Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

Problem 1: Texture Analysis

1. Abstract and Motivation

Many portions of images of natural scenes are devoid of sharp edges over large areas. In these areas, the scene can often be characterized as exhibiting a consistent structure analogous to the texture of cloth. Image texture measurements can be used to segment an image and classify its segments. Texture is the description of spatial arrangement of intensities in an image or a region of image. The notion of texture appears to depend upon three varieties:
(i) Some local 'order' is repeated over a region which is large in comparison to the order's size.
(ii) The order consists in the nonrandom arrangement of elementary parts and
(iii) The parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.

Signal Processing algorithms use texture filters applied to the image to create filtered images from which texture features are computed. One of these texture filters is Laws filters as shown in the figure1

Name	Kernel
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]

Figure 1: Laws filters kernels

Based on the 1D kernels shown in figure 1, 5x5 Laws filters are constructed by tensor product of each of them. The example for filter E5L5 is shown in figure 2.

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Figure 2: Example of E5L5 kernel

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

2. Approach and Procedure

(a) Texture Classification

For texture classification, the effects of different Laws filters is shown in the figure 3.

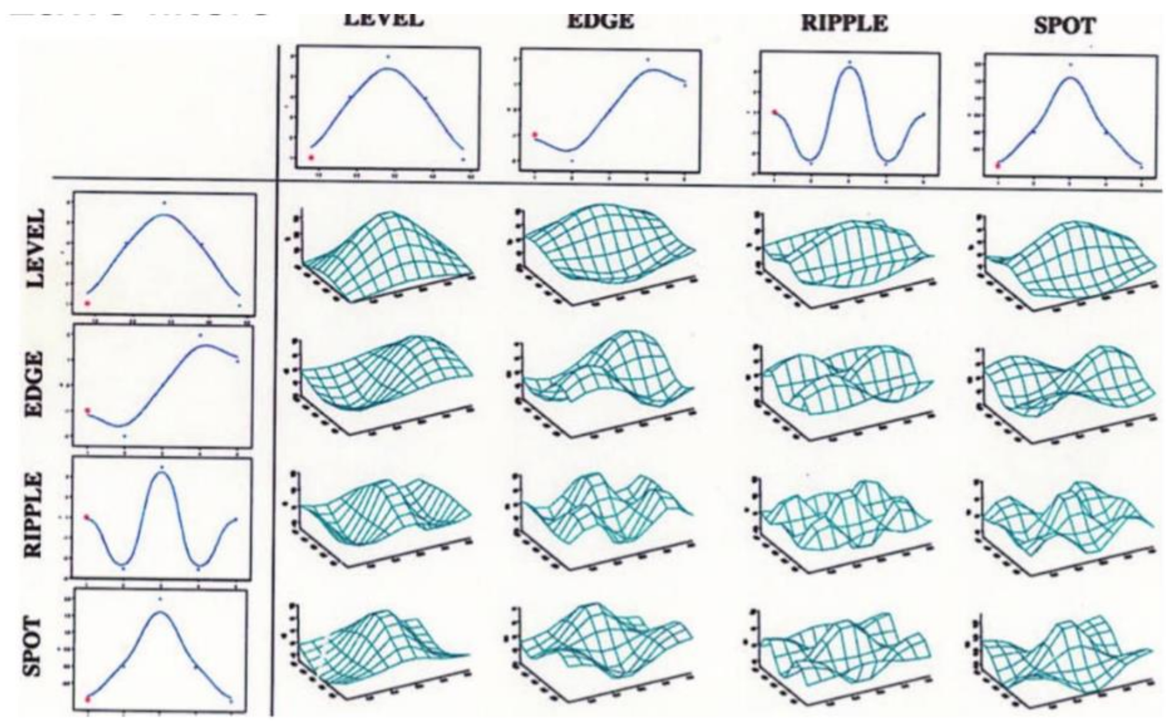


Figure 3: Effects of Laws filters

As shown above, each of 1D kernels have specific applications:
(i) L5 gives a center weighted local average (Gaussian)
(ii) E5 responds to row or col step edges (Gradient)
(iii) S5 detects spots (LOG)
(iv) R5 detects ripples (Gabor)
(v) W5 detects waves

After finding 25 kernels corresponding to tensor product of each of five 1D kernels with each other, the input images are first subtracted with the mean image value to reduce illumination effects. The mean reduced images are convolved with 25 Laws filters to obtain 25 filtered images for each of the input image. The average energy of each of the filtered image was computed as

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

$$1/N\sum|F(i,j)|^2$$

where N is number of pixels in the image.
In this way, 25 energy values are obtained and formed into a 25D vector. This process is done repetitively for all the input images. The flowchart of this procedure for 9 Laws filters is shown in figure 4.

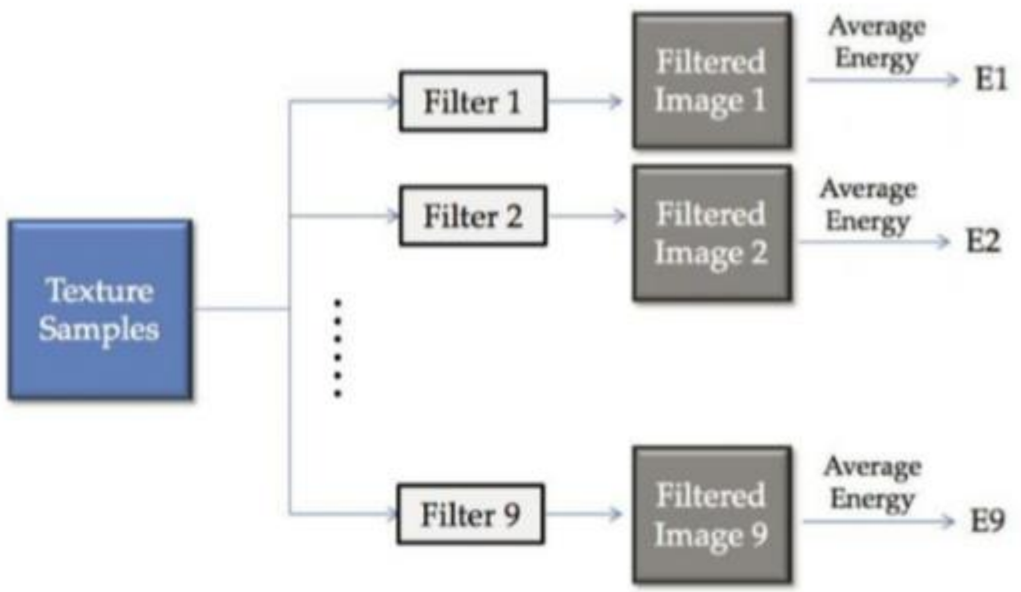


Figure 4: Flowchart to find vector for each texture sample

Further, because of having obtained similar features that differ in horizontal and vertical directions for the symmetry of pairs of 1D kernels, a 15D vector was formed by replacing each of the pairs with its average value as shown below:

L5L5	L5E5/E5L5	E5S5/S5E5
E5E5	L5S5/S5L5	E5W5/W5E5
S5S5	L5W5/W5L5	E5R5/R5E5
W5W5	L5R5/R5L5	S5W5/W5S5
R5R5	W5R5/R5W5	S5R5/R5S5

After obtaining the 15D vector, features were normalized in each of the axes corresponding to all the features by subtracting with mean and dividing it over by standard deviation of value in each dimension. Because the data points are sparse in higher dimensions and computation also becomes expensive for higher dimension, feature vector needs to be reduced to lower dimensions. Possible approach is feature

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019
selection using forward or backward selection algorithms. But, this is unsupervised case. Hence, PCA was made use of for reducing the dimensions.

PCA is a dimension reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information in the large set. It transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components. First principal component accounts for as much of the variability in the data as possible. Subsequently, each succeeding component accounts for as much of remaining variability as possible. This is achieved by following singular value decomposition of the feature space from which its eigen values corresponding to the right and left singular vectors are found. Reduction to k dimension space is performed by keeping significant k singular values. The reduction from 3D to 2D is shown in the figure5 below.

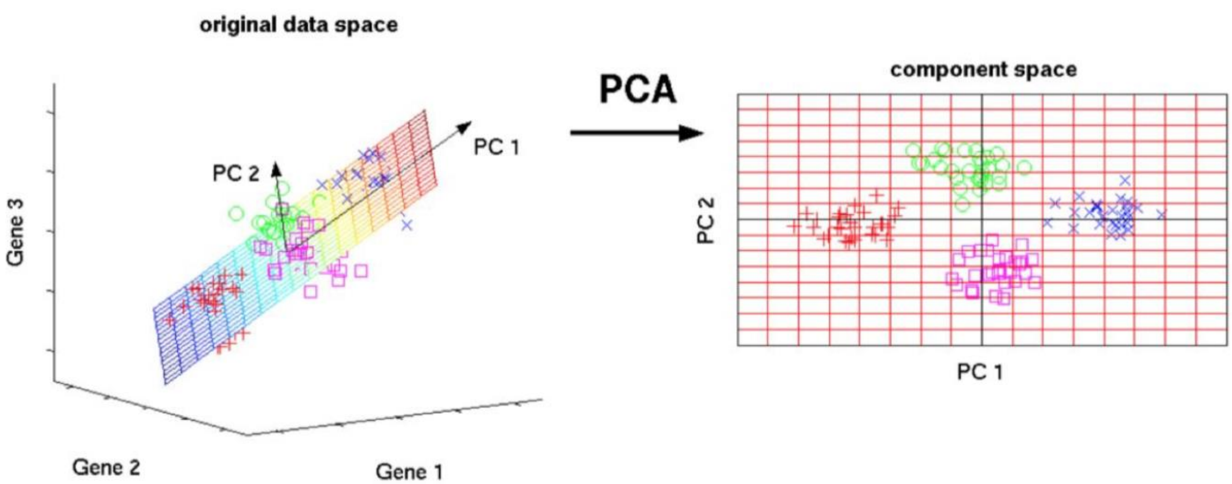


Figure 5: PCA reduction from 3 feature axes to 2 principal axes

Steps to perform PCA:
(i) Standardize the scale of the data, which is already done by subtracting mean and diving by standard deviation.
(ii) Perform SVD to find U, D, V where U contains left singular vectors in column wise and V contains right singular vectors in row wise and D is a diagonal matrix containing eigen values such that original data matrix = U * D * V
(iii) To reduce to k dimensions, k eigen values has to be kept. Hence, projecting it to R^k is obtained as:

$$\sum_{i=1}^{k} \sigma \vec{u}_i \vec{e}_i^T$$

where σ corresponds to eigen values and $\vec{u}_i$ is corresponding left singular vector and $\vec{e}_i^T$ is unit vector in R^k
(iv) The obtained matrix is reduced to k dimensions.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019
In this problem, features were reduced to 3 dimensions from the existing 15 dimensions using PCA. Further, the features were observed in the 3 dimensions as to how close is every feature with every other feature. For this purpose, k means clustering was used to cluster the data points that are close to each other. The closeness is defined by Euclidean distance in this problem. The k means cluster is based on the following algorithm:
(i) Initialize cluster centroids as $\mu_{1,}$ $\mu_{2,}$ $\mu_{3,}$ …, μ_k as random centroids.
(ii) Repeat the following until convergence i,e. until change in centroids are effectively negligible:

For every sample $x^{(i)}$, update the centroid assignment for each sample as:

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

For every cluster centroid $c^{(i)}$, update cluster centroid coordinate as:

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}$$

The flowchart and illustration of k means is shown in figure 6 and figure 7 respectively.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

