EE 569 Digital Image Processing: HOMEWORK #4

Problem 1: Texture Analysis

(a) Texture Classification

1.a.1 Abstract and Motivation

A texture is a quasi periodic pattern. Texture analysis is widely used in remote sensing image applications. Texture Classification, texture segmentation and texture synthesis are few applications where texture analysis is vital. Texture classification makes use of set of filters that work in parallel (filter bank) to obtain the response map, which is used to look up on different texture classes.

1.a.2 Approach and Procedure

Texture classification makes use of 5*5 Laws filter constructed by the tensor product of five 1D kernels:

L5 (Level) [1 4 6 4 1]
E5 (Edge) [-1 -2 0 2 1]
S5 (Spot) [-1 0 2 0 -1]
W5 (Wave) [-1 2 0 -2 1]
R5 (Ripple) [1 -4 6 -4 1]

Each tensor product of five 1D kernel decomposes 2D frequency region into 25 frequency bands in spectral domain.

Texture classification involves the following steps:

1. Feature Extraction (Law's filter)
   Twenty-five 5x5 Law's Filter is applied to each input image to extract energy from each pixel in the response map using appropriate boundary extensions leading to a 25-D feature vector for each input image. Average Energy across response map(r) obtained from applying a 5x5 Law filter is computed as:

   $$E = \frac{1}{\#\text{total number of samples}} \sum_{i,j,n} |r\ i,j,n|^2$$

   (i,j) – pixel position

   Feature Vector is composed/stacked using energy components as $\begin{bmatrix} E1 \\ E2 \\ \vdots \\ E25 \end{bmatrix}$

2. Feature Normalization: 12 images have 25D vector, for each image, all dimension/features is standardized.

3. Feature Dimension Reduction (PCA)
   Each feature vector has 25D (dimensions). Any analysis performed should be applied to 25D feature space which has the following short come:
   a. Data points sparse with respect to dimensionality

b. Unreliable modelling

c. Computationally expensive

Hence, Principal Component Analysis is used to reduce 25D feature space to 3D feature space. Since the components are orthogonal, uncorrelated features are obtained, which are easier to handle/ analyse.

4. Procedure for computing PCA:

PCA can be computed either by using eigen values or by singular value decomposition (SVD):
Following is the procedure for computing it using SVD:

a. Consider feature matrix X of size n*m

n: number of features

m: number of data points

Compute Mean mX

X_i: i^{th} column of matrix X

mX= $(\sum_{i=1}^{m} Xi) / m$

$ZX_i = X_i - mX$

b. Compute SVD of ZX

$ZX=U*S*V^T$

U:nxn S:nxm V:mxm

c. Dimensionality reduction from n to n’

Smaller singular values are discarded, Only n’ singular values and n’ columns of U are retained.

Ur=U(:,1:n’)

d. Transformation step: let Rx be dimensionality reduced matrix,

$Rx=Ur^T * Z * X$

Rx is of size n’xm

5. Classifier (k-means)

Clustering is widely used technique for unsupervised learning.
k-means clustering is a classifier which groups ‘k’ clusters
Following is the procedure for implementing k-means clustering:

a. Initialization: Choose k random seeds

b. Iteration: Generalized Lloyd iteration

1. Grouping based on the seeds and sample distance (Euclidean or Mahalanobis)
2. For each cluster, find new centroid and use it as new seed
3. Repeat until convergence

The results are not unique as the convergence depends on its initialization and is non-optimization solution.

k-means ++ : Procedure for initialization seeds for k-means clustering

The *k*-means++ algorithm chooses seeds as follows, assuming the number of clusters is *k*.

1. Select an observation uniformly at random from the data set, *X*. The chosen observation is the first centroid, and is denoted c_1.
2. Compute distances from each observation to c_1. Denote the distance between c_j and the observation *m* as $d(x_m,c_j)$
3. Select the next centroid, c_2 at random from *X* with probability

$$\frac{d^2(xm, cj)}{\sum_{j=1}^{n} d^2(xm, cj)}$$

4. To choose centre *j*:
   a. Compute the distances from each observation to each centroid, and assign each observation to its closest centroid.
   b. For $m = 1,...,n$ and $p = 1,...,j - 1$, select centroid j at random from X with probability

$$\frac{d^2(xm, cp)}{\sum_{h:xh\ \varepsilon\ cp} d^2(xh, cp)}$$

where C_p is the set of all observations closest to centroid c_p and x_m belongs to C_p.

That is, select each subsequent center with a probability proportional to the distance from itself to the closest center that you already chose.

6. Repeat step 4 until *k* centroids are chosen.

Algorithm:

Step 1: For each input image, image mean is subtracted to reduce illumination effects.
Step 2: Boundary is extended by neighbour padding to convolve with 5x5 Law's filter.
Step 3: Convolution with 5x5 Law's filter to obtain response map.
Step 4: Compute the average energy across the response map to obtain a component of 25D feature vector
Step 5: Repeat procedure 3~4 for all 25 Law's filters to obtain 25D (dimensional) feature vector
Step 6: Repeat procedure 2~5 for 12 texture input image
Step 7: Obtain 12x25 matrix
Step 8: Feature Normalization: for each sample (data/image), all the features are standardized
Step 9: Mean is subtracted along each dimension
Step10: Singular Value Decomposition is performed for resulting matrix
Step11: 25D feature vector is reduced to 3D feature vector by retaining the first 3 columns of U and 3 rows of S
Step12: Reduced dimension set is standardized for each input image
Step13: Plot of PCA of feature vector for 12 data points are plotted
Step14: k-means ++: Initialization of k- centroids is performed (Implemented k-means++ on own)
Step15: k-means clustering is performed for 12 data points to obtain the index/label of classification (Implemented k-means on own)

1.a.3,4 Results and Discussion

1. Feature Averaging

Which feature dimension has the strongest discriminant power? Which has the weakest? Please justify your answer.

Feature	Variance	Standard Deviation
Feature1 (L'*L)	1.27E+15	35648835.86
Feature2 (E'*E)	1.45E+10	120352.8437
Feature3 (S'*S)	2.62E+08	16182.74027
Feature4(W'*W)	3.22E+08	17953.70173
Feature5 (R'*R)	9.2E+10	303313.0287
Feature6 (L'*E)	4.28E+12	2068626.154

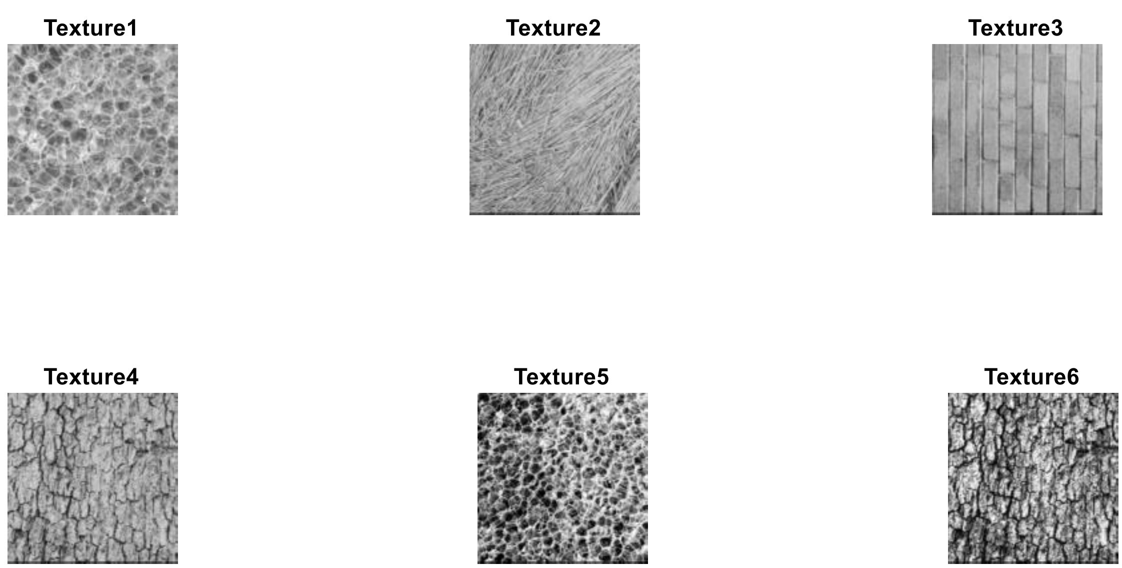
Feature7 (E'*L)	2.22E+12	1488560.793
Feature8 (L'*S)	6.7E+11	818610.4012
Feature9 (S'*L)	1.62E+11	402142.1139
Feature10 (L'*W)	8.11E+11	900830.6914
Feature11 (W'*L)	9.44E+10	307191.6505
Feature12 (L'*R)	1E+13	3164547.114
Feature13 (R'*L)	9.99E+11	999481.7484
Feature14 (W'*R)	4.88E+09	69843.05518
Feature15 (R'*W)	5.51E+09	74244.59107
Feature16 (E'*S)	1.81E+09	42546.79516
Feature17 (S'*E)	1.57E+09	39565.93477
Feature18 (E'*W)	1.54E+09	39283.89293
Feature19 (W'*E)	1.27E+09	35706.97715
Feature20 (E'*R)	1.65E+10	128431.4157
Feature21 (R'*E)	1.38E+10	117612.8352
Feature22 (S'*W)	2.7E+08	16419.94593
Feature23 (W'*S)	2.67E+08	16330.82803
Feature24 (S'*R)	3.53E+09	59443.89866
Feature25 (R'*S)	3.63E+09	60208.71878

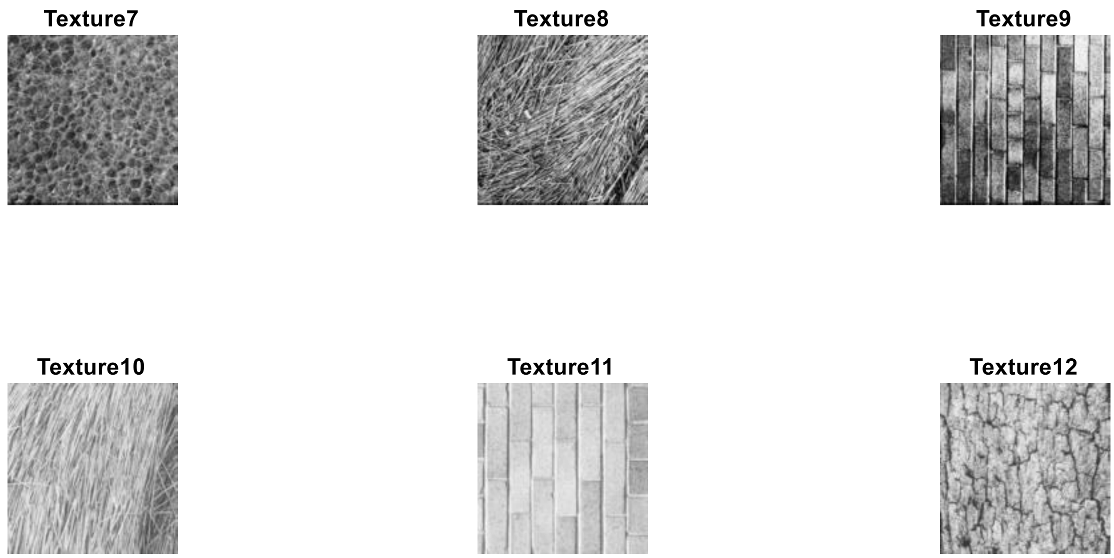
From the above table,

a. The feature with lowest variance is S^T*S (Feature3), thus it has highest discriminant power, since the data points are less spread in the feature space for this feature.

b. The feature with highest variance is L^T*L (Feature1), thus it has the weakest discriminant power, since the data points are more spread in the feature space for this feature.

