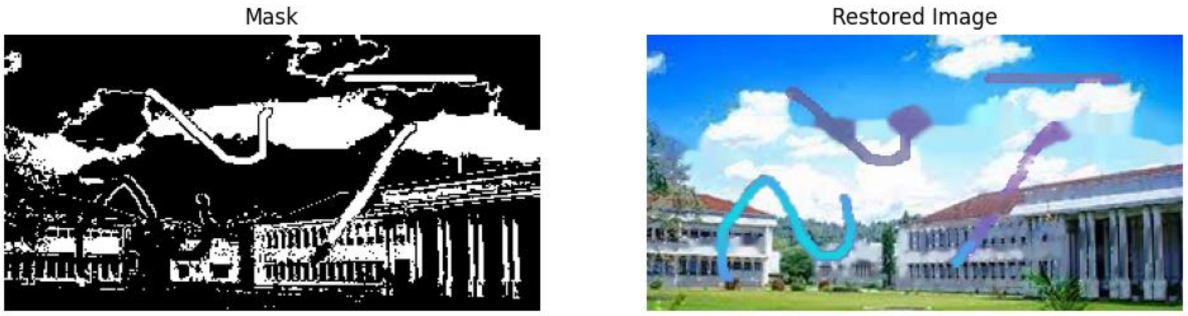


Fig. 10: Damaged image

Steps

- Load the damaged image and check whether it is loaded properly
- Reshape the Image to a 2D Array of Pixels
- Perform K-means Clustering to Segment the Image
- Create a Mask of the Segmented Areas
- Choose One of the Clusters to Inpaint
- Perform Inpainting on the Color Image
- Convert the Restored Image to RGB and Display

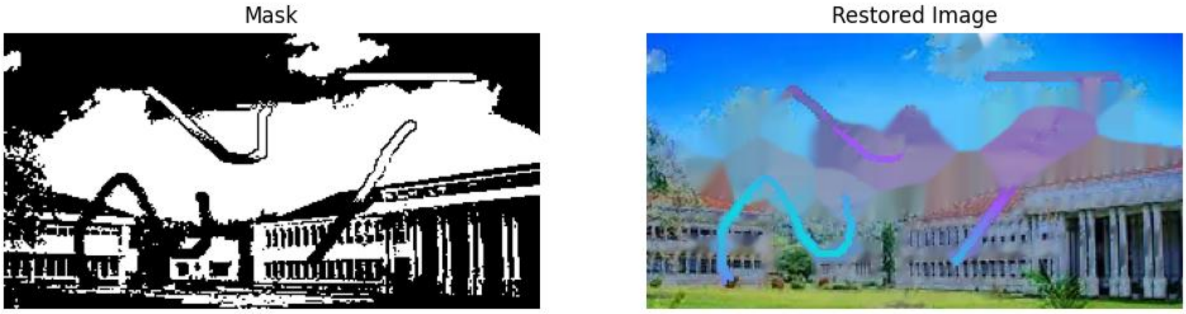
Output



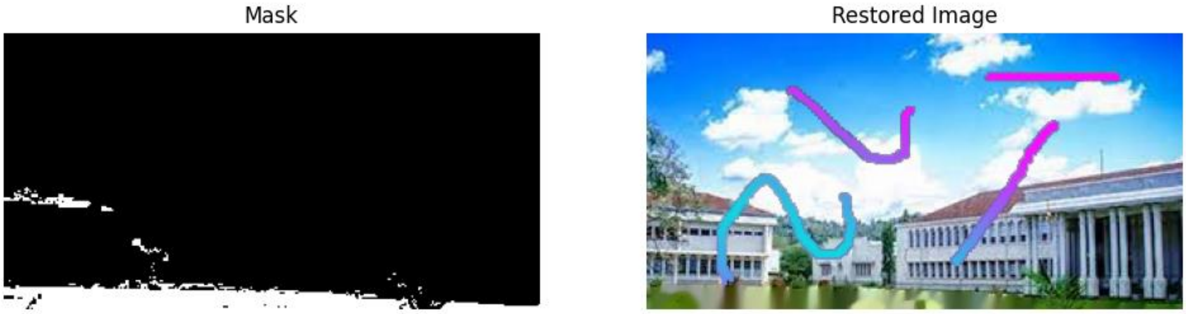
- *Fig. 11: Output: 5 clusters (mask- cluster 1)*

Here, I experimented with no. of clusters and the cluster selected for the mask. Some of the outputs I got are shown below

- 2 clusters (mask- cluster 1)



- 7 clusters (mask – cluster 5)



- 7 clusters (mask – cluster 3)