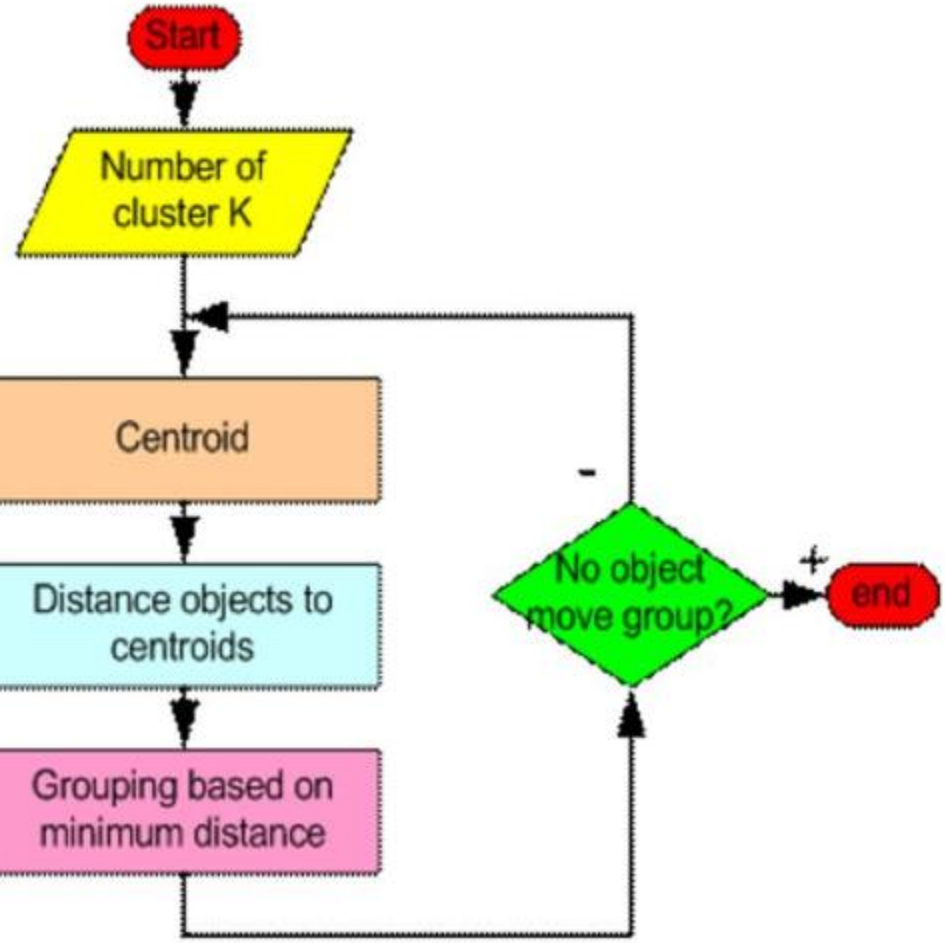


Figure 6: Flowchart for implementing kmeans clustering

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

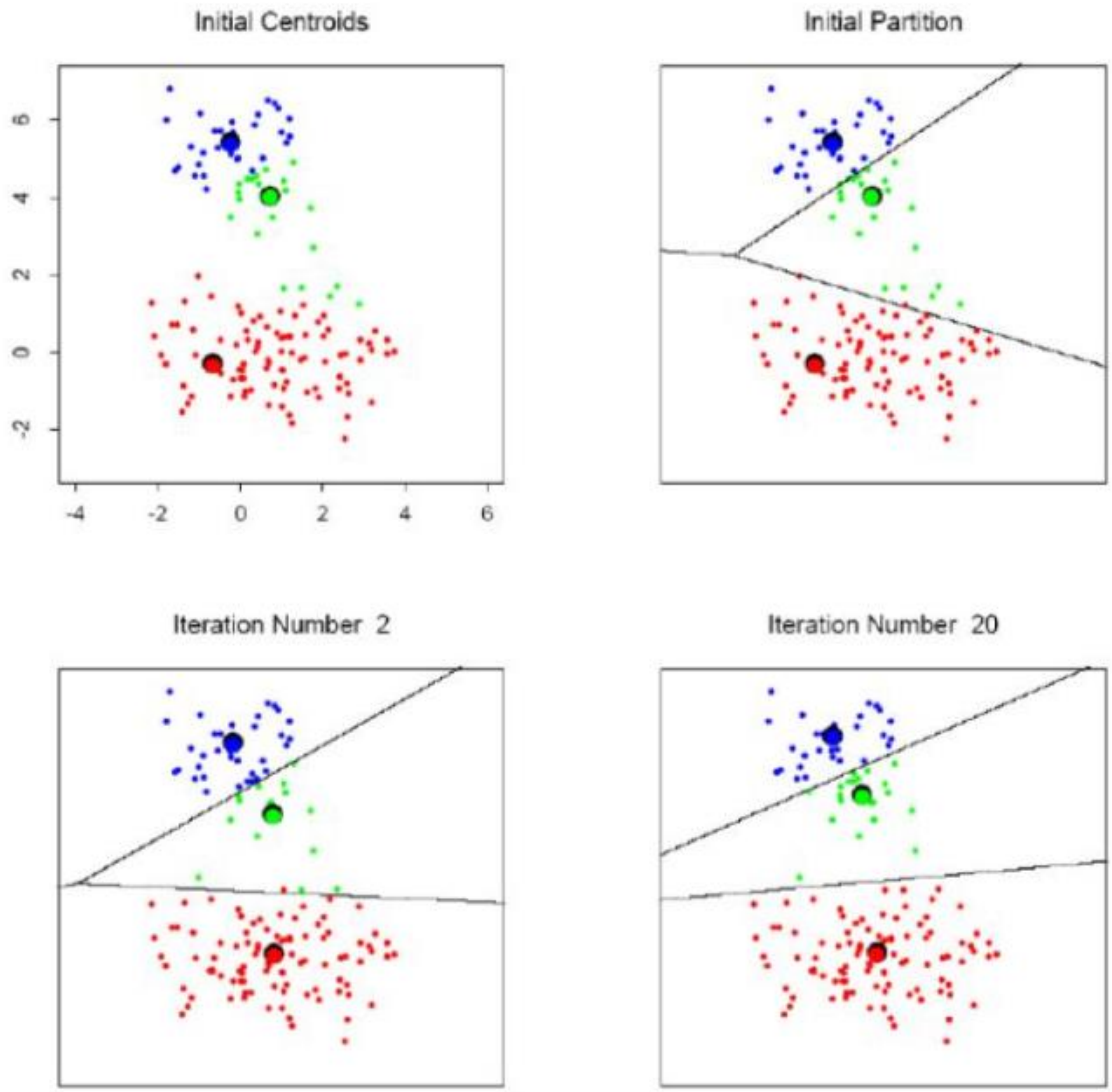


Figure 7: Illustration of kmeans clustering

The clustering was implemented for number of clusters=4 to the reduced feature in this problem.

Procedure:

- The input image 'pattern1.raw' is read byte by byte into a 2D array of size 128x128.
- Mean image value was subtracted from each pixel.
- 25 Laws filters were calculated using five 1D kernels.
- For each filter, the input image was convolved with these filters and the energy was obtained for each convolved image making a 25D vector as explained above. Neighbor padding was used as boundary extension technique.
- Each vector was reduced to 15D by replacing each pairs of kernels with its average as explained above.
- The above procedure was repeated for pateern2.raw through pattern12.raw images.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

- All the 25D features are concatenated to form a (12,15) feature space.
- Feature normalization was done for this feature space by subtracting with mean and dividing by standard deviation for each dimension.
- PCA was carried to reduce the feature space to 3 dimensions.
- Kmeans clustering algorithm was used to update the class labels for each of the texture samples.
- Results were observed and analyzed.

(b) Texture Segmentation

Texture Segmentation is an extension for texture classification problem in a sense that different textures are considered in a single image. Hence, every single pixel value needs to be determined whether or not it falls in one of the given k textures. A similar approach as to Texture classification was followed. All the 25 Laws filters was applied to the image to obtain 25 filtered images. Then, a window approach was used to form a feature vector for each of the pixels. Energy value for each of the pixel was found by using the energy equation explained in the previous section for particular window with the pixel as center. This was found in every 25 filtered images obtaining 25D vector for every pixel. No PCA or other kinds of dimension reduction was performed to cluster the vectors to k groups. Kmeans was used to cluster them into 7 groups and each group was given a certain gray level value to analyze the output.

Procedure:

- Input image ‘comb.raw’ was read byte by byte into a 2D array of size 510x510.
- The 25 Laws filters were convolved with the input image to obtain 25 filtered images. Nearest neighbor padding was used as boundary extension technique.
- For each of the 25 filtered images, a neighbor window of 15x15 was considered to find energy value for the center pixel. This forms 25D vector for the particular pixel.
- This was performed for every pixel in the image obtaining 25D vector for every pixel.
- Feature normalization was done using L5L5 filter as it was not a useful feature for texture classification.
- K means algorithm was implemented to perform clustering of vectors into 7 groups. Each of the 7 levels was given 7 different gray level values to represent in the output.
- The output image was saved, observed and analyzed.

(c) Advanced Texture Segmentation Techniques

Since not good texture segmentation results were obtained in 1(b), few advanced techniques were employed to improve the results in 1(b). The sparsity of the feature space was reduced by using PCA. The PCA technique is similar to the one explained in the previous section. PCA for different number of components was checked. Further, a post processing technique to merge small holes was used. The technique used is a Morphological closing technique which is the result of dilation followed by erosion. Morphological closing is used to fill the holes in the image using a structure element that is matched for every pixel as a center pixel. This improves the readability of each regions in terms of colors represented. An illustration of closing is shown in the figure 8 below.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

Figure 8: (a)Image Before Morphological Closing *(b)Image After Morphological Closing*

Further, boundary of two regions were enhanced by using edge crispening techniques. For edge sharpening or crispening, first gaussian blur was calculated for the image using gaussian kernel of size 5x5. And, the original image was subtracted by this blurred image to get sharpened edges in the images.
Procedure:

- Input image ‘comb.raw’ was read byte by byte into a 2D array of size 510x510.
- The 25 Laws filters were convolved with the input image to obtain 25 filtered images. Nearest neighbor padding was used as boundary extension technique.
- For each of the 25 filtered images, a neighbor window of 15x15 was considered to find energy value for the center pixel. This forms 25D vector for each pixel.
- This was performed for every pixel in the image obtaining 25D vector for every pixel.
- Feature normalization was done using L5L5 filter as it was not a useful feature for texture classification.
- From the obtained feature space with 25 columns, the dimensions was reduced using PCA to 5 columns.
- K means algorithm was implemented to perform clustering of vectors into 7 groups. Each of the 7 levels was given 7 different gray level values to represent in the output.
- From the obtained segmented image, morphological closing was performed to merge holes in each segment using 5x5 ones matrix as structural element.
- Edge sharpening was performed on the image by subtracting it with its gaussian blurred form. Gaussian weights with 5x5 window was chosen.
- The output image was saved, observed and analyzed.

3. Observation and Discussions:

(a) Texture Classification
The energy vectors for each of the texture samples are as shown in the figure below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

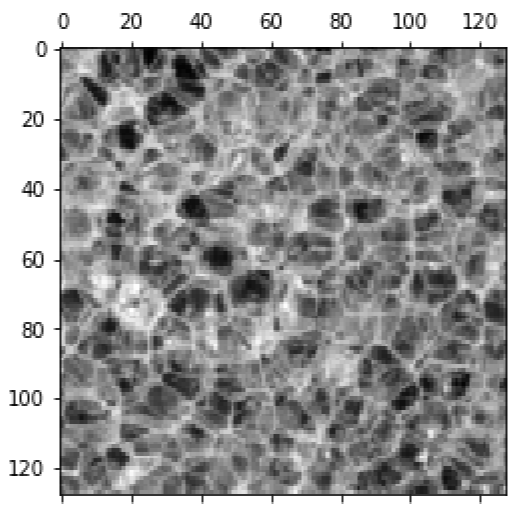
	l5l5	l5e5	l5s5	l5w5	l5r5	e5l5	e5e5	e5s5	e5w5	e5r5	s5l5	s5e5
0	3916.289	921.9019	437.5992	349.5112	548.0909	928.7536	223.5404	120.6561	105.855	175.1978	452.7578	124.0949
1	2594.815	982.963	665.823	682.1825	1318.949	460.182	190.4713	150.6587	177.0545	400.4208	199.3497	80.52581
2	2171.941	788.9244	502.808	471.5323	707.097	216.4353	59.01836	39.12633	42.11735	89.75449	127.3389	34.97542
3	4502.573	1339.36	715.0123	620.6013	1036.876	844.3194	247.7012	149.4102	142.8325	263.451	443.9063	134.1419
4	2031.659	630.2465	381.2578	357.3487	606.0182	602.1679	207.9023	138.1406	135.933	241.0626	363.132	131.3793
5	2246.106	750.6801	571.2279	624.8129	1107.103	253.1464	63.6052	41.33988	42.64618	87.52005	147.3341	37.17599
6	4251.515	1365.009	783.2253	702.2582	1168.25	903.346	268.0366	166.5265	161.1532	291.2865	488.5829	148.2178
7	7137.605	1984.161	999.6324	792.5194	1219.008	1890.448	485.9462	267.9067	235.5757	390.1654	965.8822	269.0163
8	6753.702	2209.515	1268.788	1125.039	1873.379	1430.185	431.5508	269.2702	260.0231	474.1546	769.5003	239.1213
9	3722.108	1043.049	525.3087	419.0709	645.8138	992.7114	254.8071	141.5562	124.7516	206.9862	510.3784	141.0458
10	3883.782	1224.676	737.9007	688.0338	1162.465	1153.66	407.4474	270.9018	265.1838	468.2207	678.6709	259.2516
11	4877.111	1579.949	1212.877	1314.583	2318.79	640.5317	169.1009	107.8845	111.7586	232.6003	354.1309	101.2696

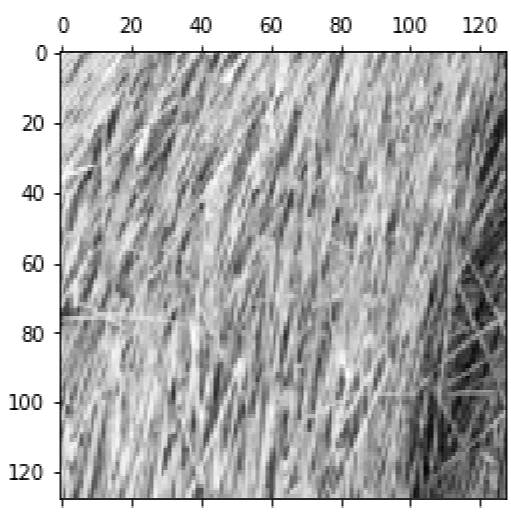
	s5s5	s5w5	s5r5	w5l5	w5e5	w5s5	w5w5	w5r5	r5l5	r5e5	r5s5	r5w5	r5r5
0	72.94143	67.13607	115.6625	361.714	111.7448	69.921	67.27964	120.9337	571.8787	191.8915	127.6847	130.6575	244.7124
1	65.6956	81.85942	205.1789	156.3651	58.73695	46.58415	58.8413	154.7656	241.8234	87.12972	66.88985	82.34253	215.2396
2	23.65677	25.81002	56.52183	128.0999	35.28348	24.18306	26.77813	59.07951	258.8076	72.96312	50.46594	56.64687	128.0041
3	83.16012	81.57326	155.7787	393.63	122.6993	77.18377	77.14974	150.7729	672.1841	215.8395	141.0308	146.3038	294.347
4	91.7018	94.36816	179.3561	335.5414	125.7506	91.82677	98.72862	198.532	642.6621	228.3137	175.2641	199.1102	417.4157
5	24.61023	25.63567	52.8876	143.6717	37.11116	24.88301	26.42882	55.34917	335.5737	75.16914	51.12981	55.7924	117.8139
6	93.94194	92.01544	170.2824	439.6246	136.8953	87.96753	87.25093	164.1672	805.3919	241.8908	158.784	162.4522	315.9775
7	158.2574	146.0989	253.9904	791.8511	244.8028	151.8558	145.7239	264.1155	1356.159	434.5364	283.4642	284.0392	533.2858
8	152.4883	149.5345	279.1316	691.9299	221.9888	143.2631	142.4351	271.7808	1243.029	396.8386	260.9464	268.5085	533.9127
9	83.31534	77.13376	134.3137	423.1612	128.802	80.25277	77.08306	139.2824	741.8153	229.3444	149.8861	150.0735	281.7064
10	181.6659	186.6891	352.38	622.5817	248.8711	182.7491	196.4699	392.9333	1125.54	448.3321	347.0516	395.7503	831.1139
11	66.41059	69.51199	145.1711	348.8849	103.2028	68.93917	73.17877	154.4843	736.6845	214.0823	144.7323	157.1581	331.7884

Based on variance calculated in each column, the maximum variance was obtained for L5L5(2.57e+06) which corresponds to least discriminant power as higher variability leads to poor clustering. Least variance was observed for S5S5(2.26e+03) and S5W5(2.12e+03) which leads to highest discriminant power.