2. Input Texture images



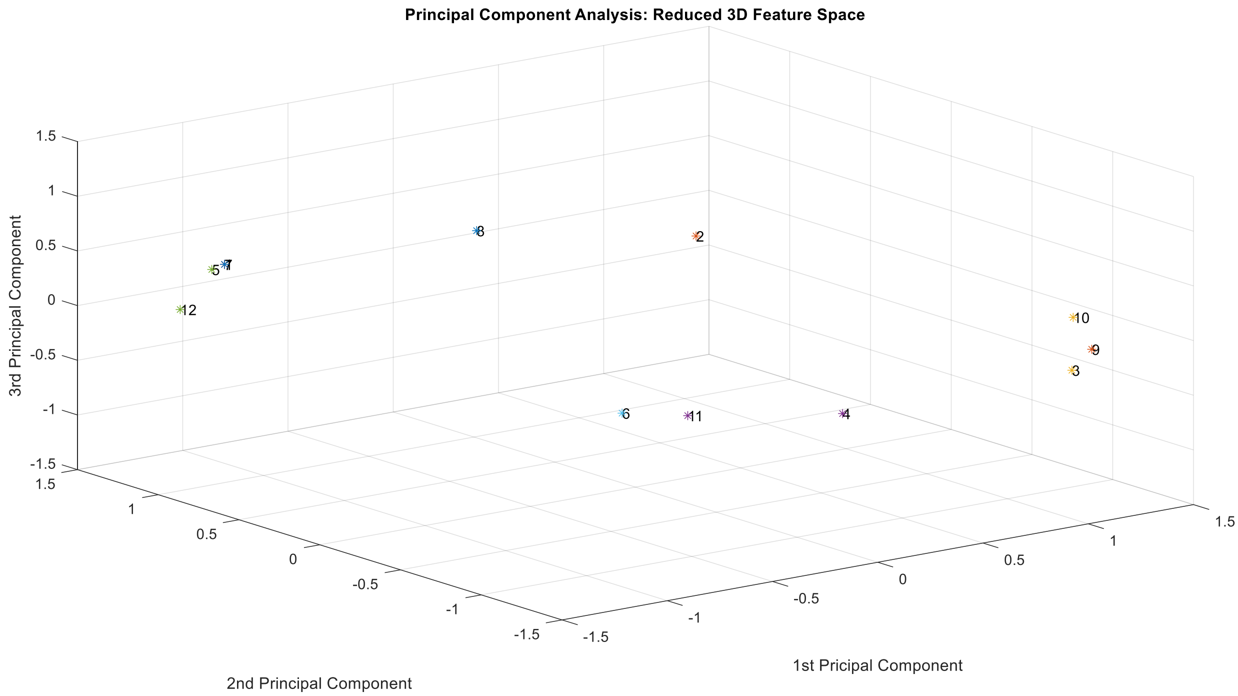


[Ref *] By visual inspection the textures can be classified as follows:

Cluster Group	Texture image
1	1,5,7
2	2,8,10
3	3,9,11
4	4,6,12

3. Reduced 3-D feature vector using Principal Component Analysis

Case1: Energy computed as E = $\frac{1}{\#\text{total number of samples}} \sum_{i,j,n} |r\ i, j, n|^2$ (sum of squares)



Data point 1,7 are close by, so it cannot be seen clearly in the plot

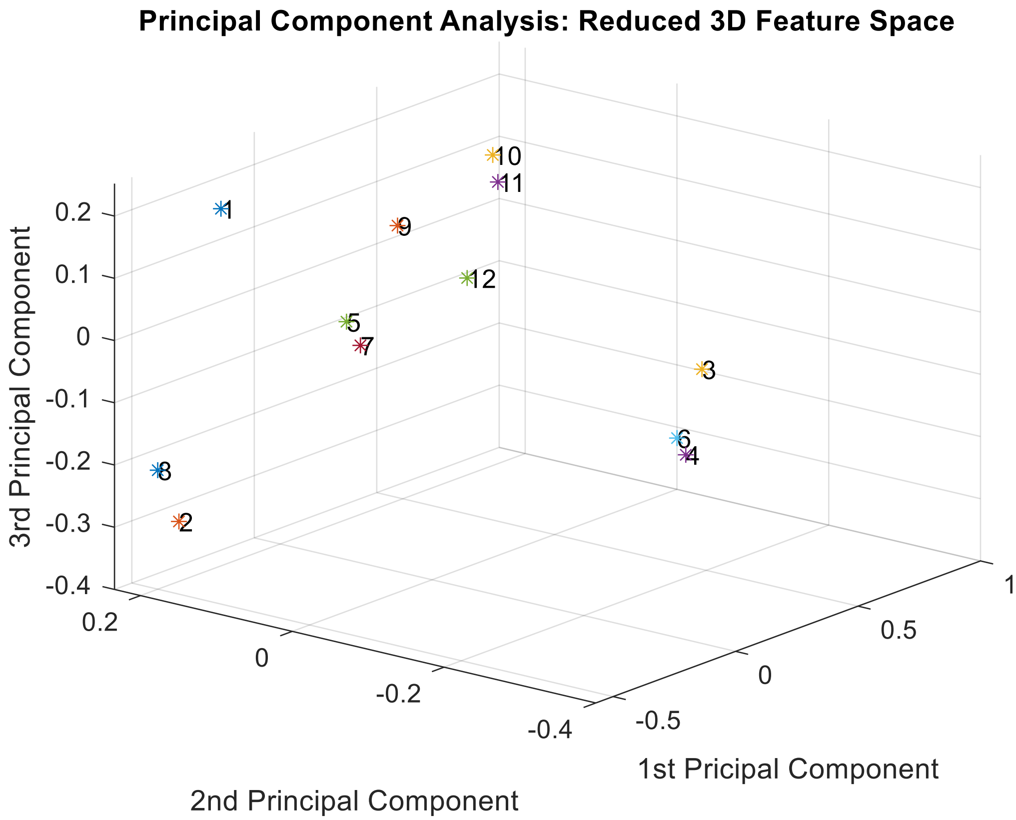
```
Command Window
New to MATLAB? See resources for Getting Started.
>> Problem1_part1
After clustering using k++ initialization
     2
     4
     3
     1
     2
     1
     2
     4
     3
     3
     1
     2

Initialization of centroids using k++ : Index position of data points
    11
     8
     4
     2
```

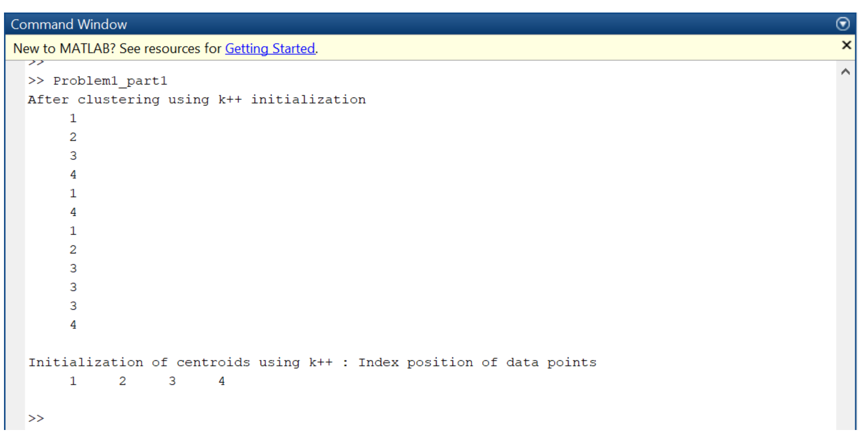
The k-means clustering using k++ initialization (implemented on own) yields the following classification

Cluster Group	Texture image
1	1,5,7,12
2	2,8
3	3,9,10
4	4,6,11

From the above table and [Ref*], it can be inferred that 9 textures are classified correctly

Case2: Energy computed as $E = \frac{1}{\#\text{total number of samples}} \sum_{i,j,n} |r\ i, j, n|$ (sum of absolute)





The k-means clustering using k++ initialization (implemented on own) yields the following classification

Cluster Group	Texture image
1	1,5,7
2	2,8
3	3,9,10,11
4	4,6,12

From the above table and [Ref*], it can be inferred that 11 textures are classified correctly

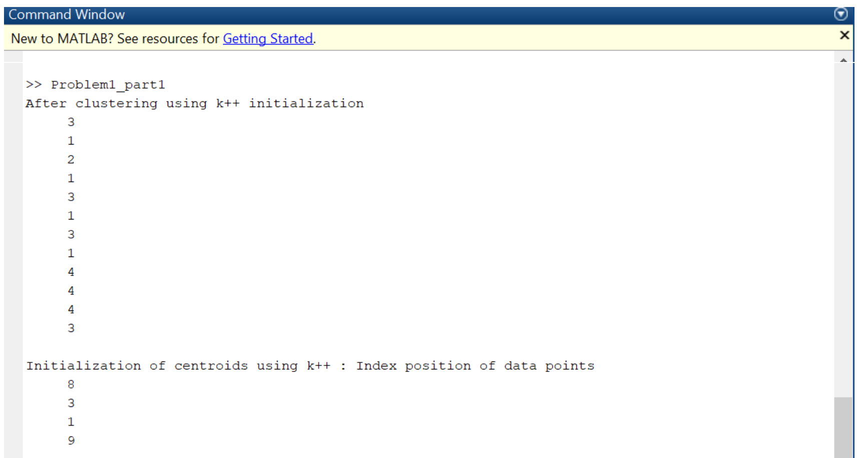
Comment:

1. k-means clustering with random initialization may not yield in better classification accuracy results. So k-means clustering with k- means++ initialization is adopted compared to random selection, so that farthest away centroids are chosen for further clustering.

4. Clustering: K-means algorithm for image clustering based on 25-D and 3-D

The effectiveness of feature dimension reduction over k-means

Case 1: Energy computed as E = $\frac{1}{\#\text{total number of samples}} \sum_{i,j,n} |r\ i, j, n|^2$ (sum of squares)

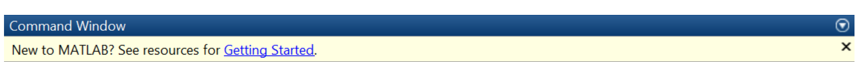


The k-means clustering using k++ initialization (implemented on own) yields the following classification

Cluster Group	Texture images
1	1,5,7
2	2,4,6,8
3	3
4	9,10,11

Instead of using PCA, k-means clustering is used in 25 D.
From the above table and [Ref*], it can be inferred that only 6 textures are classified correctly

Case2: Energy computed as E = $\frac{1}{\#\text{total number of samples}} \sum_{i,j,n} |r\ i, j, n|$ (sum of absolute)



```
>> Problem1_part1
After clustering using k++ initialization
     4
     2
     3
     2
     4
     2
     4
     2
     1
     1
     1
     2

Initialization of centroids using k++ : Index position of data points
    10
     7
     3
     1
```

The k-means clustering using k++ initialization (implemented on own) yields the following classification

Cluster Group	Texture images
1	1,5,7
2	2,4,6,8,12
3	3
4	9,10,11

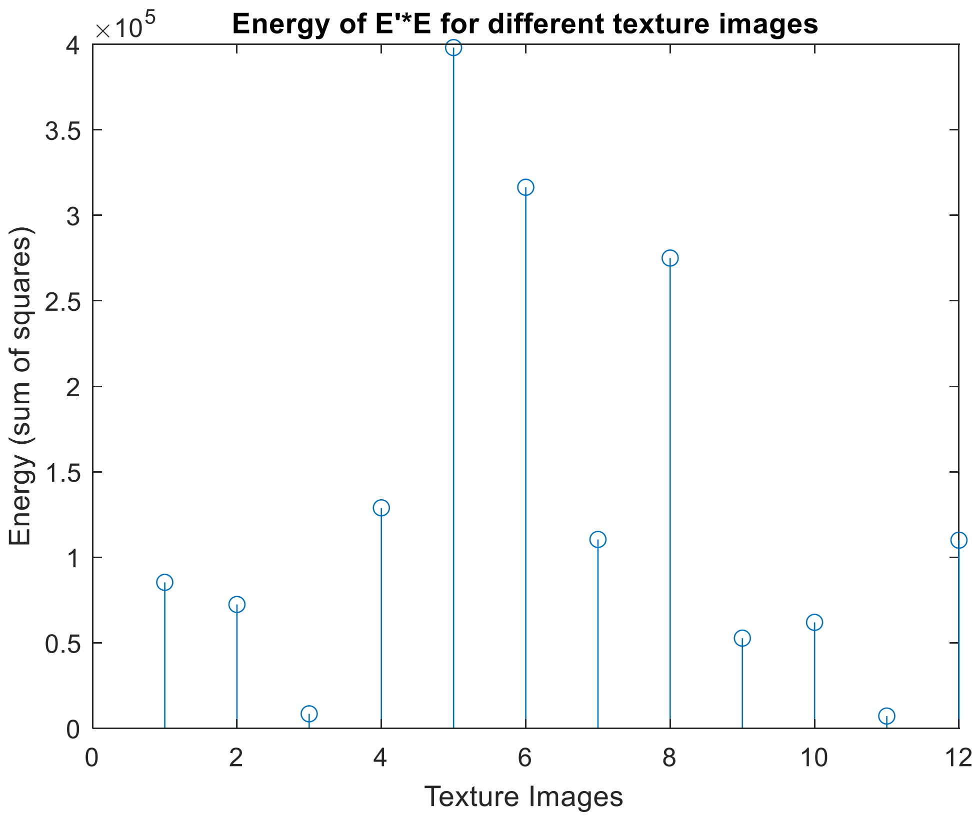
Instead of using PCA, k-means clustering is used in 25 D.
From the above table and [Ref*], it can be inferred that only 6 textures are classified correctly

Comment:

a. The effectiveness of feature dimension reduction over k-means: as the dimension is higher, the data points are sparsely populated leading to high probability that they are misclassified (centroids are far away). When k-means is applied to reduced dimension, the data points are classified in a better way since similar data points come closer in reduced dimension/feature space. relative distances to reduced dimension are computed easily
b. Applying k means to higher dimension involves higher computational cost since, euclidean or Mahanalobis distance has to be computed for each data point to each centroids. Hence smaller dimensions involve less computational cost.