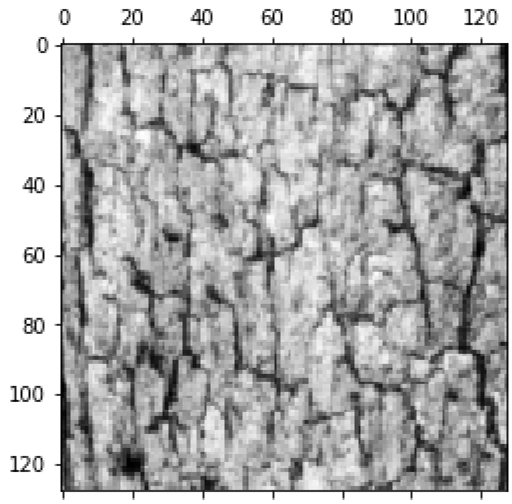
Cluster results for original 25D features are as shown below:

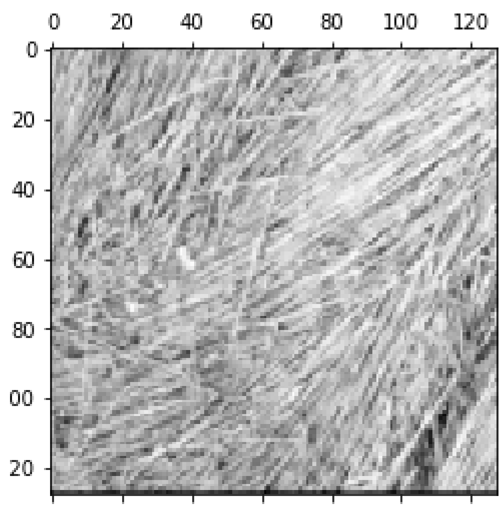
Cluster 0

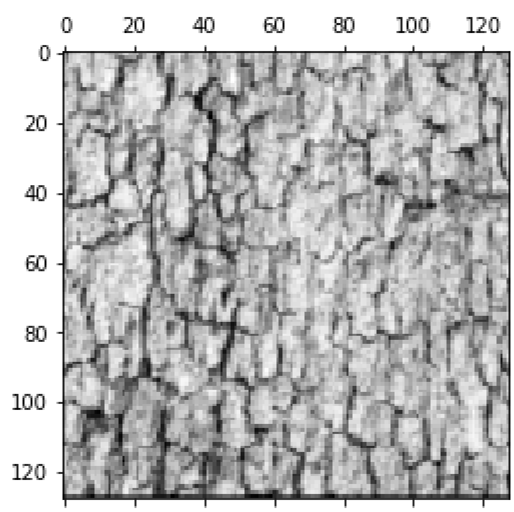




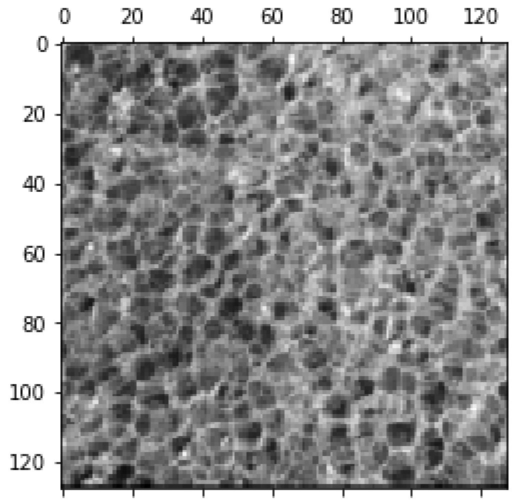
Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



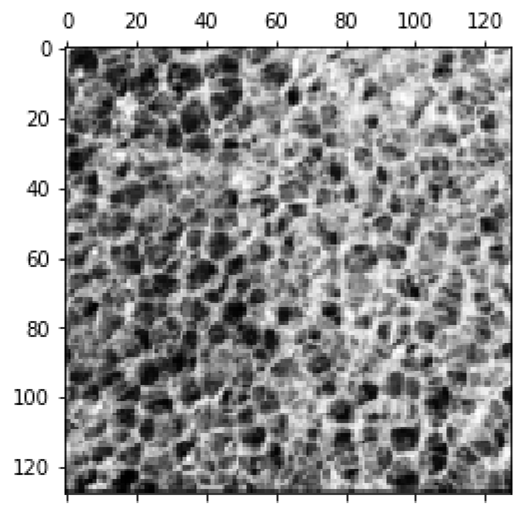


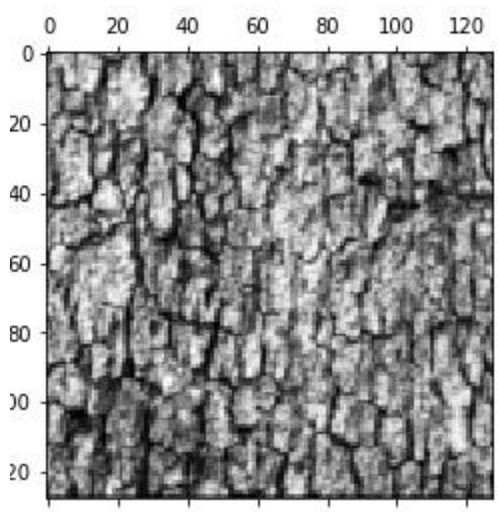


Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

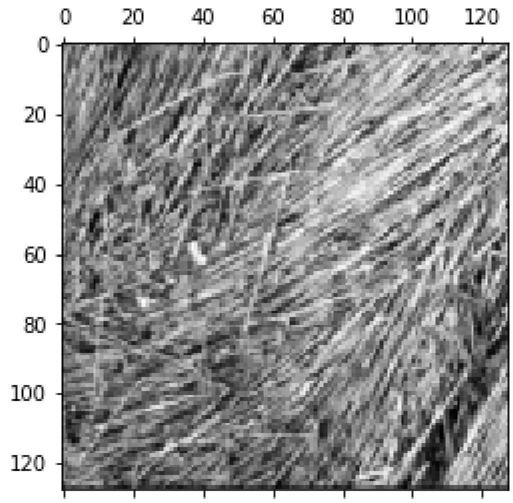


Cluster1

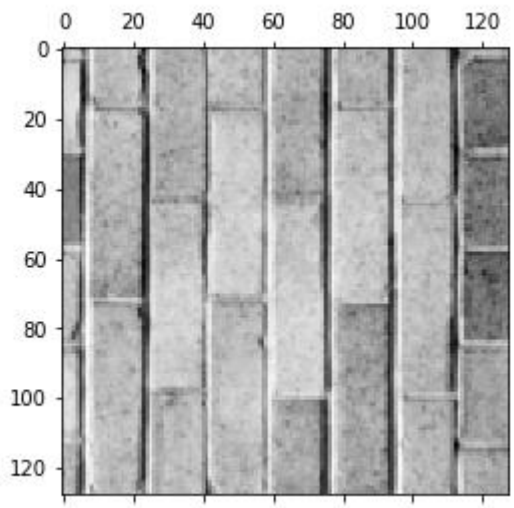




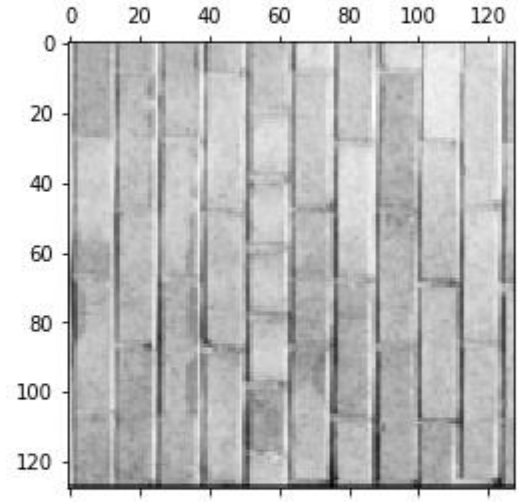
Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



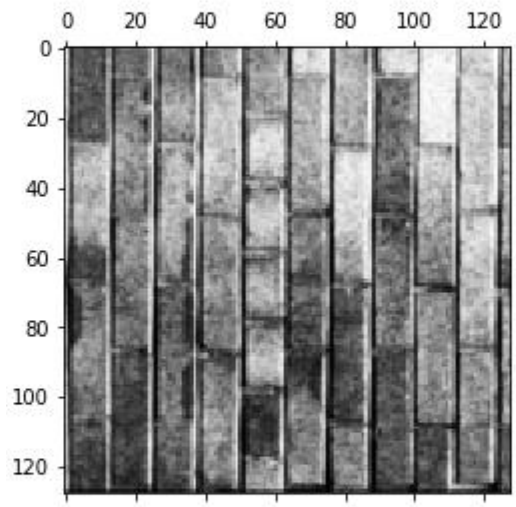
Cluster 2



Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

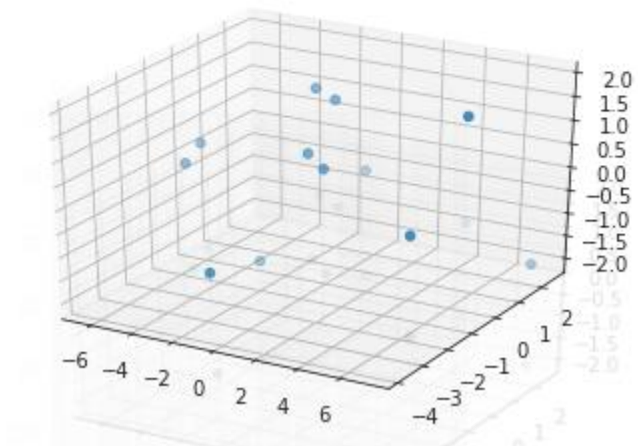


Cluster 3

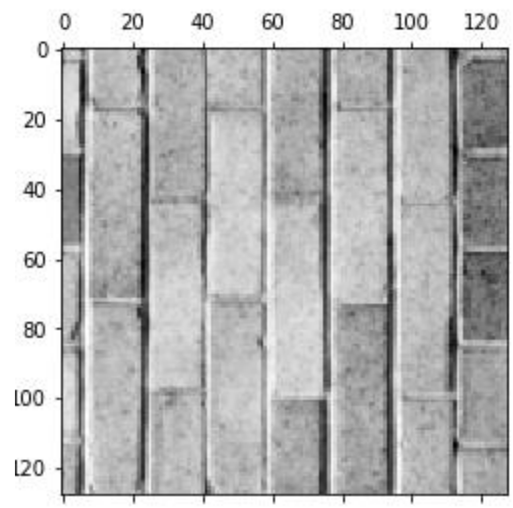


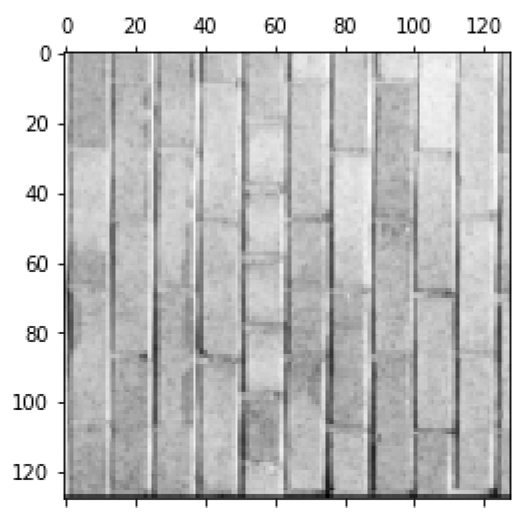
First, I used normalization for each dimension of feature vector and then took reduced components to 3. The result for dimension reduction and clustering are shown below. However, because of small dataset, the clustering algorithm couldn't perform well.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



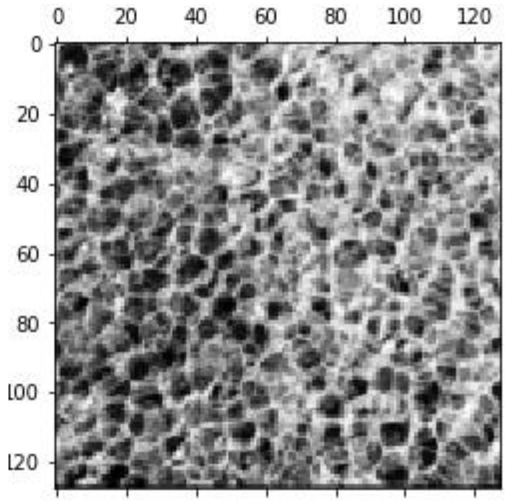
Cluster 0

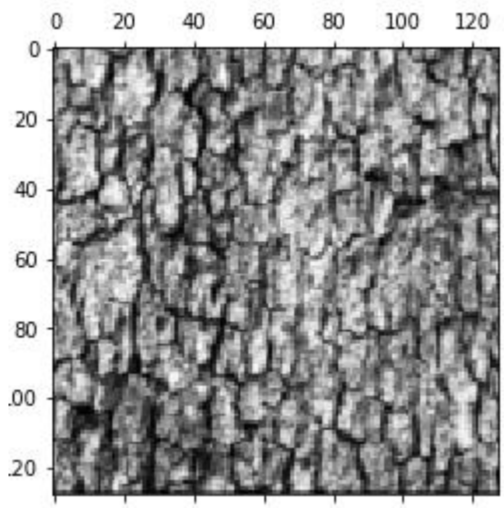


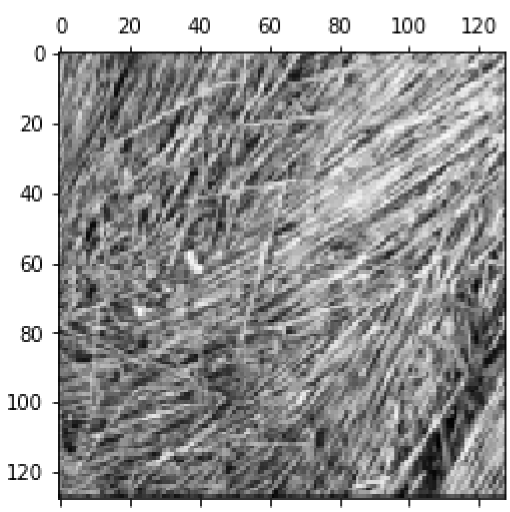


Cluster1

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

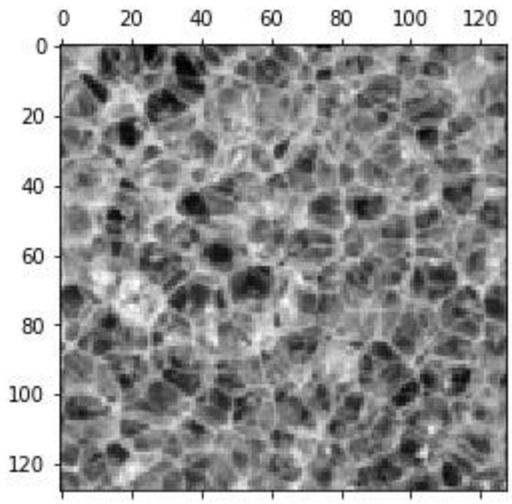


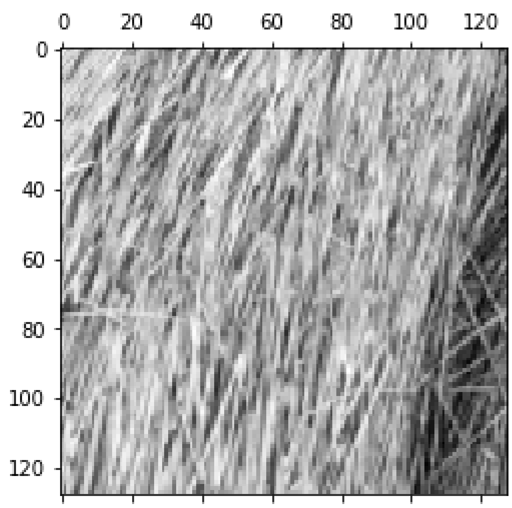


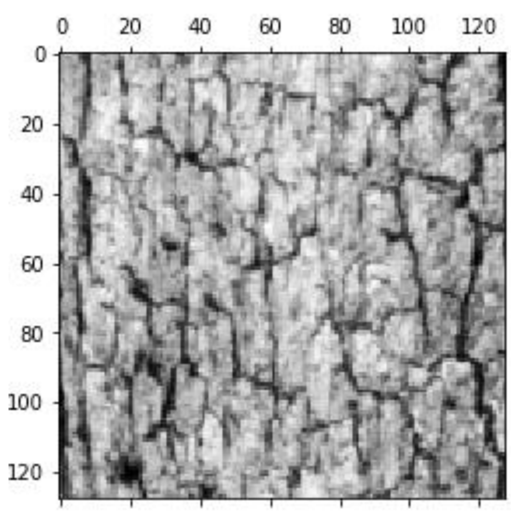


Cluster 2

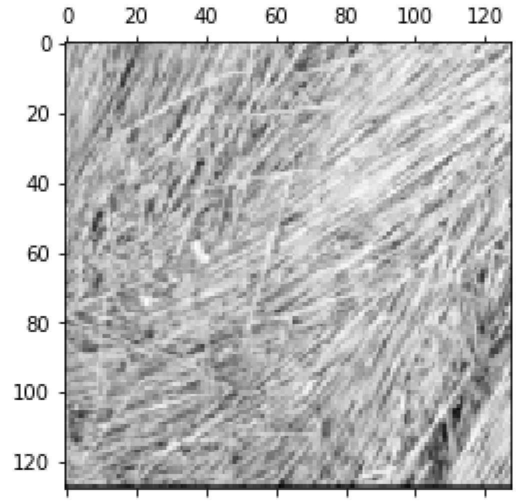
Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

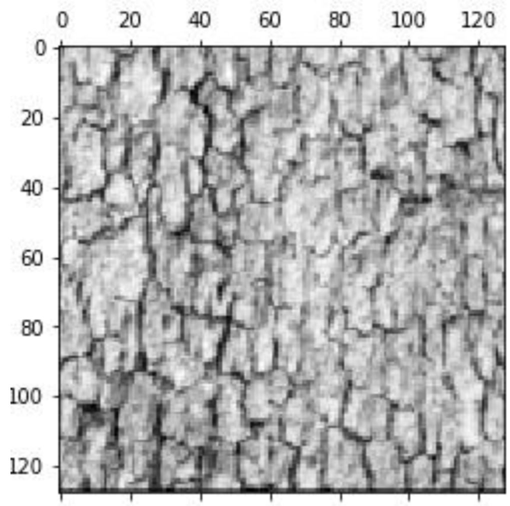


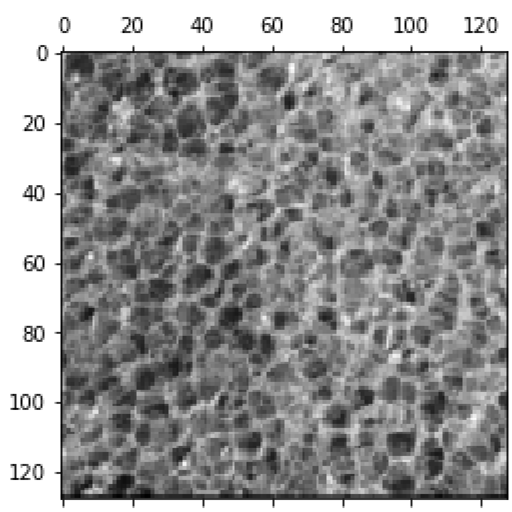




Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

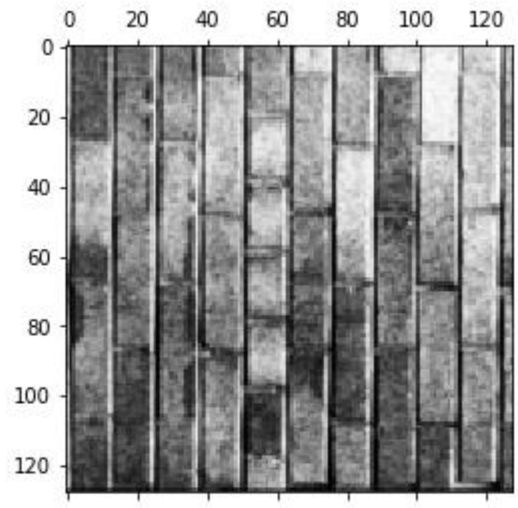




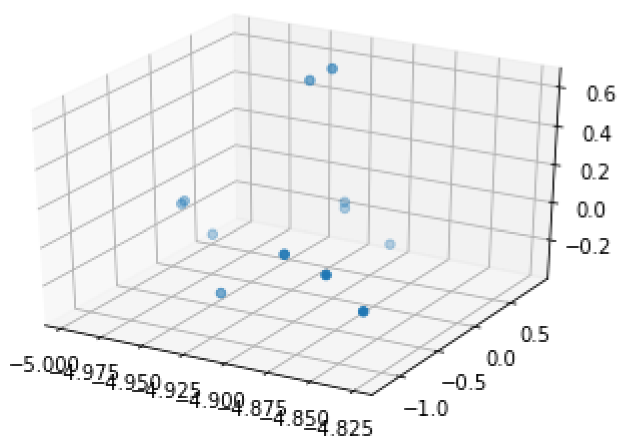


Cluster 3

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

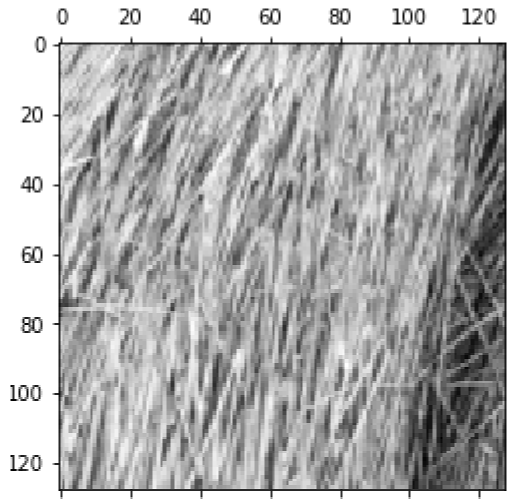


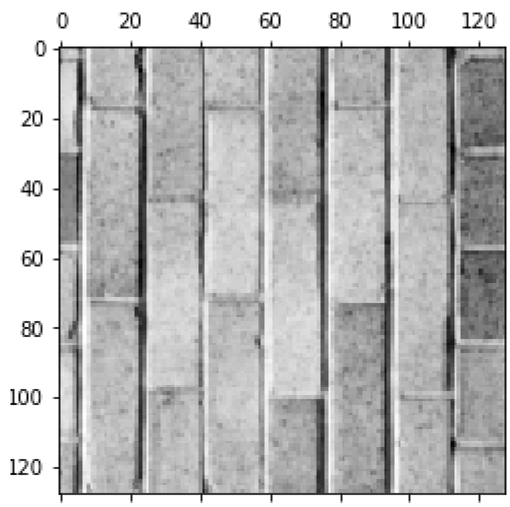
Different normalization techniques were tried to achieve better classification. Finally, normalization along the row of the feature space gave the best classification with just one mis classification error. The PCA reduced data plot and clustering results are as shown below:

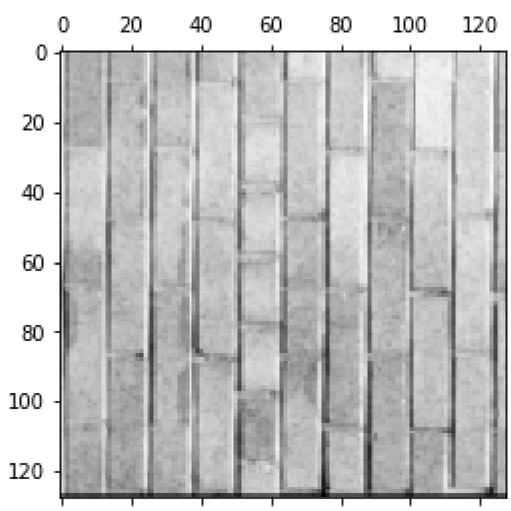


Cluster 0

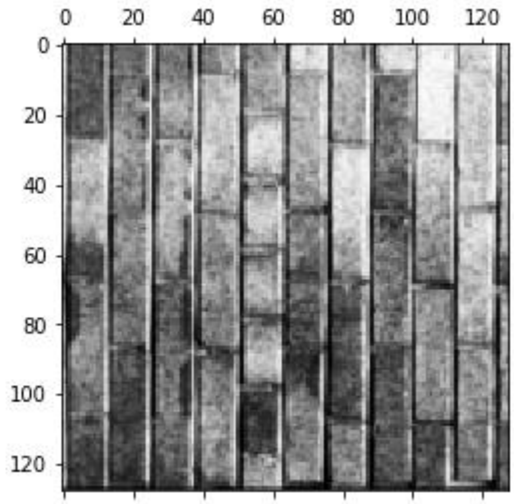
Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