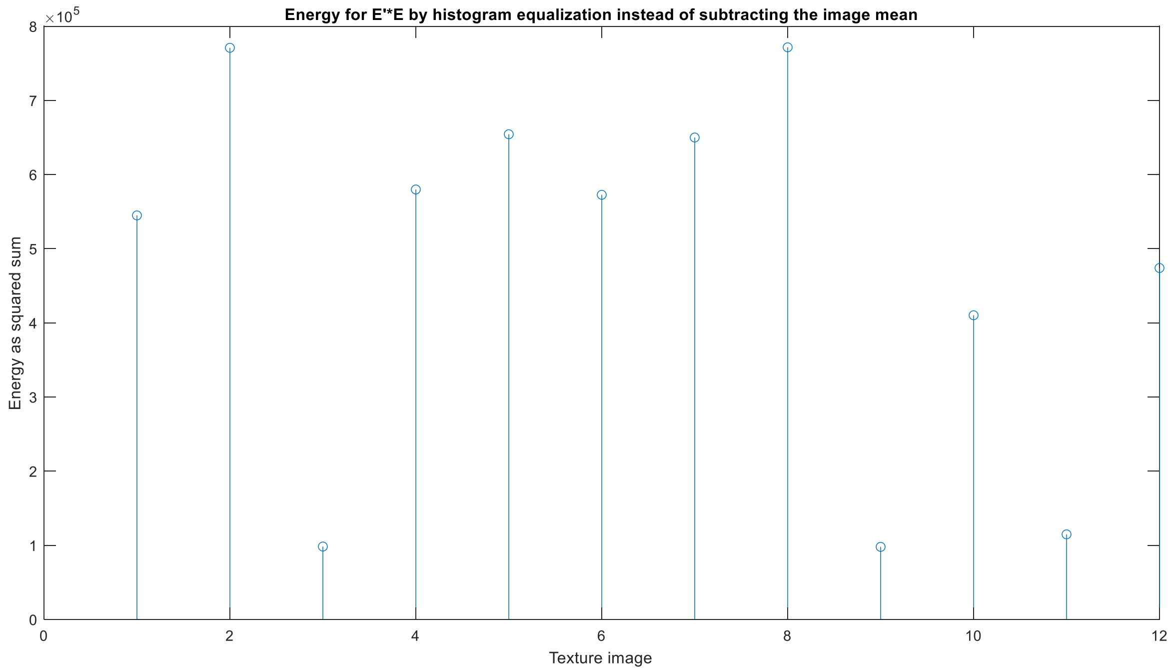
5.From 3,4: Comparison of Energy calculated as sum of squares vs absolute sum

1.From results of classification of texture images, when energy is calculated as average sum, the classification is better. For texture images 5,6,8,9: the images have high contrast (dark) compared to other textures.
2.When mean of the image is subtracted, since dark regions are high in number, the negative scale is squared yielding much higher energy compared to other similar images.
3. This problem can be resolved when absolute value is considered, since the relative magnitude remains the same, yielding similar energy range for similar texture images:
4.



From the above result/plot, it can be seen that among similar texture image:
a. 1,5,7 = 5 has higher energy compared to 1,7
b. 2,8,10 = 8 has higher energy compared to 2,10
c. 3,9,11 = 9 has higher energy compared to 3,9
d. 4,6,12 = 6 has higher energy compared to 4,12

6. Use histogram equalization instead of subtracting the image mean



1.From results of classification of texture images, when energy is calculated as average sum, the classification is better. For texture images 5,6,8,9: the images have high contrast (dark) compared to other textures.
2.When mean of the image is subtracted, since dark regions are high in number, the negative scale is squared yielding much higher energy compared to other similar images.
3. The relative energy of the texture are comparable for similar texture images.

(b) Texture Segmentation

A texture is a quasi periodic pattern. Texture analysis is widely used in remote sensing image applications. Texture Classification, texture segmentation and texture synthesis are few applications where texture analysis is vital. Texture Segmentation problem is similar to texture classification, but each pixel of an image has to be classified into different segments based on the energy calculated per window size.

1.b.1 Abstract and Motivation

Texture classification makes use of 5*5 Laws filter constructed by the tensor product of five 1D kernels:

L5 (Level) [1 4 6 4 1]
E5 (Edge) [-1 -2 0 2 1]
S5 (Spot) [-1 0 2 0 -1]
W5 (Wave) [-1 2 0 -2 1]
R5 (Ripple) [1 -4 6 -4 1]

Each tensor product of five 1D kernel decomposes 2D frequency region into 25 frequency bands in spectral domain.

Texture classification involves the following steps:

1. Feature Extraction (Law's filter)
   Twenty-five 5x5 Law's Filter is applied to each input image to extract 25 D energy/feature vector for each pixel in the input image using appropriate boundary extensions. Average Energy across input image along sliding window of size n is obtained from applying a 25 5x5 Law filter to input image is computed as:

   $$E = \frac{1}{\#\text{total number of samples}} \sum_{i,j,n} |r\ i, j, n|^2$$

   (i,j) – pixel position

   Feature Vector is composed/stacked using energy components as $\begin{bmatrix} E1 \\ E2 \\ \vdots \\ E25 \end{bmatrix}$

2. Each pixel has 25 D feature vector. All the 25D feature vector from all pixel are stacked to form a data set.

3. Classifier (k-means)
   Clustering is widely used technique for unsupervised learning.
   k-means clustering is a classifier which groups 'k' clusters
   Following is the procedure for implementing k-means clustering:
   a. Initialization: Choose k random seeds
   b. Iteration: Generalized Lloyd iteration
      1. Grouping based on the seeds and sample distance (Euclidean or Mahalanobis)
      2. For each cluster, find new centroid and use it as new seed
      3. Repeat until convergence

The results are not unique as the convergence depends on its initialization and is non-optimization solution.

k-means ++ : Procedure for initialization seeds for k-means clustering

The *k*-means++ algorithm chooses seeds as follows, assuming the number of clusters is *k*.

1. Select an observation uniformly at random from the data set, *X*. The chosen observation is the first centroid, and is denoted c_1.
2. Compute distances from each observation to c_1. Denote the distance between c_j and the observation *m* as $d(x_m,c_j)$
3. Select the next centroid, c_2 at random from *X* with probability

$$\frac{\mathrm{d}^2(\mathrm{xm,cj})}{\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{d}^2(\mathrm{xm,cj})}$$

4. To choose centre *j*:
   c. Compute the distances from each observation to each centroid, and assign each observation to its closest centroid.
   d. For $m = 1,...,n$ and $p = 1,...,j - 1$, select centroid *j* at random from *X* with probability

$$\frac{\mathrm{d}^2(\mathrm{xm,cp})}{\sum_{\mathrm{h:xh}\ \varepsilon\ \mathrm{cp}} \mathrm{d}^2(\mathrm{xh,cp})}$$

   where C_p is the set of all observations closest to centroid c_p and x_m belongs to C_p.
   That is, select each subsequent center with a probability proportional to the distance from itself to the closest center that you already chose.
5. Repeat step 4 until *k* centroids are chosen.
6. Once the classification index is obtained for each pixel, depending on the index of classification, a gray Scale type is looked up to find the gray scale magnitude, which is stored as the intensity of the new image.

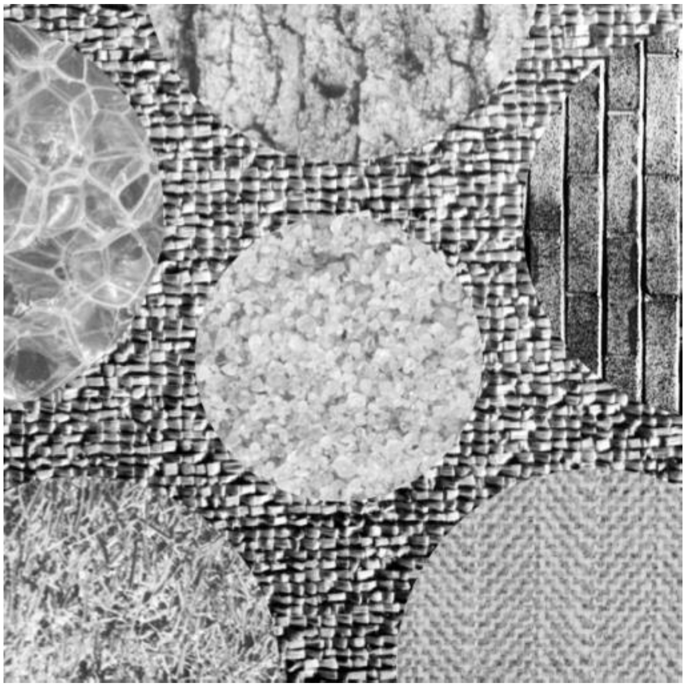
Algorithm:

Step 1: For each input image, image mean is subtracted to reduce illumination effects.
Step 2: Boundary is extended by neighbour padding to convolve with 5x5 Law's filter.
Step 3: Convolution with 25 5x5 Law's filter to obtain response map.
Step 4: Compute the average energy of a pixel across a window size nxn for a single response map to obtain a component of 25D feature vector
Step 5: Repeat procedure 4 for all 25 Law's filters response map to obtain 25D (dimensional) feature vector for each pixel
Step 6: Feature Normalization: for each pixel, 25D feature vector is normalized by dividing it against its first component vector (L^TL)
Step 7: All the pixel energy vector are stacked to form a data set
Step 8: k-means ++: Initialization of k- centroids is performed on data set
Step 9: k-means clustering is performed for data set to obtain the index/label of classification
Step10: Depending on the cluster index, gray scale is assigned as intensity to a new image
Step11: The same procedure is repeated for multiple times for different window size 'n' for an optimal segmentation results.

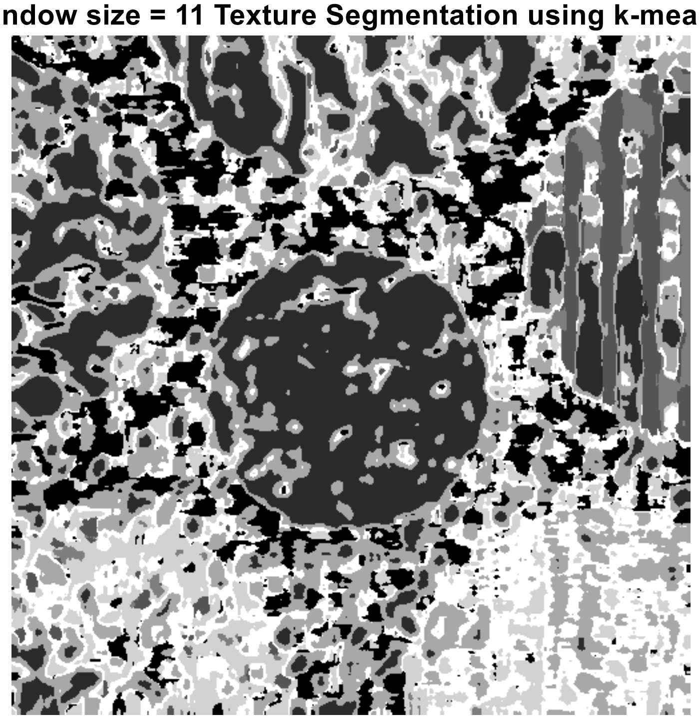
1.b.2 Approach and Procedure

1.b.3 Results

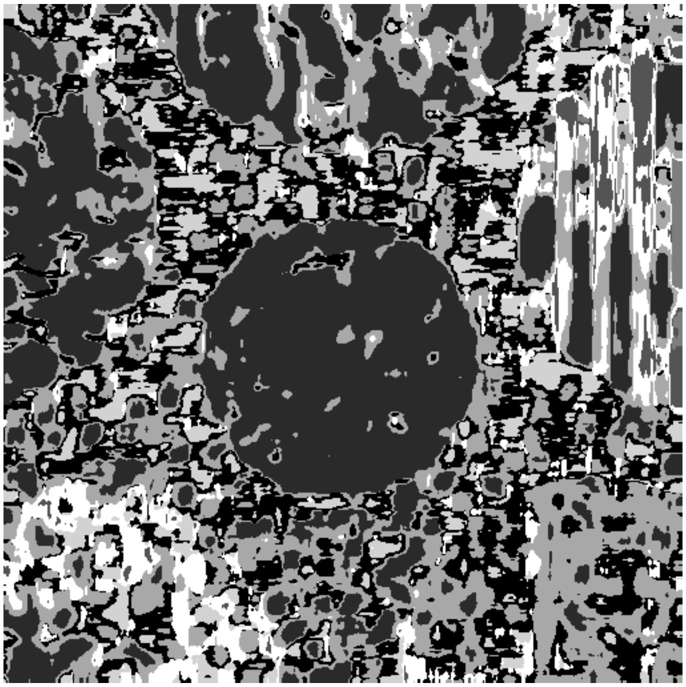
Input Image



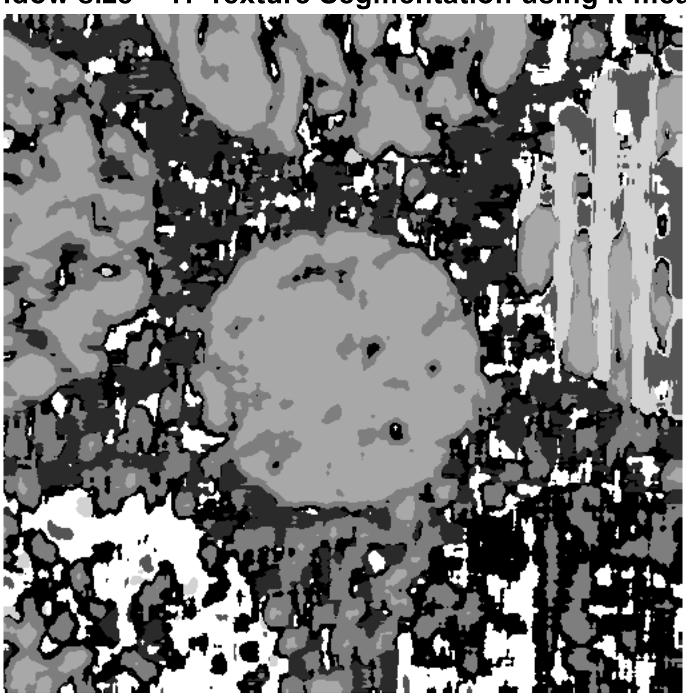
Window size = 11 Texture Segmentation using k-means