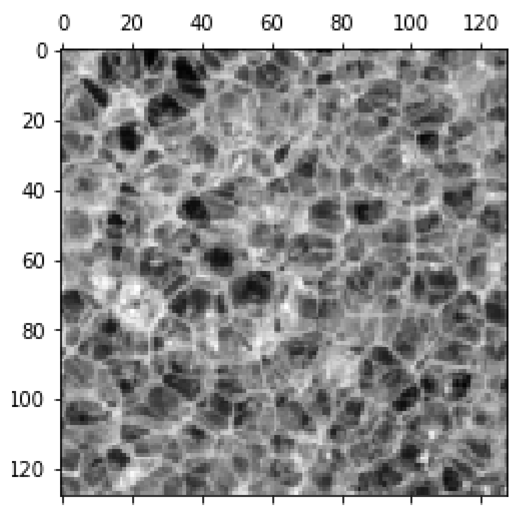


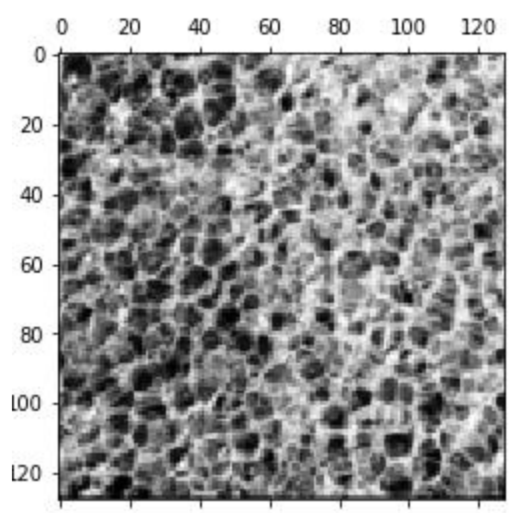


Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

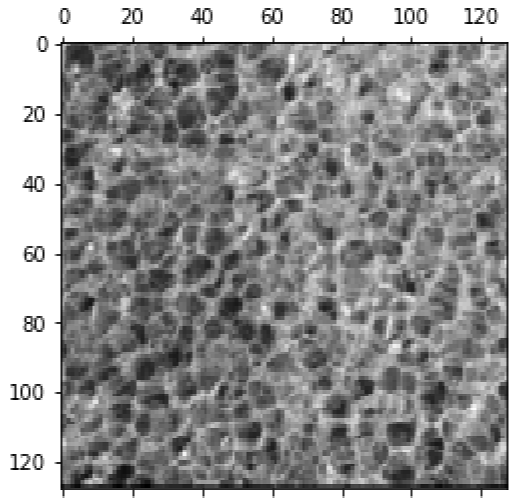


Cluster 1

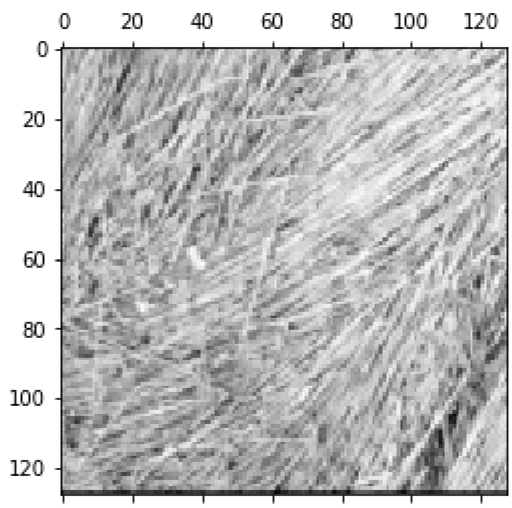


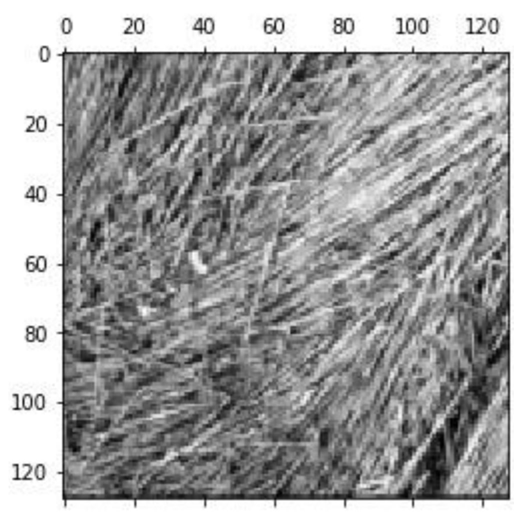


Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



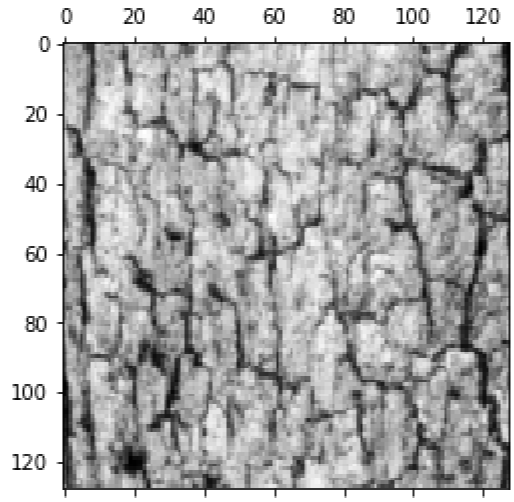
Cluster 2

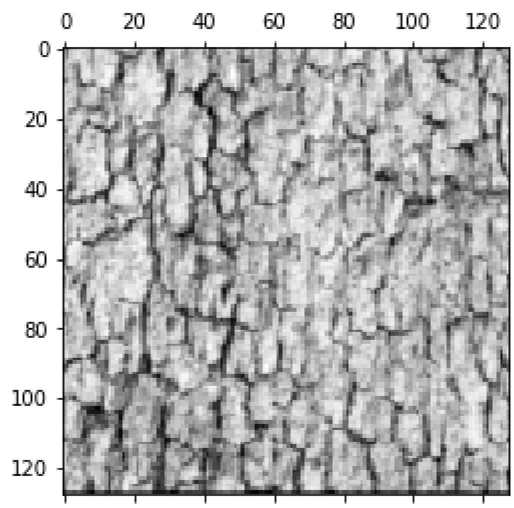


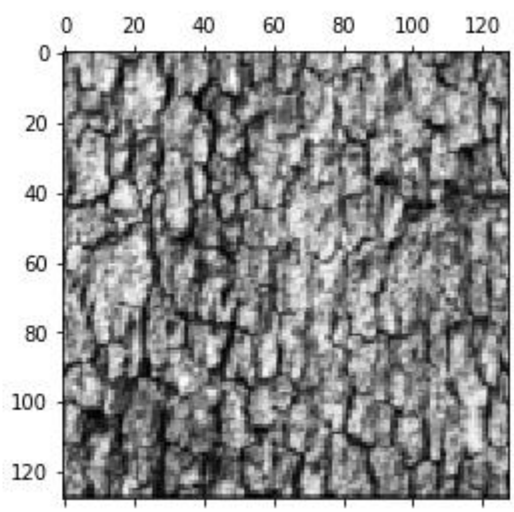


Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

Cluster 3





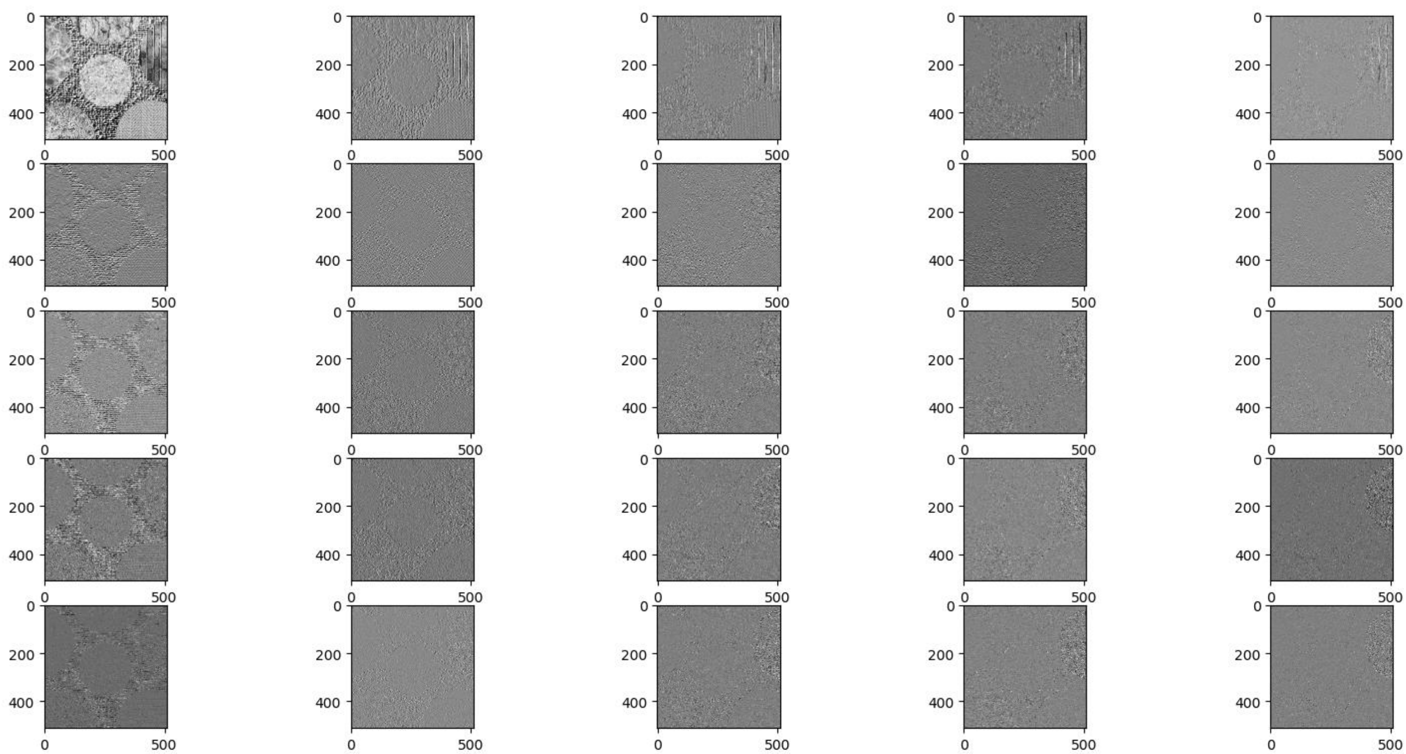


Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019
Comparison between clustering with 25D and 3D:

- Misclassification error is nearly the same as the data set is very small.
- Except for brick, all other textures are falling in wrong clusters because barks, bubbles and straws all have spots and give larger outputs for kernels involving S5
- Dimension reduction might be useful for larger data set.
- Computationally faster for 25D as it doesn't involve an additional step of PCA.
- K means performs better on 3D because of sparsity present in 25D.

(b) Texture Segmentation

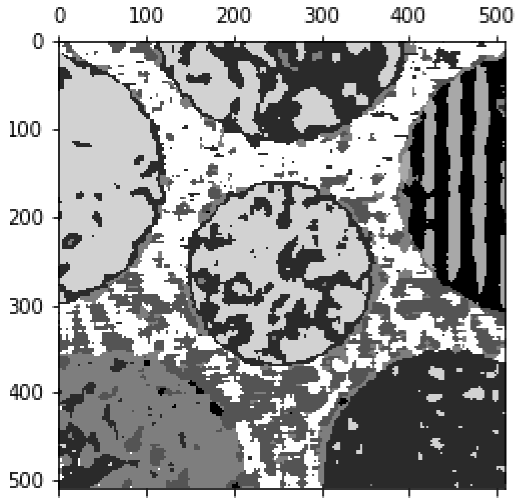
The texture segmentation output for each of the kernels is shown in the figure below:



- As can be seen, l5 l5 is just smoothing of image and other filters correspond to finding regions of edge(vertical), spots, wave and ripples.
- Energy corresponding to L5L5 hence is not useful and used for normalization.

The texture segmentation after kmeans clustering and assigning different colors to each of the clusters for window sizes of 13x13 and 15x15 are as shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



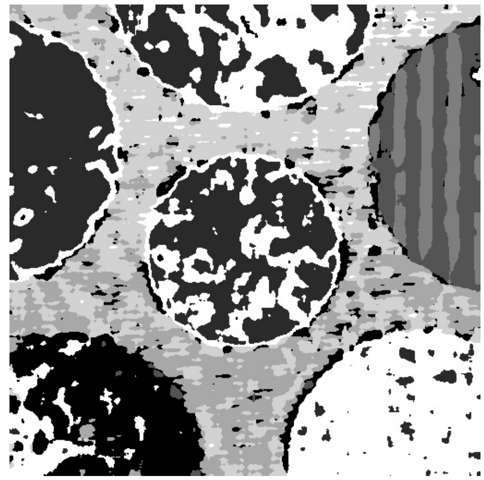


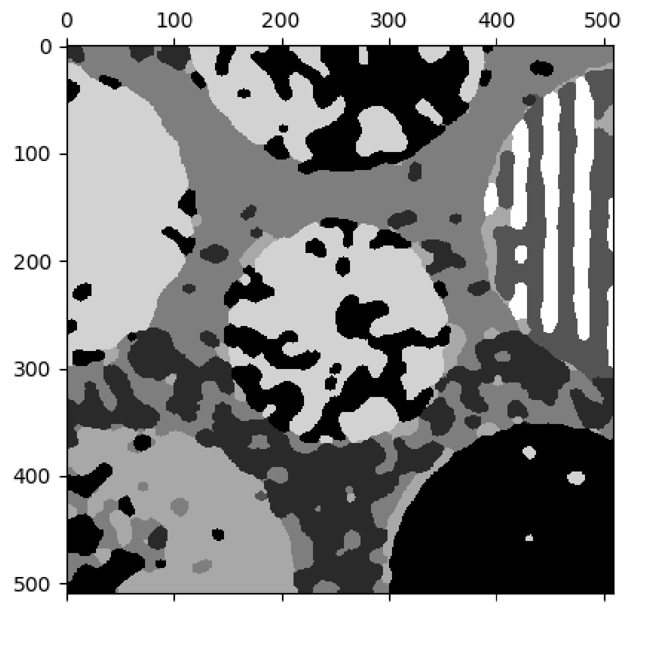
Figure 9(a) Segmentation with window size 13x13 *(b) Segmentation with window size 15x15*

- There is not much difference in using 13x13 and 15x15 windows.
- Dividing of regions is done correctly but there is more misclassification in the brick region because of vertical regions having different energies and hence falling in different clusters.
- Normalization was done using l5l5 energy values.
- Mean was subtracted from each window for reducing illumination effects which gave better results compared to subtracting by mean of entire image.
- Computationally slow for higher dimension images.