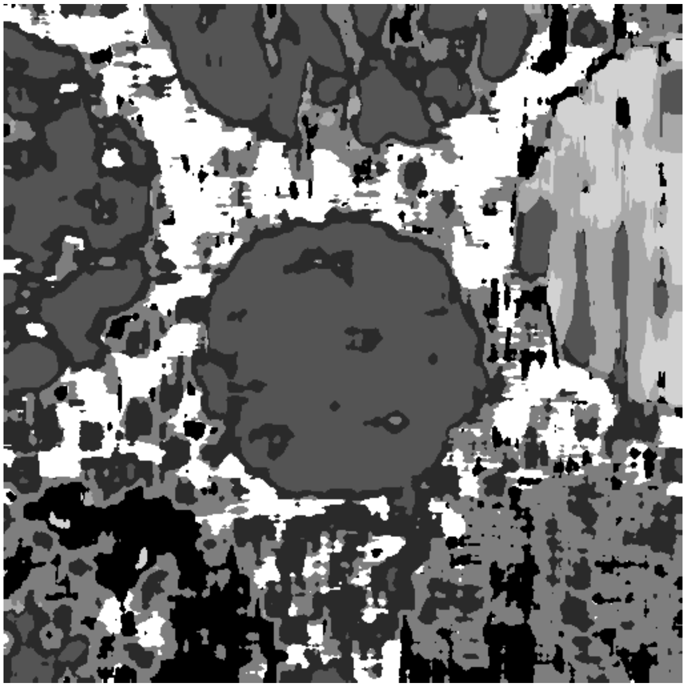
Window size = 13 Texture Segmentation using k-means



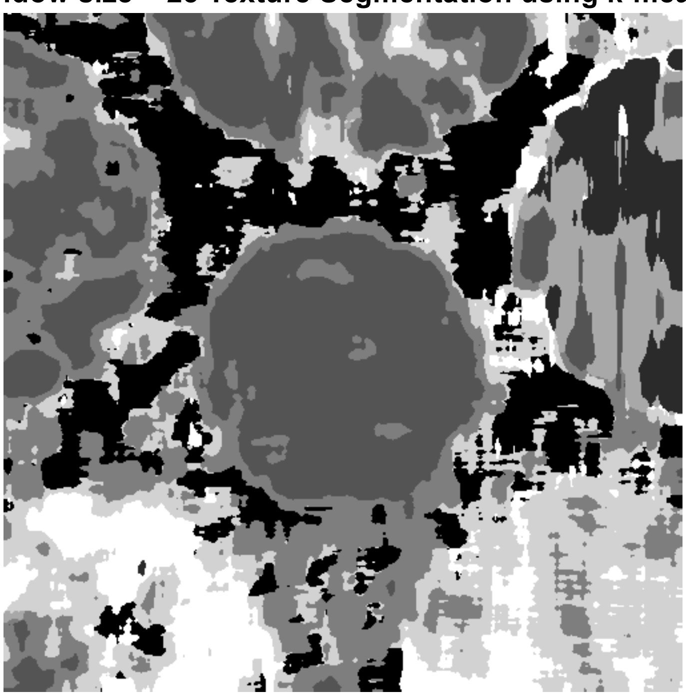
Window size = 17 Texture Segmentation using k-means



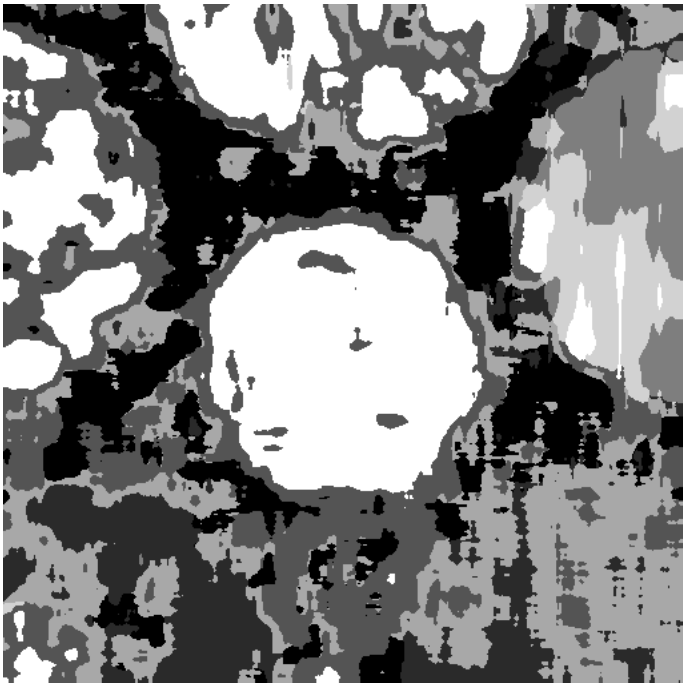
Window size = 19 Texture Segmentation using k-means



Window size = 23 Texture Segmentation using k-means



Window size = 25 Texture Segmentation using k-means



1.b.4 Discussion

1. With Textures segmentation with higher window sizes :
a. The boundary of the image are distinctively seen
b. there are few gaps/holes inside the segments

2.The process is computationally expensive since 25D feature vector has to be stacked to a data set with dimension as its size. Performing k-means is itself expensive, since the data set is more and the euclidean distance to each cluster centroid has to be calculated, which further increases the computational complexity of the image.

3. If an image has more segments/ boundaries within an image, it may lead to extensive computational complex.

4. An alternate approach of adopting Principal component analysis for reducing the dimension of feature vector would be useful.

5. The ideal/recommended window size for this segmentation of image would be 19~23, beyond or below which lot of distortions affect the boundary regions and the interior fill as well.

(c) Advanced Texture Segmentation Techniques

1.c.1 Abstract and Motivation

A texture is a quasi periodic pattern. Texture analysis is widely used in remote sensing image applications. Texture Classification, texture segmentation and texture synthesis are few applications where texture analysis is vital. Texture Segmentation problem is similar to texture classification, but each pixel of an image has to be classified into different segments based on the energy calculated per window size. The following procedure can be adopted for improving the segmentation results obtained from 1(b):

1. Adopt the PCA for feature reduction and, thus, cleaning.
2. Develop a post-processing technique to merge small holes.
3. Enhance the boundary of two adjacent regions by focusing on the texture properties in these two regions only.

1.c.2 Approach and Procedure

Texture classification makes use of 5*5 Laws filter constructed by the tensor product of five 1D kernels:

L5 (Level) [1 4 6 4 1]
E5 (Edge) [-1 -2 0 2 1]
S5 (Spot) [-1 0 2 0 -1]
W5 (Wave) [-1 2 0 -2 1]
R5 (Ripple) [1 -4 6 -4 1]

Each tensor product of five 1D kernel decomposes 2D frequency region into 25 frequency bands in spectral domain.

Texture classification involves the following steps:

1. Feature Extraction (Law's filter)
   Twenty-five 5x5 Law's Filter is applied to each input image to extract 25 D energy/feature vector for each pixel in the input image using appropriate boundary extensions. Average Energy across input image along sliding window of size n is obtained from applying a 25 5x5 Law filter to input image is computed as:

   $$E = \frac{1}{\text{\#total number of samples}} \sum_{i,j,n} |r\ i, j, n|^2$$

   (i,j) – pixel position

   Feature Vector is composed/stacked using energy components as $\begin{bmatrix} E1 \\ E2 \\ \vdots \\ E25 \end{bmatrix}$

2. Each pixel has 25 D feature vector. All the 25D feature vector from all pixel are stacked to form a data set.

3. Feature Dimension Reduction (PCA)
   Each feature vector has 25D (dimensions). Any analysis performed should be applied to 25D feature space which has the following short come:
   a. Data points sparse with respect to dimensionality

b. Unreliable modelling

c. Computationally expensive

Hence, Principal Component Analysis is used to reduce 25D feature space to 3D feature space. Since the components are orthogonal, uncorrelated features are obtained, which are easier to handle/ analyse.

4. PCA can be computed either by using singular value decomposition or eigen value/vector to reduce 25D to a lower dimension 'dim'

3. Classifier (k-means)

Clustering is widely used technique for unsupervised learning.
k-means clustering is a classifier which groups 'k' clusters
Following is the procedure for implementing k-means clustering:

a. Initialization: Choose k random seeds

b. Iteration: Generalized Lloyd iteration

1. Grouping based on the seeds and sample distance (Euclidean or Mahalanobis)
2. For each cluster, find new centroid and use it as new seed
3. Repeat until convergence

The results are not unique as the convergence depends on its initialization and is non-optimization solution.

k-means ++ : Procedure for initialization seeds for k-means clustering

The *k*-means++ algorithm chooses seeds as follows, assuming the number of clusters is *k*.

1.Select an observation uniformly at random from the data set, *X*. The chosen observation is the first centroid, and is denoted c_1.

2. Compute distances from each observation to c_1. Denote the distance between c_j and the observation *m* as $d(x_m, c_j)$

3. Select the next centroid, c_2 at random from *X* with probability

$$\frac{d^2(xm, cj)}{\sum_{j=1}^{n} d^2(xm, cj)}$$

4. To choose centre *j*:

   a. Compute the distances from each observation to each centroid, and assign each observation to its closest centroid.

   b. For $m = 1,...,n$ and $p = 1,...,j - 1$, select centroid *j* at random from *X* with probability

$$\frac{d^2(xm, cp)}{\sum_{h:xh\ \varepsilon\ cp} d^2(xh, cp)}$$

where C_p is the set of all observations closest to centroid c_p and x_m belongs to C_p.

That is, select each subsequent center with a probability proportional to the distance from itself to the closest center that you already chose.

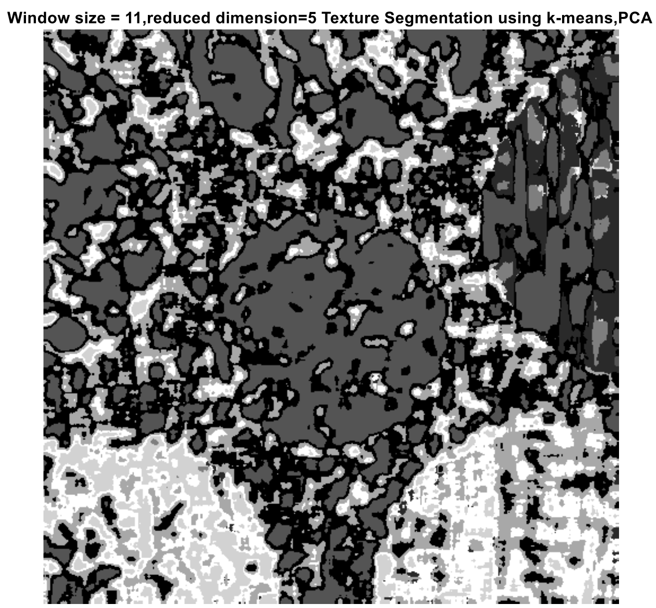
5. Repeat step 4 until *k* centroids are chosen.

6. Once the classification index is obtained for each pixel, depending on the index of classification, a gray Scale type is looked up to find the gray scale magnitude, which is stored as the intensity of the new image.

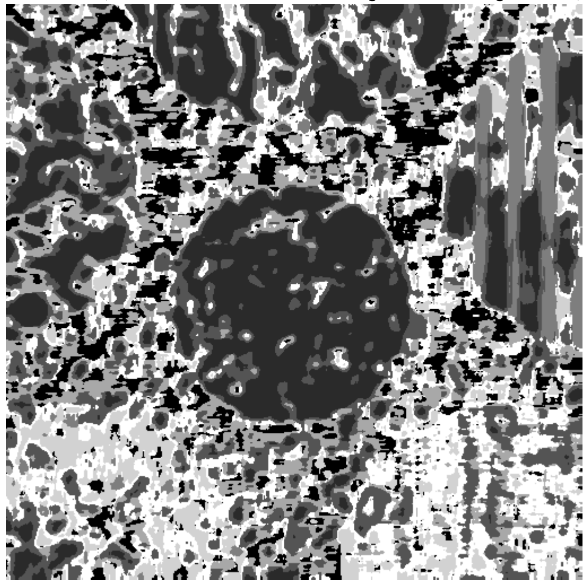
Algorithm:

Step 1: For each input image, image mean is subtracted to reduce illumination effects.
Step 2: Boundary is extended by neighbour padding to convolve with 5x5 Law's filter.
Step 3: Convolution with 25 5x5 Law's filter to obtain response map.
Step 4: Compute the average energy of a pixel across a window size nxn for a single response map to obtain a component of 25D feature vector
Step 5: Repeat procedure 4 for all 25 Law's filters response map to obtain 25D (dimensional) feature vector for each pixel
Step 6: Feature Normalization: for each pixel, 25D feature vector is normalized by dividing it against its first component vector (L^TL)
Step 7: All the pixel energy vector are stacked to form a data set
Step 8: Stacked data set of 25D feature vectors are reduced to lower dimension 'dim'
Step 7: k-means ++: Initialization of k- centroids is performed on data set
Step 8: k-means clustering is performed for data set to obtain the index/label of classification
Step 9: Depending on the cluster index, gray scale is assigned as intensity to a new image
Step10: The same procedure is repeated for multiple times for different dimension 'dim' for an optimal segmentation results.