(c) Advance Segmentation Techniques

- First, PCA was implemented to reduce the number of dimensions to 3. The result is as shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



- Further, to merge small holes, initially the neighbors of a 15x15 window was considered and each pixel was replaced with the maximum occurrence of pixel in the neighborhood.
- Next, morphological closing was performed which is explaned in the procedure section. The results for closing iterations is shown below:

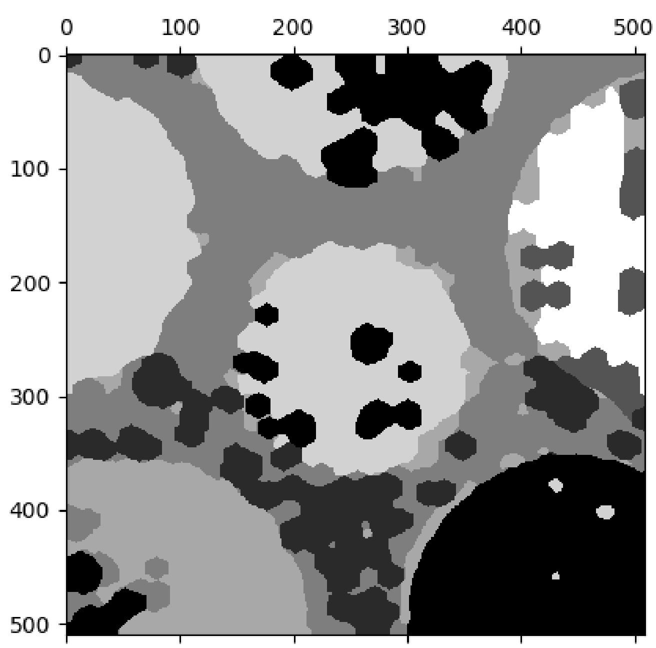


Figure 10: Closing after 5 iterations

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

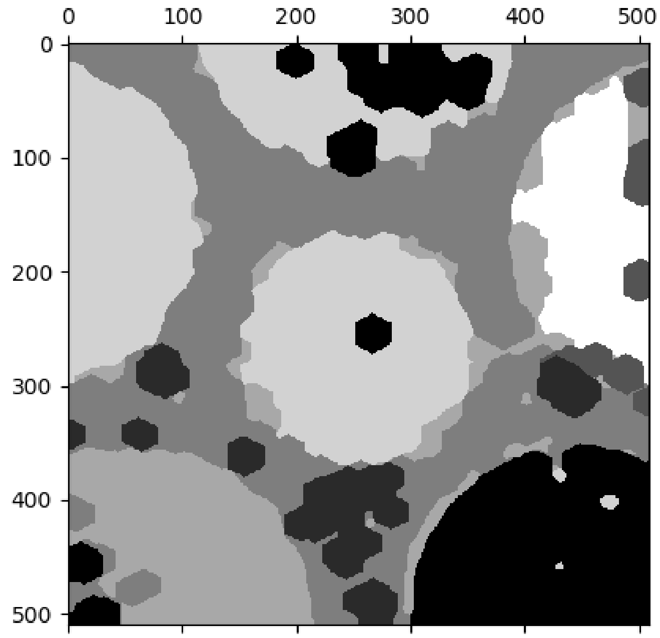


Figure 11: Closing after 7 iterations

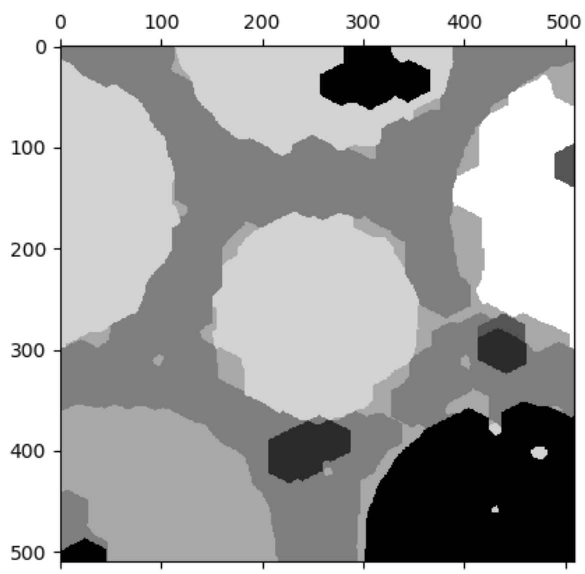


Figure 12: Closing after 10 iterations

- As can be seen, there is misclassification of the data in different clusters because of improper data.
- PCA gives better results than without reducing as shown in 1(b) because of cleaning of sparsity in the feature space.
- Also, cure of dimensionality affects the kmeans algorithm.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

- Computationally slow.
- The merging using morphological closing uses brighter colors as foreground and darker as background. Hence in the regions containing both the colors, with higher iterations brighter color is filled in the region as can be seen above where three regions are the same.
- Different techniques like recursive feature elimination/ forward selection/ backward selection can be used as techniques to reduce features and implement kmeans over that.

Problem 2: Image Feature Extractor

1. Abstract and Motivation

Object recognition in cluttered real-world scenes requires local image features that are unaffected by nearby clutter or partial occlusion. The features must be at least partially invariant to illumination, 3D projective transforms, and common object variations. On the other hand, the features must also be sufficiently distinctive to identify specific objects among many alternatives. The difficulty of the object recognition problem is due in large part to the lack of success in finding such image features. However, recent research on the use of dense local features has shown that efficient recognition can often be achieved by using local image descriptors sampled at a large number of repeatable locations.

Hence, David Lowe came up with a method called Scale Invariant Feature Transform (SIFT) which transforms image into a large collection of local feature vectors each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection.

2. Approach and Procedure

(a) SIFT

Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and motion tracking. This method describes image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise.

Following are the major stages of computation used to generate the set of image features:
1. Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation. Scale space of an image could be found by convolution of Gaussian of different sigma values with the input image. This can be achieved by convolving DoG with the image as shown in the equation below.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

The illustration for DoG is shown in figure 9.

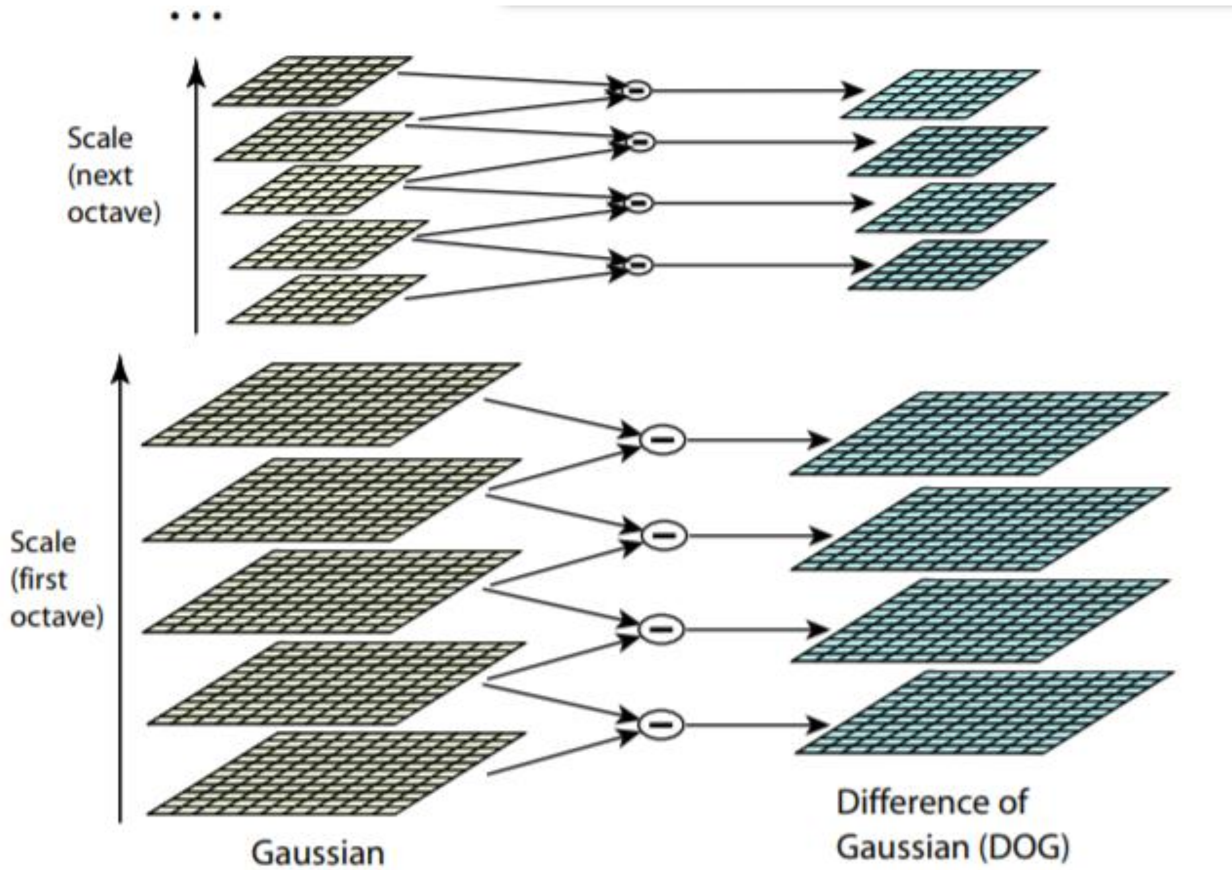


Figure 13: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

In order to detect the local maxima and minima of D(x, y, σ), each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below as shown in figure 10. It is selected only if it is larger than all of these neighbors or smaller than all of them.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

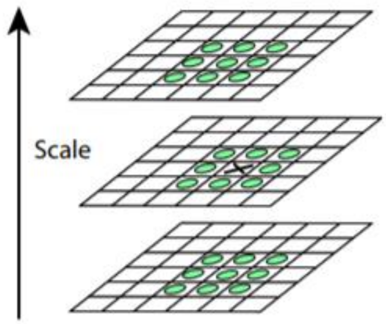


Figure 14: Detecting maxima and minima of DoG images

2. Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability. Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. DoG gives large values for pixels in the edges even though the edges are defined poorly. Hence these are eliminated using thresholding. Also, principal curvatures of each keypoint is found using Hessian matrix and further thresholding is done on the ratio of these curvatures to eliminate more keypoints. The illustration of keypoint selection is shown in figure 11 below.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

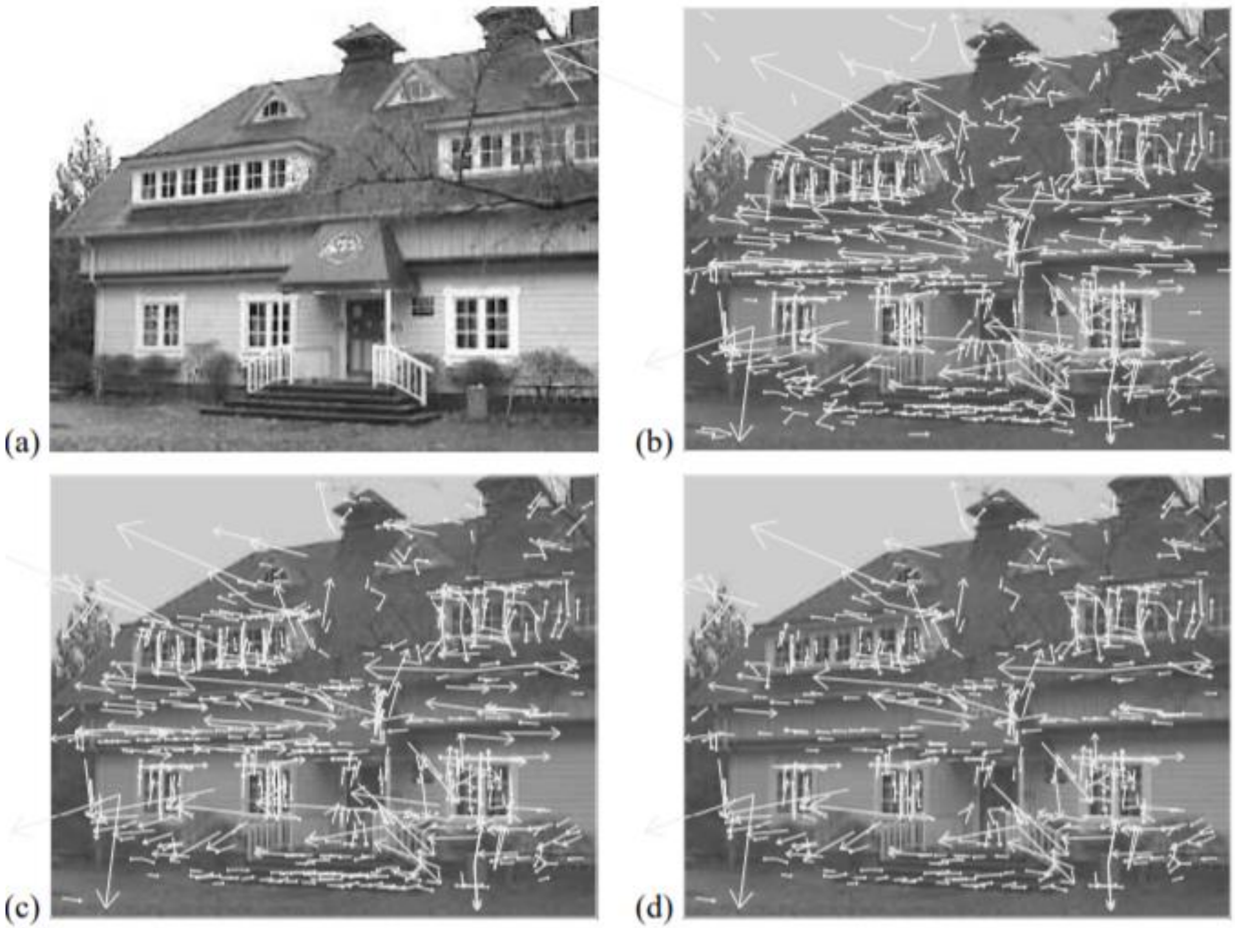


Figure 15: (a) Original Image, (b): Keypoints locations at maxima and minima of DoG, (c) Keypoints after performing thresholding on minimum contrast, (d): Keypoints after additional thresholding on ratio of principal curatures

3. Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations. For each sample, the gradient magnitude and orientation is computed using equation:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

4. Keypoint descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. First, image gradient magnitudes and orientations are

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019
sampled around the keypoint location. Image gradients and corresponding descriptors are shown in figure 12.

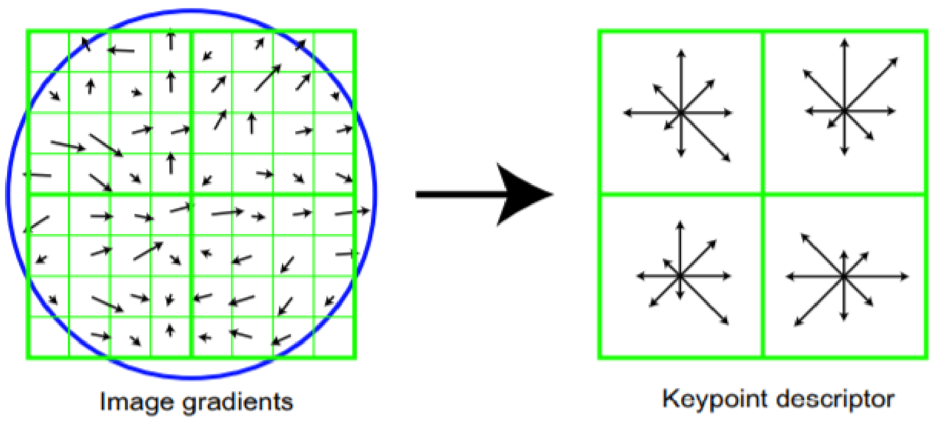


Figure 16: Image gradients and corresponding Keypoint descriptor

(b) Image Matching

One of the applications of finding SIFT features is to object matching. Given a train image, all of its keypoints and descriptors can be found using SIFT. And corresponding descriptors can be matched in the training data using nearest neighbor approach. It is known that each of the descriptor is 128D vector. Hence all the descriptors in train image is first plotted in the 128-dimensional space. Further, the descriptors of query images are found. For each of these descriptors, the location in the 128 -dimensional space is found. For each of this location. The nearest location for descriptors in train image is checked. If the distance is within certain threshold, the keypoints are matched. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken. If it is greater than 0.8, they are rejected. It eliminates around 90% of false matches while discards only 5% correct matches, as per the paper.

Procedure:

- The input image 'river1.raw' was read byte by byte into a 3D array of size 1024x768x3.
- The input image 'river2.raw' was read byte by byte into a 3D array of size 1024x768x3.
- The SIFT object was initialized using the sift_create method of OpenCV.
- The sift object was used to detect and compute keypoints and descriptor in each of the input images.
- The keypoints in each image was highlighted using a circle of small radius around each point.
- Further, the keypoint in the first image corresponding to highest l2 norm of the descriptors was found.
- The descriptor in the second image that matched with this descriptor was then found.
- A match line was drawn from these two descriptors by stacking the images horizontally.

- The results were saved, observed and analyzed.

(c) Bag of Words

Bag of Words is a technique used for image and text classification. In text classification, the number of times each word appears in a document is counted and made a histogram from it. Similarly, in image classification, this document is called Bag of Visual Words. The words here correspond to image features. Each image is represented by set of features and descriptors. Just like mentioned in the previous section, keypoints and descriptors can be sift features and its descriptors. So, each image is represented as frequency of these features. The illustration is as shown in figure 13.

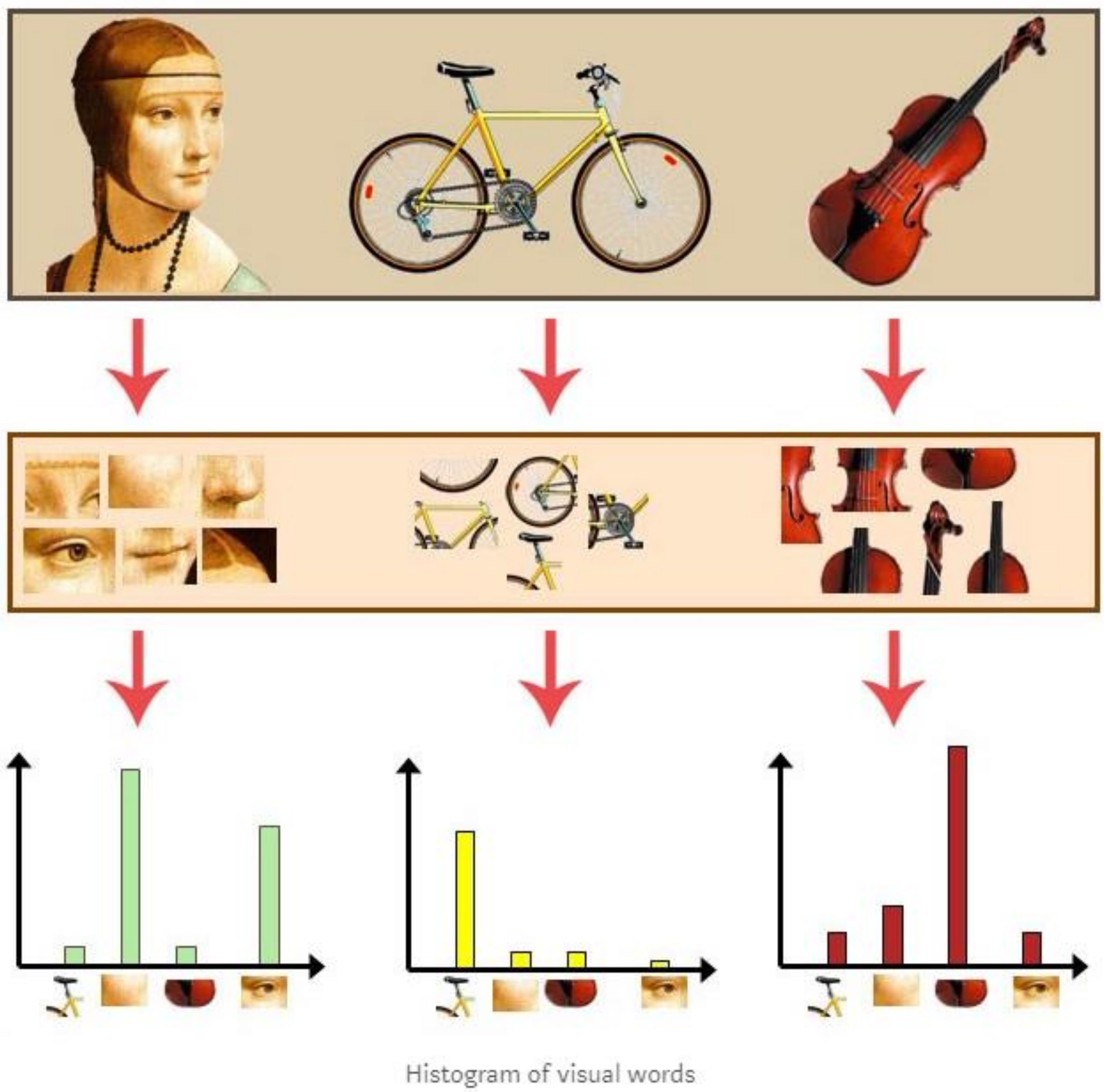


Figure 17: Histogram of visual words

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019
In this problem, the MNIST samples of zeros and ones are given. The histogram for the descriptors of zeros and ones are found using kmeans clustering for these descriptors. Further, the kmeans model is fit on the training data containing both the images of 0 and 1 for two clusters. The histogram of two bins for each of the images was separately plotted. The model was predicted on test image data of 8 image. The histogram was plotted for this. Now, min to max ratio was taken as:

min_max_eight_zero = eight_histogram[0]/zero_histogram[0] + eight_histogram[1]/zero_histogram[1]

min_max_eight_one = eight_histogram[0]/one_histogram[0] + eight_histogram[1]/one_histogram[1]

Finally, eight image was put in the cluster which gave higher min max ratio.

Procedure:

- Input images 'zero_1.raw' through 'zero_5.raw' and 'one_1.raw' through 'one_5.raw' were read byte by byte into 2D arrays of sizes 28x28.
- Sift object was created.
- Key points and descriptors for the zero images and one images were found out.
- The descriptors were fed as data to the kmeans object.
- Kmeans model was fit on these data.
- Labels for features in 0 and 1s were separated.
- Input image 'eight.raw' was read byte by byte into 2D array of size 28x28.
- The sift object was used to find key points and descriptors for eight image.
- The kmeans model was used to predict on the descriptors of eight image.
- The histogram was plotted for labels of each image.
- Min/max ratio was found using formula mentioned above.
- Eight image was classified into the higher one of the min/max ratios.
- Results were saved, observed and analyzed.

3. Results and Discussions

(a) SIFT

I. From the paper abstract, the SIFT is robust to what geometric modifications?

- Translation
- Rotation
- Scaling

II. How does SIFT achieves its robustness to each of them?

Translation: In translation, there is no interpolation in finding the pixel intensities except for the regions that are opposite from the translation direction. Hence all the neighborhoods for every pixel remains the same and hence the SIFT features and its orientations doesn't change. The gradient gradient and orientation histograms also do not change and hence can be matched for the same bins given the query image.
Rotation: Rotation invariance in achieved by selecting key locations at maxima and minima of a difference of Gaussian function applied in scale space. This is achieved by building image pyramid with

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019
resampling between each level. It also locates key points at regions and scales of high variation, making it very stable for characterizing the image.
Scaling: The maxima and minima of the scale space function are determined by comparing each pixel in the pyramid to its 8 neighbors, 9 neighbors in higher and lower scales taking account the 1.5 times resampling. This is how scale invariance is achieved by taking into consideration different scales.

III. How does SIFT enhances its robustness to illumination change?

The robustness to illumination change is enhanced by thresholding the gradient magnitude 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.

IV. What are the advantages that SIFT uses difference of Gaussians (DoG) instead of Laplacian of Gaussians (LoG)?

- Faster than LoG with sufficiently equal accuracy.
- Approximates use lesser complexity in space as well.