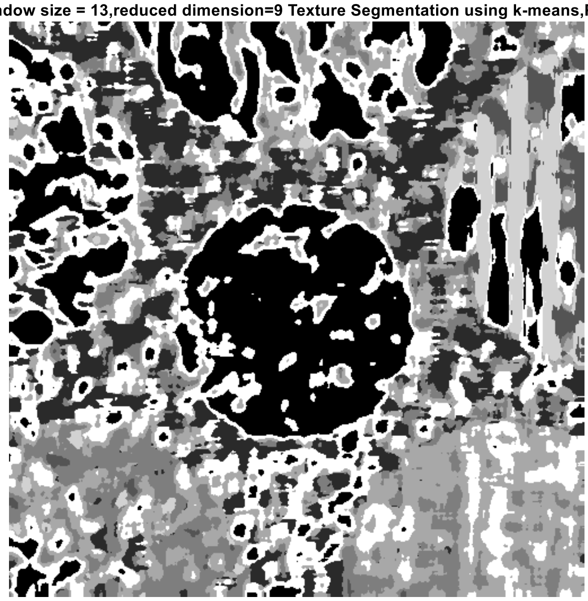
1.c.3 Results



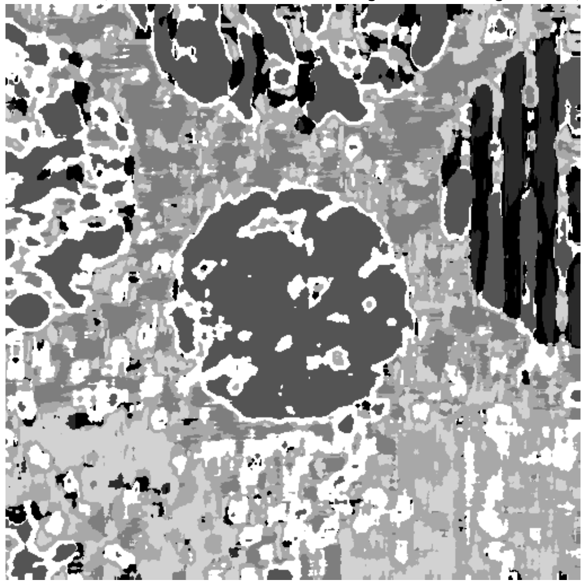
Window size = 11,reduced dimension=7 Texture Segmentation using k-means,PCA



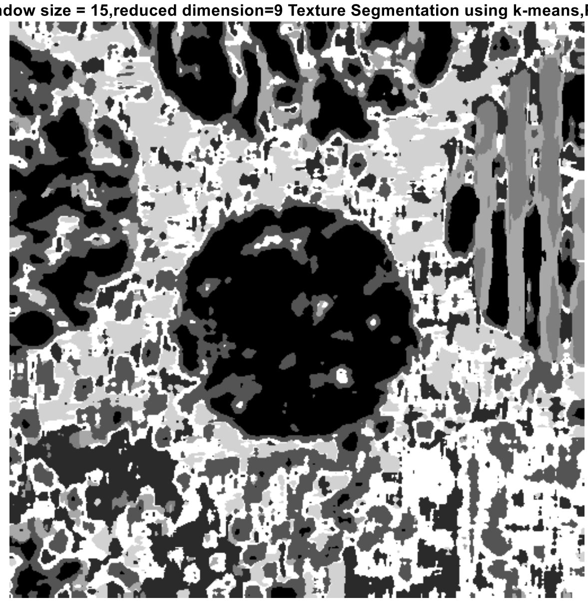
Window size = 13,reduced dimension=9 Texture Segmentation using k-means,PCA



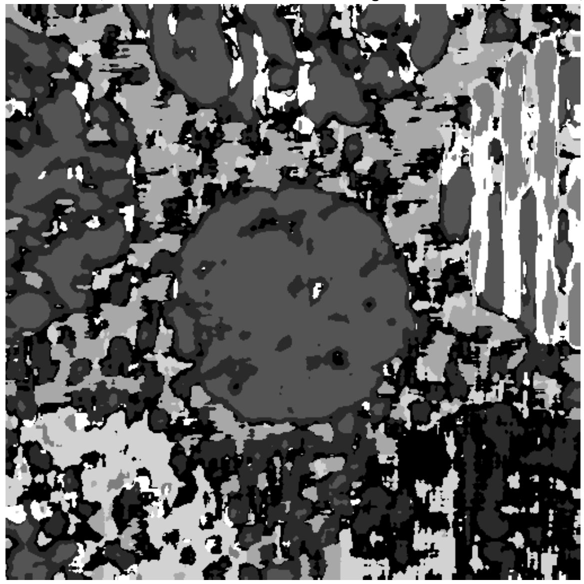
Window size = 15,reduced dimension=7 Texture Segmentation using k-means,PCA



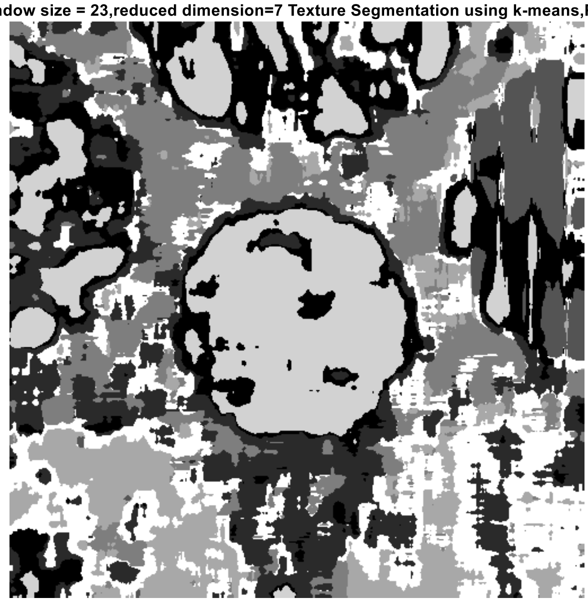
Window size = 15,reduced dimension=9 Texture Segmentation using k-means,PCA



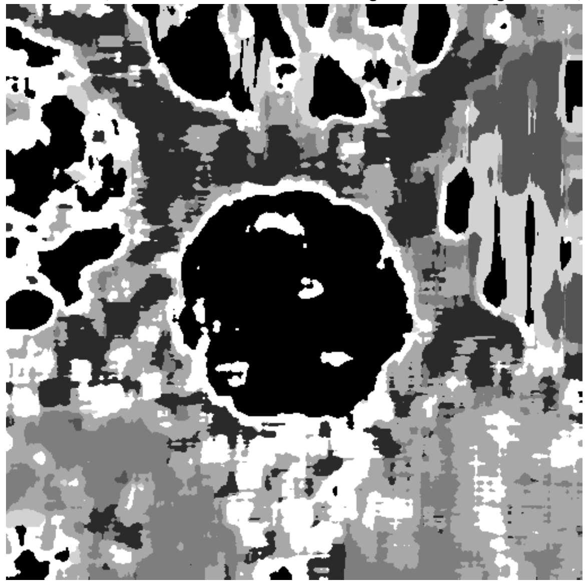
Window size = 17,reduced dimension=11 Texture Segmentation using k-means,PCA



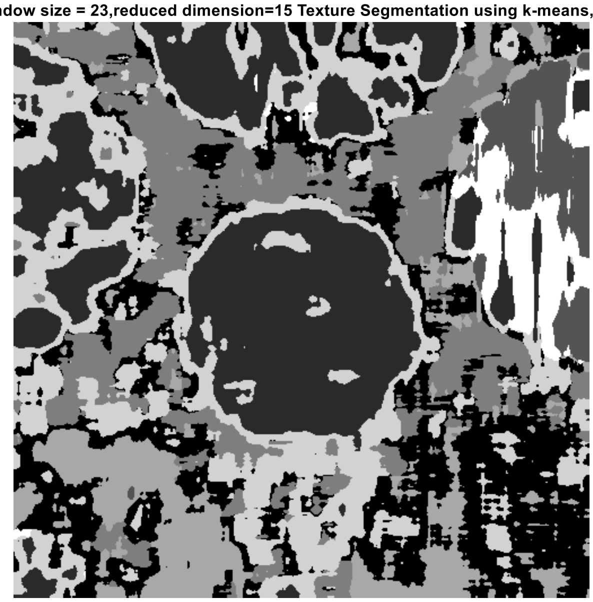
Window size = 23,reduced dimension=7 Texture Segmentation using k-means,PCA



Window size = 23,reduced dimension=9 Texture Segmentation using k-means,PCA



Window size = 23,reduced dimension=15 Texture Segmentation using k-means,PCA



1.c.4 Discussion

1. PCA improves the result since, it is less computationally expensive compared to the previous approach

2. Since PCA yields orthogonal components, the data would be less correlated.

3. The results were better when dimension is between 7~15 yielding better boundaries and less distortions.

4. Computing time is reduced extensively.

5. The resulting segmented images are good, but they also depend on the initialization of the k-means Classifier

6. For very lower or higher dimensions, the distortions are high. It is important to choose a vital dimension.

Problem 2: Image Feature Extractor

(a) SIFT

2.a.1 Abstract and Motivation

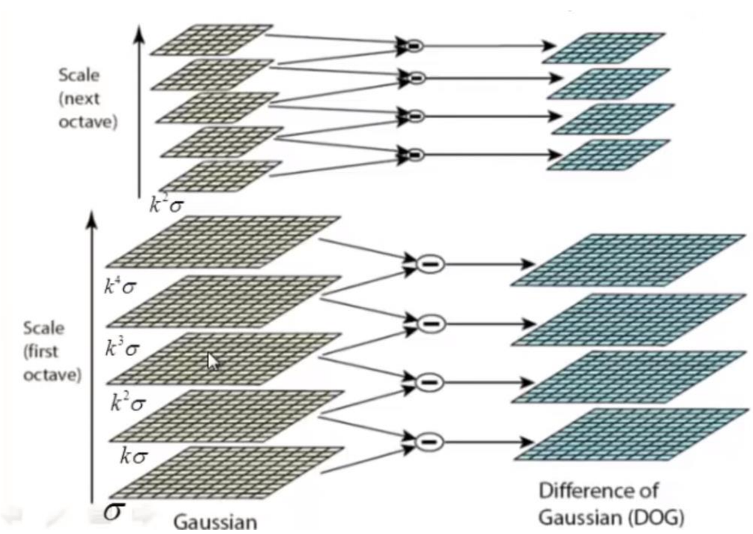
Scale Invariant Feature Transform (SIFT) is a mathematical model for extracting the discriminant/ distinct features of an object/image using Pre-Convolutional Neural Network. The main advantage of this feature extractor is, it is invariant to scaling, rotational geometric modification, partial invariant to illumination and affine/ 3D projection. The efficiency/productivity of a feature extractor lies in the discriminant power of the feature of an object/ image rather than repetitive features extracted which does not determine an object uniquely. It has various application ranging from object detection to 3D modelling, video tracking etc.

2.a.2 Approach and Procedure

[Ref 1]

The following steps are involved in extracting the key-points and descriptors of an object using SIFT:

1. Scale-Space Extrema Detection
   - It is important to examine all scales of an image to identify scale invariant feature.
   - Pyramid images are obtained by first blurring the input image with σ, $k\sigma$, $k^2\sigma$, $k^3\sigma$, $k^4\sigma$ etc (blurring factor of gaussian filter) forming the first octave, the input image is resampled to higher scale (resolution) using bilinear interpolation with a pixel spacing of 1.5 in each direction to obtain the next octave by blurring the re-sampled image with multiple $k^2\sigma$, $k^3\sigma$, $k^4\sigma$ (Gaussian parameter for blurring/smoothing) and this procedure is repeated for several octaves with $\sigma = 1.6$ and $k = \sqrt{2}$. Typically 3-5 scales are considered per octave so that repeatability is high.



   - Lindeberg has shown that under general assumptions on scale invariance, the Gaussian kernel and its derivatives are the only possible smoothing kernels for scale space analysis. Blob/Interest points can be determined using Laplacian of gaussian. Difference of Gaussian (DoG) is obtained by subtracting the pyramid images of scale space octaves, to determine the key-points. Laplacian of Gaussian can be approximated to Difference of Gaussian using Heat equation as follows:

Heat equation: $\frac{\partial G}{\partial \sigma} = \sigma^2 \nabla^2 G$

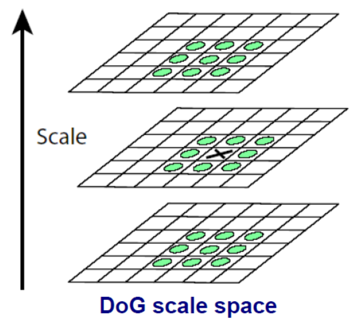
$$\frac{\partial G}{\partial \sigma} = \frac{G(x,y,k\sigma) - G(x,y,\sigma)}{k\sigma - \sigma}$$

Difference of Gaussian = $G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\nabla^2 G$

Interpretation of scale space contours: When gaussian/smoothing kernel is applied across different scaled versions, the parent node of interval tree (determining the key points using maxima/minima), branches into child as the resolution of the image is increased. The stability of the choice of key-point is determined by length of the interval (so that it is more stable across multiple scale spaces)

2. Key-point Localization

- Extrema/minima is determined by observing pixel (X) with 3x3 neighbourhood of the present, higher and lower scales of difference of gaussian results.
- If the centre point is maximum/minimum compared to rest 26 neighbouring points, then it is considered as a candidate key points.



- Candidate key points are considered as key points only after rejecting the outliers:

1. Low contrast candidates are thrown/discarded if $|D(X)| < 0.03$ in the following equation

$$D(\hat{X}) = D + \frac{1}{2}\frac{\partial D^T}{\partial X}\hat{X}$$

2. Poorly localized candidates along the edges are determined by Hessain matrix. Key points along the edges have to be discarded since they are not the salient feature even if they are part of an edge.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

If the ratio of $\frac{(Dxx + Dyy)^2}{DxxDyy - Dxy^2} > \frac{(r+1)^2}{r}$ where r=10 for SIFT, then discard the candidate key-points.

3. Taking derivative of Derivative of Gaussian (DoG) where $X=(x, y, \sigma)^T$

$$D(X) = D + \frac{\partial D^T}{\partial X}X + \frac{1}{2}X^T\frac{\partial^2 D}{\partial X^2}X$$

Minima or maxima is located at

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2}\frac{\partial D}{\partial X}$$

The Value of D(X) at minima/maxima must be large $|D(X)| >$ threshold

3. Orientation Assignment

- To achieve rotation invariance, orientation is assigned to key point.

- For every 10° variation in orientation, magnitude is calculated to account for 1 bin which is weighted by gradient magnitude and Gaussian – weighted window 4*4. 360° accounts for 36 bins. Histogram of magnitude vs 36 bins are plotted and the angle corresponding to top 20% of highest magnitude (or highest) determine the orientation of the key point.

4. Key-point descriptor

- Along the orientation of the key-point, 16x16 neighbourhood around the key point is divided into 16 4*4 sub blocks, with each sub-block forming 8 bin gradient orientation histogram using 16 samples.
- Gradient orientation is considered since it is a robust representation and is less sensitive to illumination or intensity differences.
- In order to sample the image at a larger scale, the same procedure is repeated for higher octave using 2*2 region resulting in 128 +8*2*2 = 160 samples.
- 128 bin values are embedded into high dimensional key point descriptor vector.
- 128D vector can be reduced to lower dimension with the help of Principal Component Analysis.

Key points along with descriptors form the features of an object/image.

2.a.3 Answers

I. SIFT is robust to scaling, translation, rotation geometric modification, partial invariant to illumination and affine/ 3D projection.

II.

❖ Robustness to scaling is achieved by:

Scale-Space Extrema Detection

- ➢ It is important to examine all scales of an image to identify scale invariant feature.
- ➢ Pyramid images are obtained by first blurring the input image with σ, kσ, $k^2σ$, $k^3σ$, $k^4σ$ etc (blurring factor of gaussian filter) forming the first octave, the input image is resampled to higher scale (resolution) to obtain the next octave by blurring the re-sampled image with multiple $k^2σ$, $k^3σ$, $k^4σ$ (Gaussian parameter for blurring/smoothing) and this procedure is repeated for several octaves with σ =1.6 and k = $\sqrt{2}$. Typically 3-5 scales are considered per octave so that repeatability is high. The key locations are picked such that they are stable across multiple scales.

❖ Robustness to rotation is achieved by:

- ➢ Each key point is assigned a canonical orientation so that the image descriptors are invariant to rotation. In order to make this as stable as possible against lighting or contrast changes, the orientation is determined by the peak in a histogram of local image gradient orientations.
- ➢ Orientation Assignment, for every 10° variation in orientation, magnitude is calculated to account for 1 bin which is weighted by gradient magnitude and Gaussian – weighted window 4*4. 360° accounts for 36 bins. Histogram of magnitude vs 36 bins are plotted and the angle corresponding to top 20% of highest magnitude (or highest) determine the orientation of the key point.
- ➢ Key-point descriptor, along the orientation of the key-point, 16x16 neighbourhood around the key point is divided into 16 4*4 sub blocks, with each sub-block forming 8 bin gradient orientation histogram using 16 samples.
- ➢ Gradient orientation is considered since it is a robust representation and is less sensitive to illumination or intensity differences.
- ➢ 128 bin values are embedded into high dimensional key point descriptor vector. 128D vector can be reduced to lower dimension with the help of Principal Component Analysis.

- In order to sample the image at a larger scale, the same procedure is repeated for higher octave using 2*2 region resulting in 128 +8*2*2 = 160 samples.

  - Robustness to affine/ 3D projection:

- The features achieve partial invariance to local variations, such as affine or 3D projections, by blurring image gradient locations. Lindeberg has shown that under general assumptions on scale invariance, the Gaussian kernel and its derivatives are the only possible smoothing kernels for scale space analysis. Blob/Interest points can be determined using Laplacian of gaussian. Difference of Gaussian (DoG) is obtained by subtracting the pyramid images of scale space octaves, to determine the key-points.
- Descriptor is further modified to improve its stability to changes in affine projection and illumination by the peak in a histogram of local image gradient orientations.

III.

  - Robustness to illumination change

- It is enhanced by thresholding the gradient magnitudes at a value of 0.1 times the maximum possible gradient value. This reduces the effect of a change in illumination direction for a surface with 3D relief, as an illumination change may result in large changes to gradient magnitude but is likely to have less influence on gradient orientation.
- Descriptor is further modified to improve its stability to changes in affine projection and illumination by the peak in a histogram of local image gradient orientations.

IV. The advantage of using Difference of Gaussians (DoG) instead of Laplacian of Gaussian(LoG) is speed.

V. SIFT’s output vector size in original paper is 160.

- Key-point descriptor: Along the orientation of the key-point, 16x16 neighbourhood around the key point is divided into 16 4*4 sub blocks, with each sub-block forming 8 bin gradient orientation histogram using 16 samples.
- Gradient orientation is considered since it is a robust representation and is less sensitive to illumination or intensity differences.
- In order to sample the image at a larger scale, the same procedure is repeated for higher octave using 2*2 region resulting in 128 +8*2*2 = 160 samples.

(b) Image Matching

2.b.1 Abstract and Motivation

Scale Invariant Feature Transform (SIFT) is a mathematical model for extracting the discriminant/ distinct features of an object/image using Pre-Convolutional Neural Network. The main advantage of this feature extractor is, it is invariant to scaling, rotational geometric modification, partial invariant to illumination and affine/ 3D projection. The efficiency/productivity of a feature extractor lies in the discriminant power of the feature of an object/ image rather than repetitive features extracted which does not determine an object uniquely. It has various application ranging from object detection to 3D modelling, video tracking etc. Image matching is one such application.

2.b.2 Approach and Procedure

[Ref 1] yields the procedure for identifying key points and to determine its descriptor of reference objects/ training images.

To find a match between two images (in-general approach/procedure):
1. It is important to identify distinct/unique features in the object/image.
2. Discriminant power of feature extraction determines the efficiency of matching an object.
3. SIFT is preferred for extracting the features since it is invariant to scaling, rotational geometric modification, partial invariant to illumination and affine/ 3D projection.
4. The SIFT keys, descriptors for sample/training images and the test image is obtained.
5. Nearest- neighbouring method is used for indexing/matching the key points.
6. The nearest neighbours are defined as the key points with minimum Euclidean distance from the given descriptor vector. The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbour to the distance of the second closest (original paper: ratio=0.8).
7. However, a modification of the k-d tree algorithm called the best-bin-first search method (Beis & Lowe) can identify the nearest neighbours with high probability using only a limited amount of computation.
8. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches.

Algorithm:

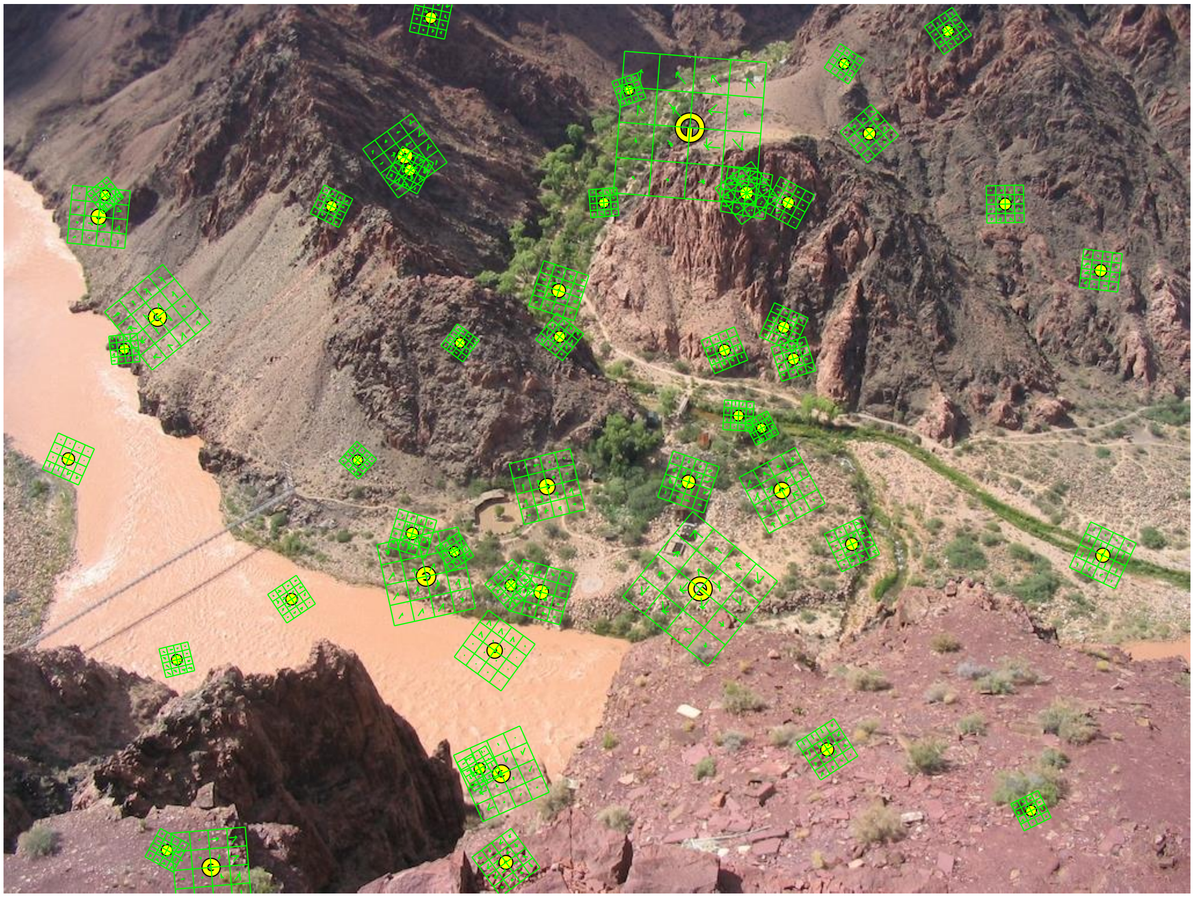
Step 1: SIFT key points are first extracted from a set of reference images (here: river1 image).
Step 2: Find the largest l2 norm of the descriptor of river1 image.
Step 3: Extract SIFT key points and descriptors of river2 image.
Step 4: For the largest l2 norm of descriptor of river1 image, find the Euclidean distance to each of the river2 Image descriptors.
Step 5: The minimum Euclidean distance yields the match between the two river images.
Step 6: The corresponding key points and descriptors are plotted

2.b.3 Results

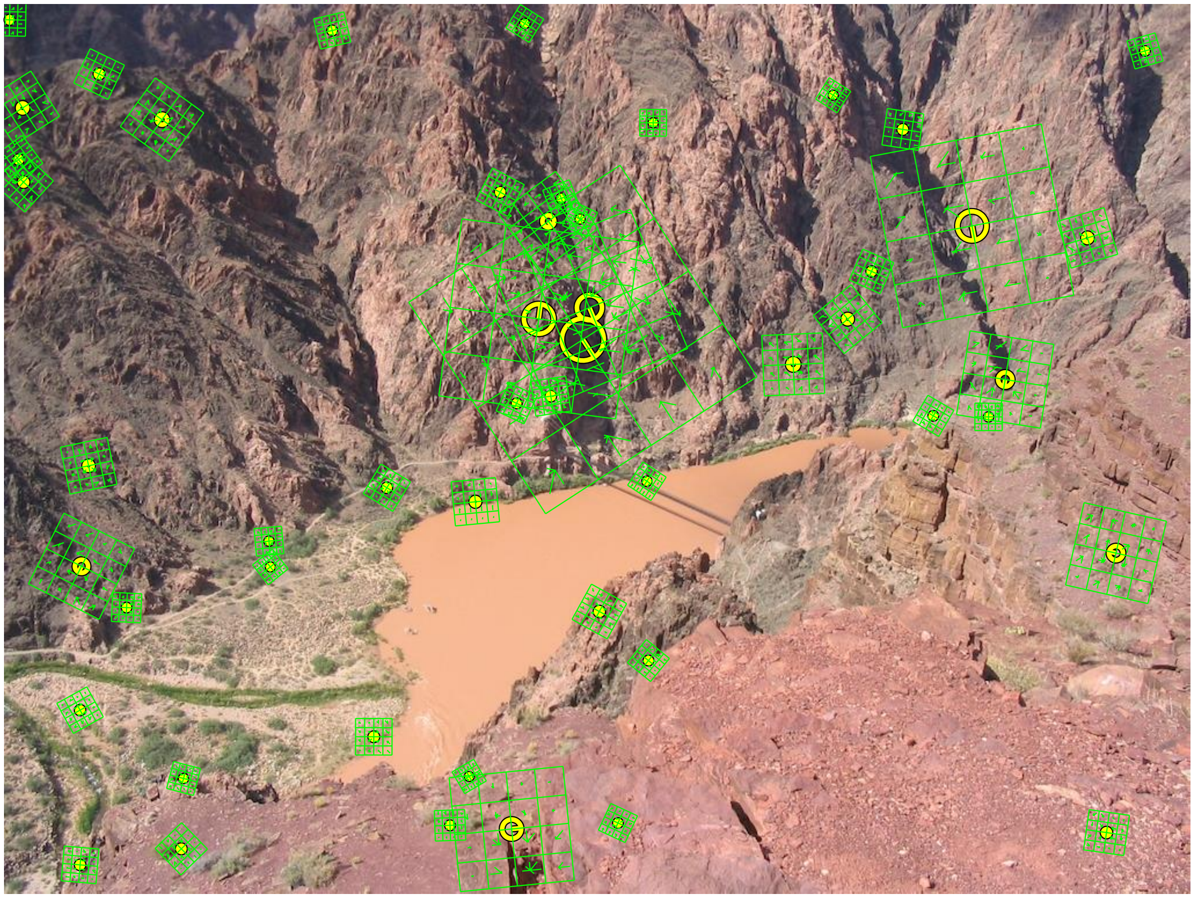
river1 image

river2 image

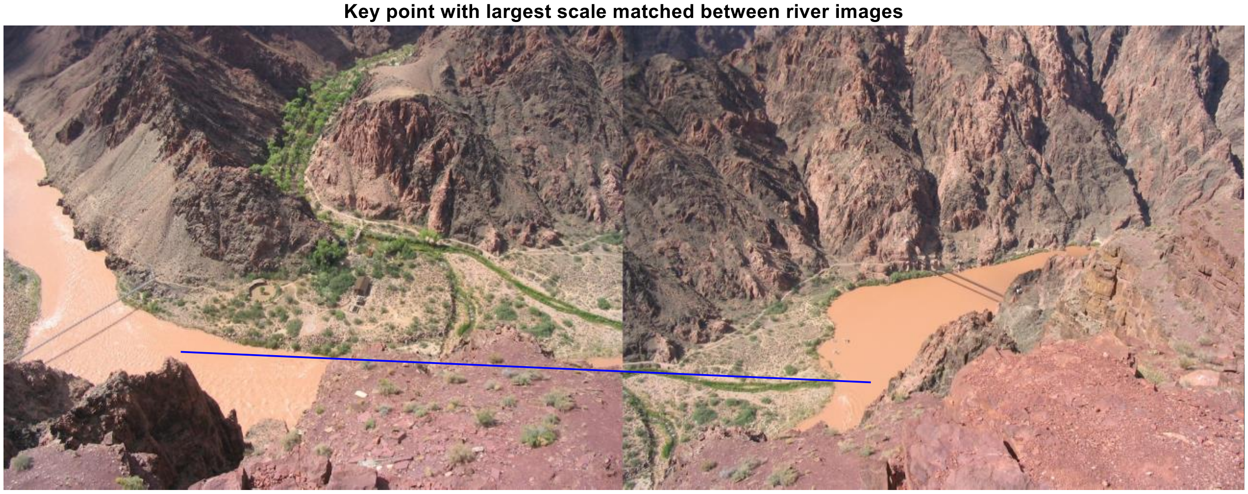
river1 image with sample key points displayed

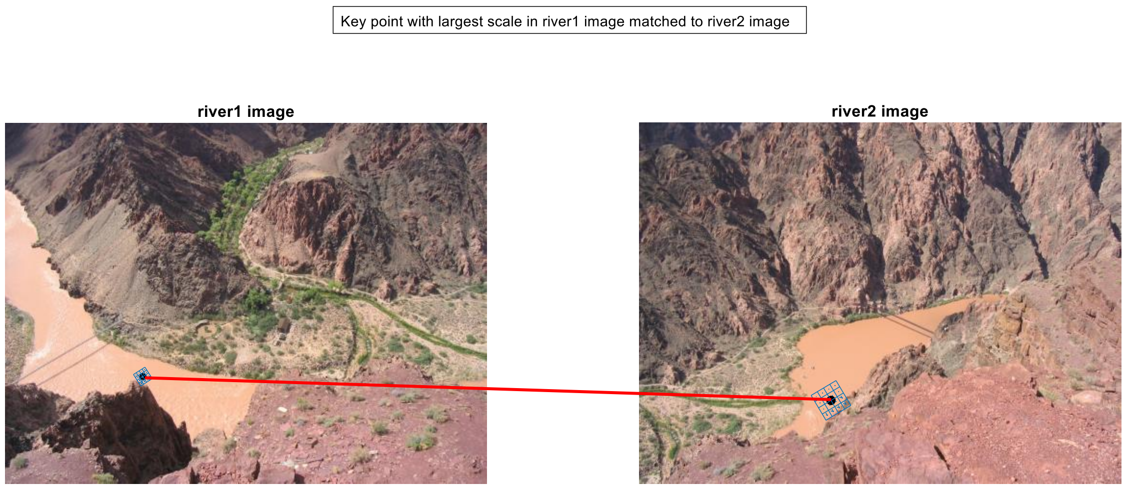


river2 image with sample key points displayed



Key point with largest scale matched between river images



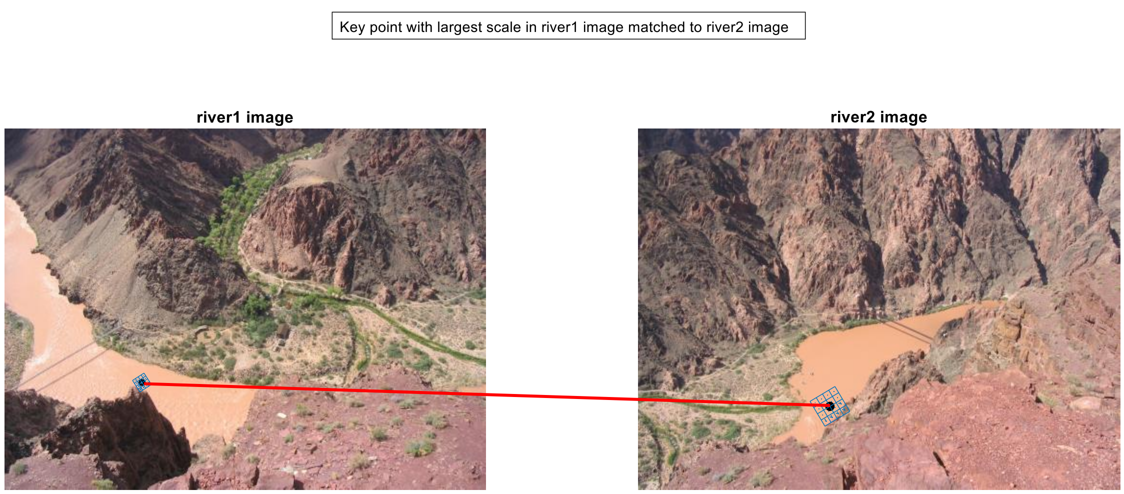


2.b.4 Discussion

1. Location, scale and orientation of matched key points are as follows:

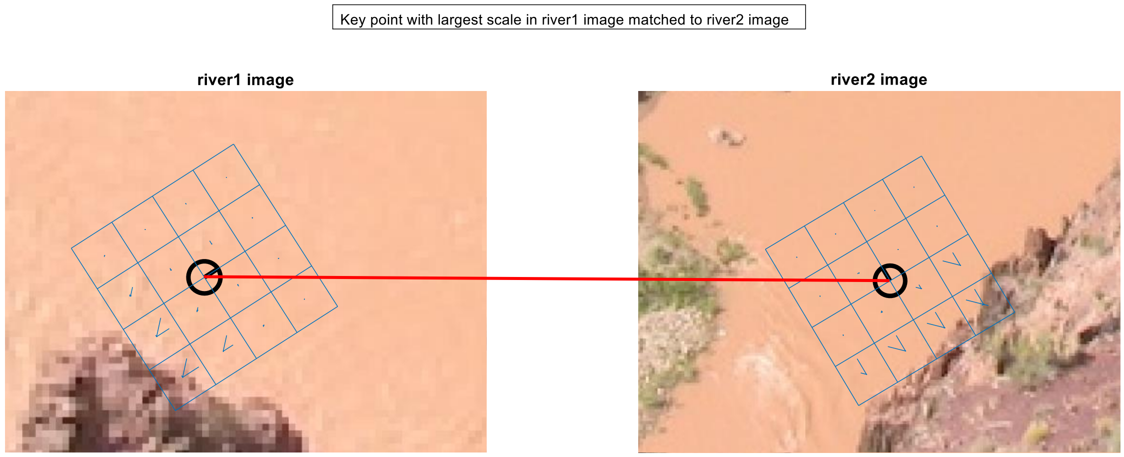
Parameters	river1 image	river2 image
Location	[292.7195 540.4269]	[407.8554 590.9029]
Scale	2.4165	5.1302
Orientation	-2.1409	-3.6799

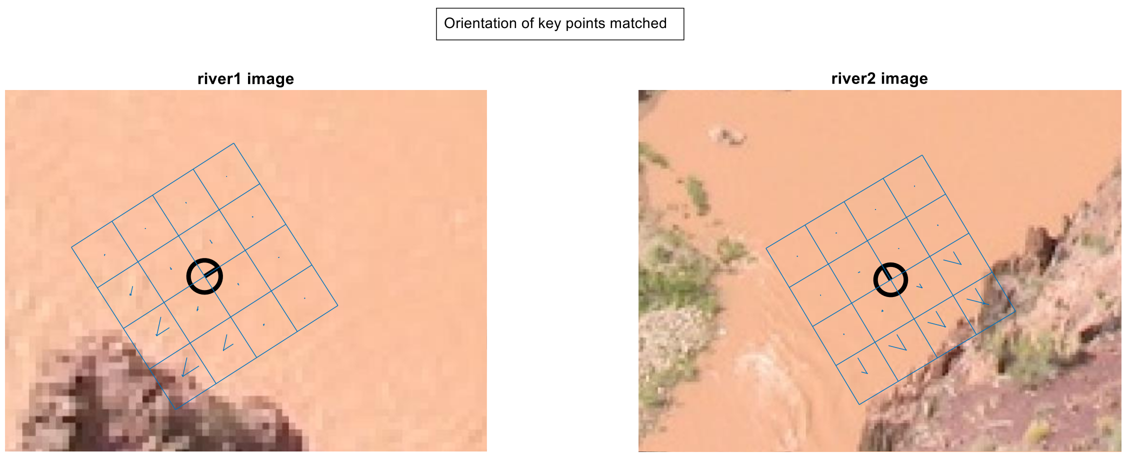
2. From the above table and the image shown below



It is evident that even though both key points are identified at different scales, position and orientation, yet they are matched because of the invariance of SIFT to scaling, rotation and translation geometric modifications

3. The below image displays the zoomed version of the output result, to capture the canonical orientation witnessed by the matched key-points.





Even if the orientation of the matched key-points are different, but their relative16*16 neighbourhood which includes 8 histogram bins of 4*4 samples are similar. Thus, descriptors play a vita role in matching the interest points and defining each key point uniquely and relatively.

4. The Euclidean distance between the matched key points is 281.5599.

5. Computational complexity of finding a good match is high because computation of nearest neighbour for each training image using euclidean distance over all the feature vector of test image would be time consuming.

6. Speed up robust features (SURF) can be used instead of SIFT to speed up the process of extraction of features.

7. To speed up the process further, 128D descriptor/feature vector can be projected to a lower dimension using Principal Component Analysis, so that the computational cost of evaluating the Euclidean distance could be minimized.

8. Computational complexity can be even reduced by using best-bin-first algorithm.

(c) Bag of Words

2.c.1 Abstract and Motivation

Bag of words model is applied to image classification problem, by treating image features as words. A (training) image is subdivided into smaller images and is contained inside dictionary. Bag of visual words is a vector of occurrences of a vocabulary of local image feature. Image features are used to construct vocabularies and represent each image as frequency histogram of features. Bag of Words is widely used for image retrieval and object recognition.

2.c.2 Approach and Procedure

Bag of Words can be implemented using various methods like Sliding window, Regular grid (partition entire image into dictionary), Interest point detector (using SIFT).

The following steps are essential:

1. Feature Detection: is performed using SIFT
2. Feature Description/Representation: filter bank response, image patches, SIFT descriptors.
3. Codebook generation: A codeword can be referred as a representation of similar patches. K-means clustering can be performed over all feature, with centroids of clusters as codewords.
4. Any (test) image can be mapped to different codewords yielding histogram of codewords.
5. To find which class (C_i) the test image (X) belongs to, intersection of histogram is computed for each class as follows:

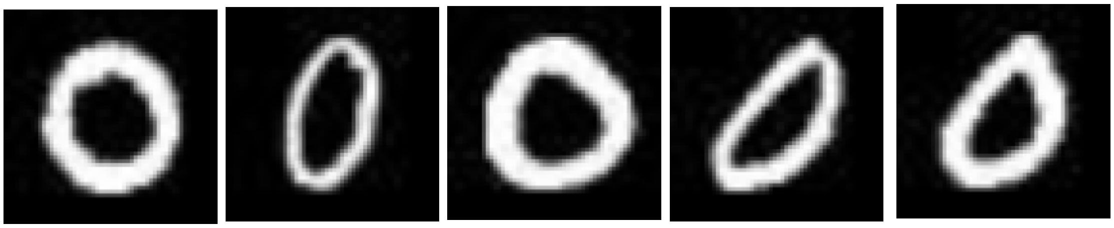
$$X \in C\,j: \quad j = \text{argmax}\left(\sum_{i=1}^{\text{feature\_size}} \frac{\min(Xi, Cji)}{\max(Xi, Cji)} \right)$$

Algorithm:

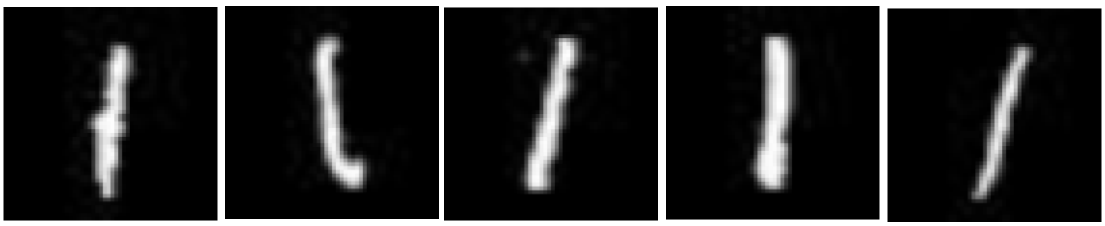
Step 1: Since MATLAB was unable to extract SIFT features, all the images were resized to 56*56
Step 2: Extract SIFT features for scaled images of one1~5, zero1~5
Step 3: Stack all the descriptors of one1~5, zero1~5
Step 4: Apply k-means clustering to the stacked descriptors for 2 class
Step 5: Obtain centroids for 2 class
Step 6: Extract SIFT features for eight
Step 7: Calculate the distance between each descriptor of ‘eight’ and each centroid of 2 class
Step 8: If the descriptor of ‘eight’ is closer to centroid of zero1~5 class, assign ‘label’ as 0
Step 9: If the descriptor of ‘eight’ is closer to centroid of one1~5 class, assign ‘label’ as 1
Step10: Plot the histogram of ‘label’ assigned to descriptors of ‘eight’

2.c.3 Results

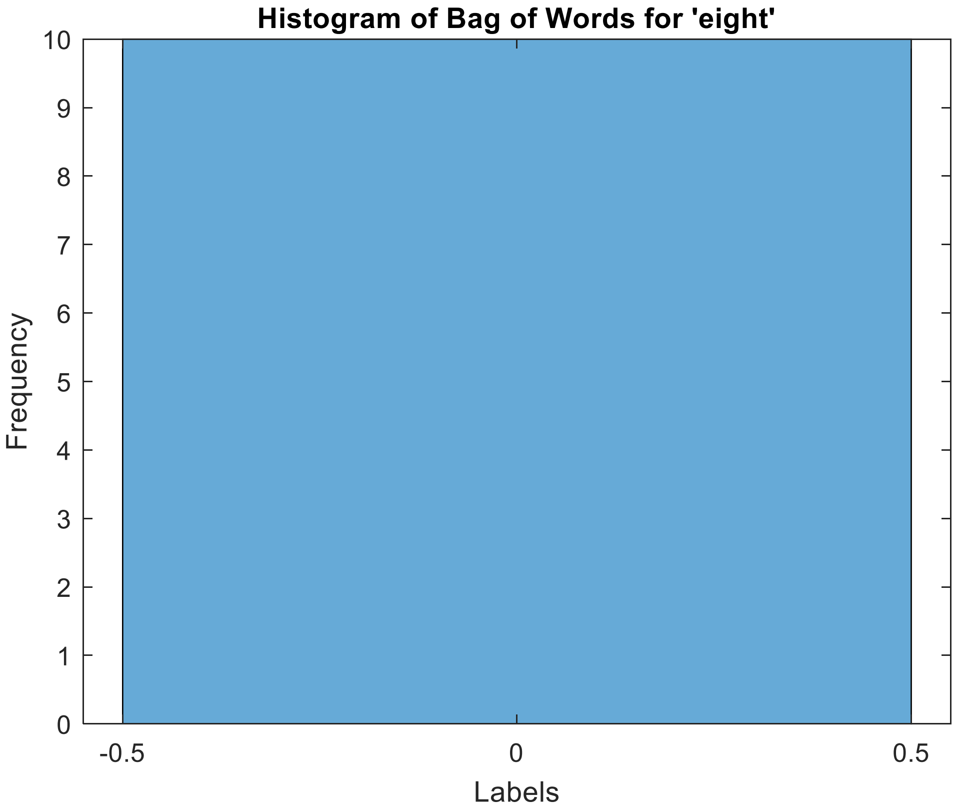
Training Images zero1~5



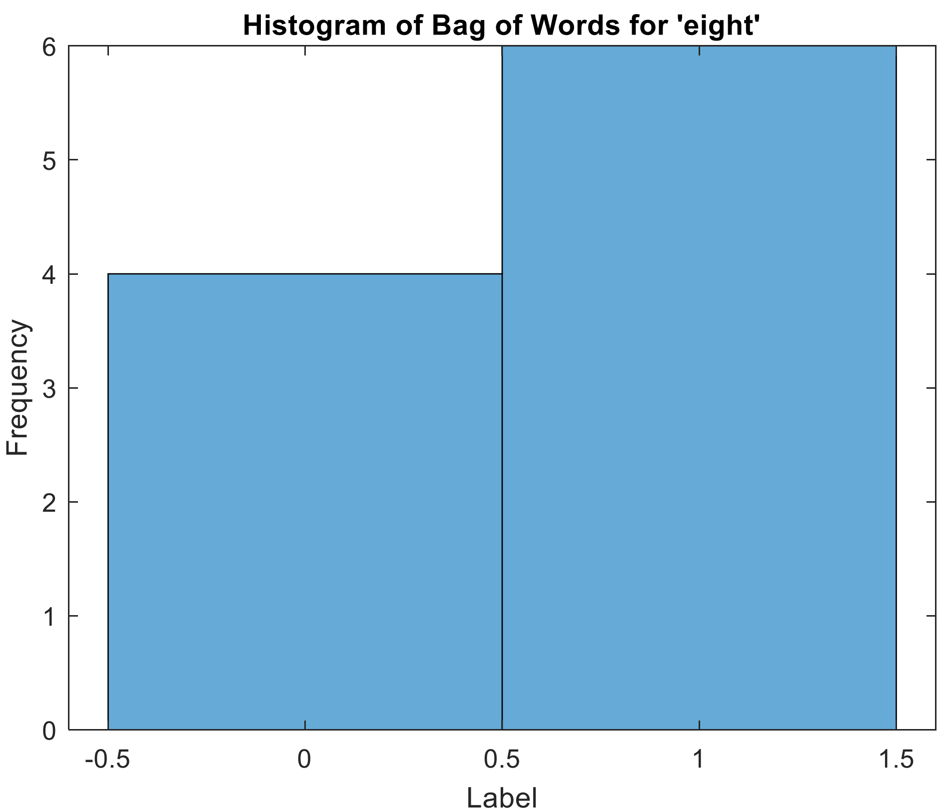
Training Images one1~5



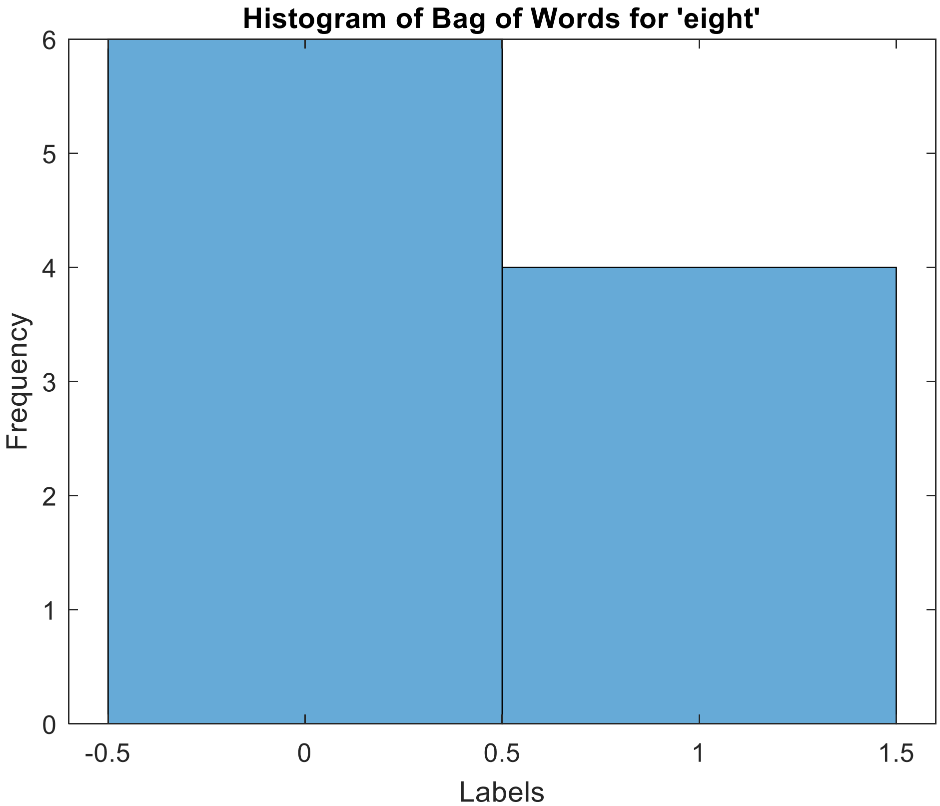
Test Image 'eight'

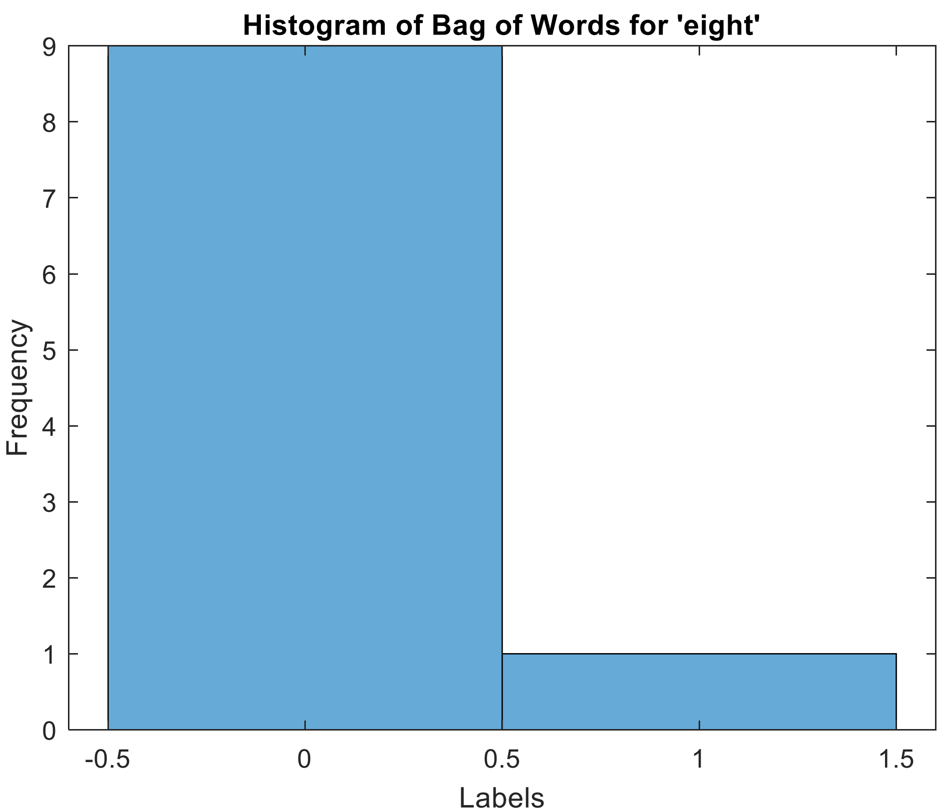


Histogram of Bag of Words for 'eight'
Frequency
Labels
10
9
8
7
6
5
4
3
2
1
0
-0.5
0
0.5



Histogram of Bag of Words for 'eight'
Frequency
Label
6
5
4
3
2
1
0
-0.5
0
0.5
1
1.5


Histogram of Bag of Words for 'eight'
Frequency
Labels
0
1
2
3
4
5
6
-0.5
0
0.5
1
1.5


Histogram of Bag of Words for 'eight'
Frequency
Labels
0
1
2
3
4
5
6
7
8
9
-0.5
0
0.5
1
1.5

2.c.4 Discussion

1. Total of 91 descriptors were detected and used for training (inclusive of zero1~5 and one1~5 descriptors).
2. 10 features was detected from ‘eight’ for testing.
3. The above histogram predicts that the labels are not unique each time the process is initiated, since k-means is used for clustering, its random initialization of centroid keeps changing.
4. From, the results it can be inferred that ‘eight’ belongs to ‘zero’ class since out of 5 trials 4 times ‘eight’ belongs to ‘zero’. From, visual perception as well, ‘eight’ has similar structure as ‘zero’.
5. The quality of the image determines distinctiveness of the features to be extracted.
6. If the number of test data and the number of clusters is increased, the process is computationally expensive to determine the Euclidean distance to each cluster centroid for each testing image to plot histogram. Computational complexity can be reduced if descriptors of 128D can be projected to a lower dimension using Principal Component Analysis.
7. Since SIFT is used for feature extraction, it is invariant to scaling, rotation and translational geometric modifications, yielding better result as spatial relation is made use by SIFT descriptors, features are distinct while bag of words using sliding window, regular grid are usually rotation variant and spatial relation is not explored much.