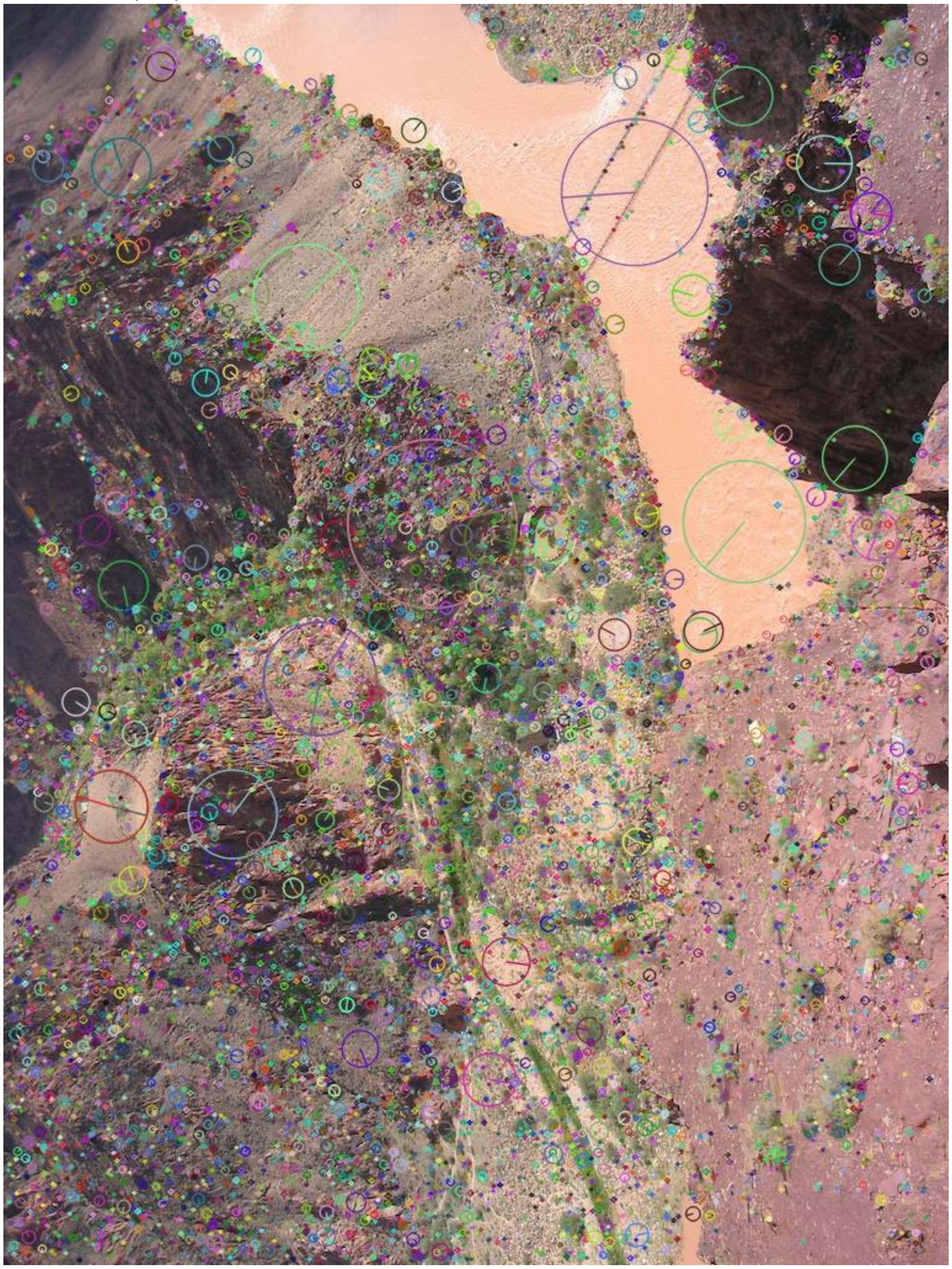
V. What is the SIFT's output vector size in its original paper?

A 4x4 sample region for 8 orientation planes and a 2x2 sample region for second level of pyramid with 8 orientation planes I,e, 8*4*4 + 8*2*2 = 160

(b) Image Matching

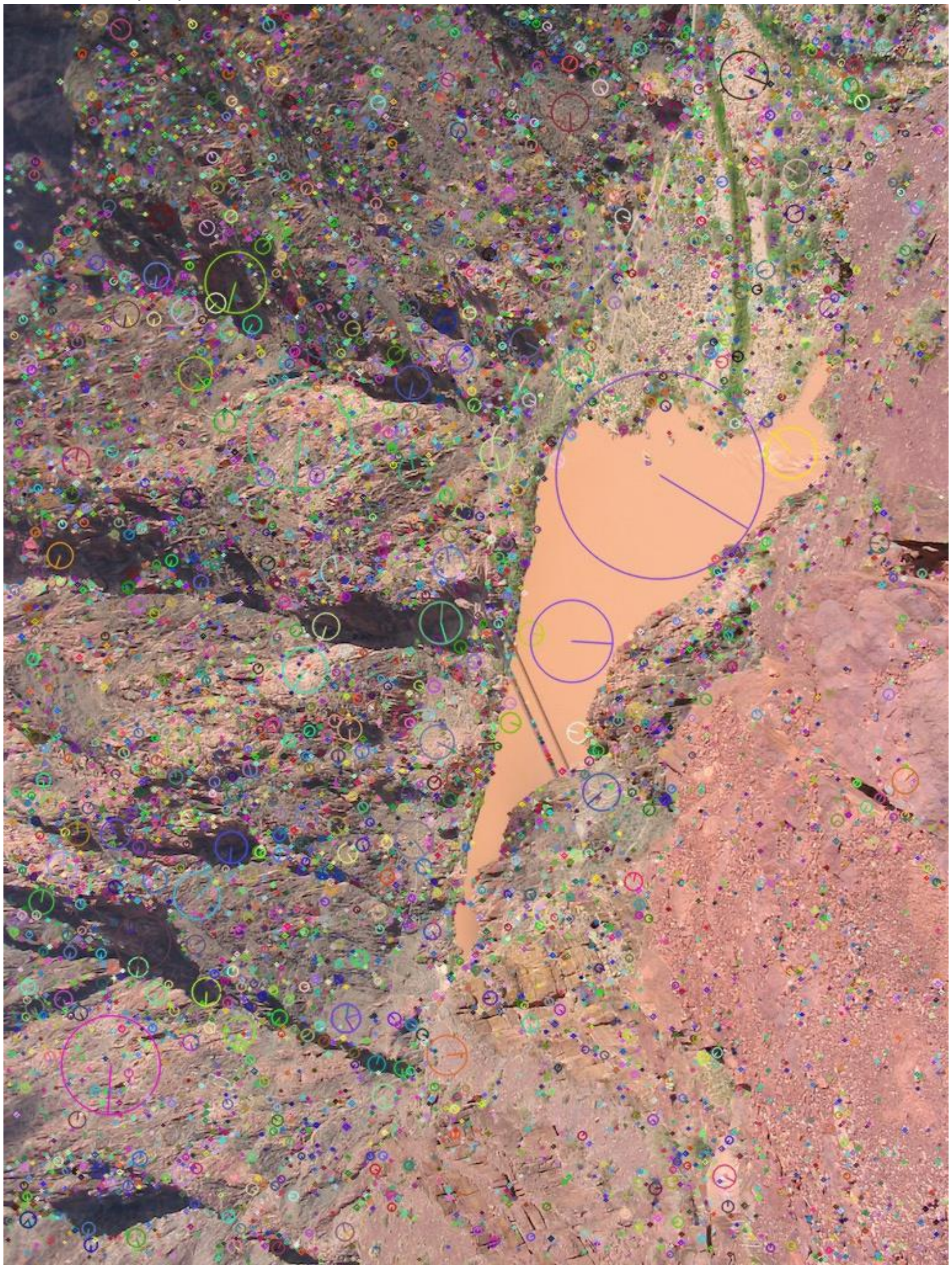
- The key points and descriptors for river1.raw is as shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



- The key points and descriptors of the river2.raw is as shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019



- The match between the largest magnitude descriptor (descriptor with highest l2 norm) of river1.raw with river2.raw is as shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

- The highest magnitude descriptor in river1.raw was at index 629 with l2 norm of 513.66.
- The orientation of each key point has this significance: An orientation histogram obtained from the gradient orientations of sample points within a region around the keypoint has 36 bins covering 360 degree range of orientations. Peaks in the orientation histogram correspond to dominant directions of local gradients. The measured orientation of matches of these descriptors is about 2.5 degrees. Hence, the keypoints in two images with similar magnitude values (same circle size in figure above) with 2.5 degree variation in orientation (line drawn from the center of the circle to its circumference) is said to be a match between train and query images.

(c) Bag of Words

With resizing all the zero, one and eight images to 56x56

- The histogram plot for descriptors of zero image is as shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

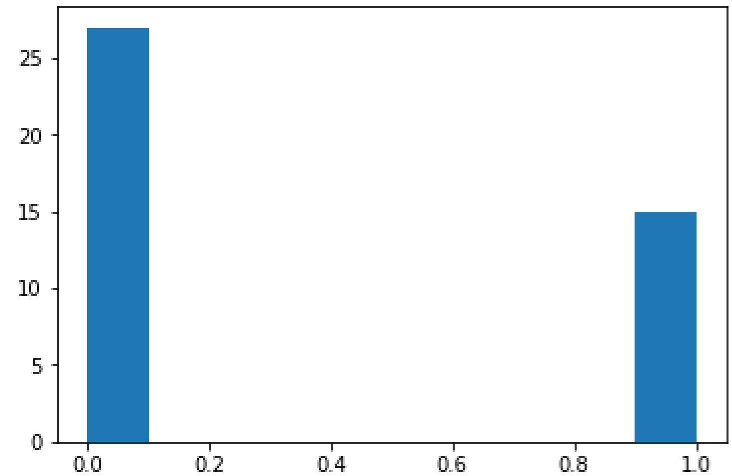


Figure 18: Histogram for zero images with two bins

- The histogram plot for one images is as shown below:

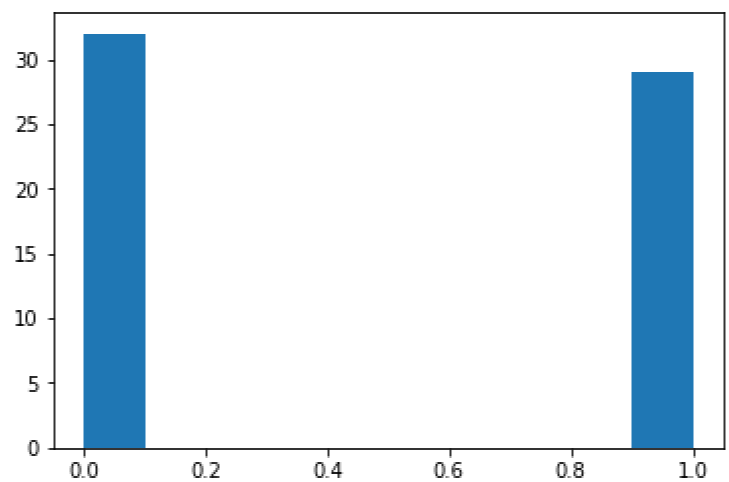


Figure 19: Histogram for one images with two bins

- The histogram for eight image is as shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

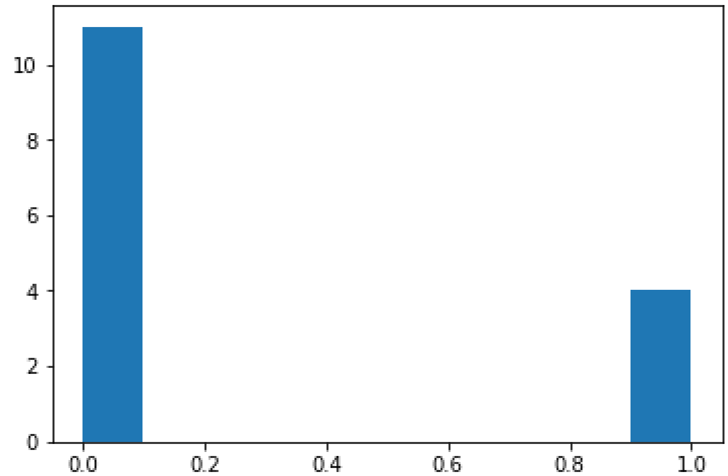


Figure 20 : Histogram for eight images with two bins

- The min_max ratio for 8 and 0 was 0.674 and the min_max ratio for 8 and 1 was 0.48168
- The eight image is now classified as 0.

Without resizing all the zero, one and eight images:

- The histogram for zero image is shown below:

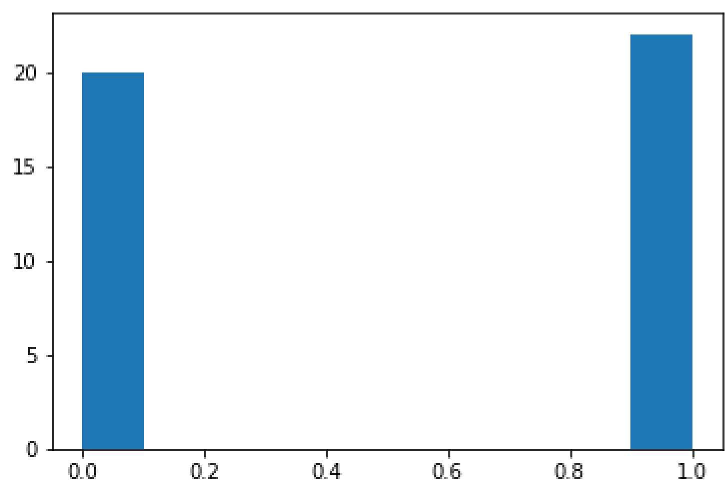


Figure 21 : Histogram for eight images with two bins

- The histogram for one image is shown below:

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

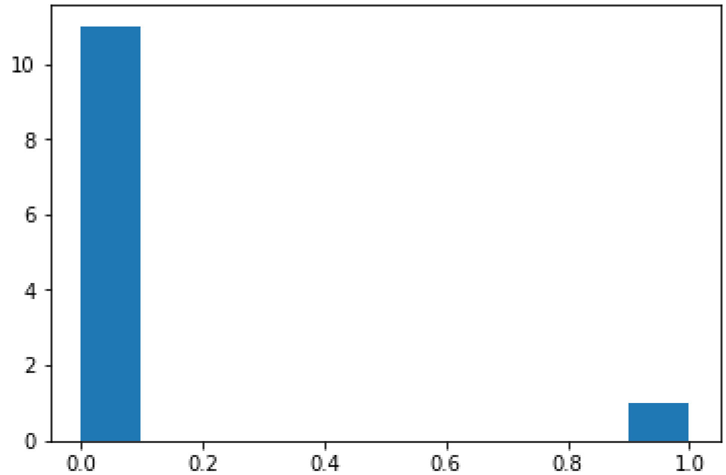


Figure 22: Histogram for eight images with two bins

- The histogram for eight image is as shown below:

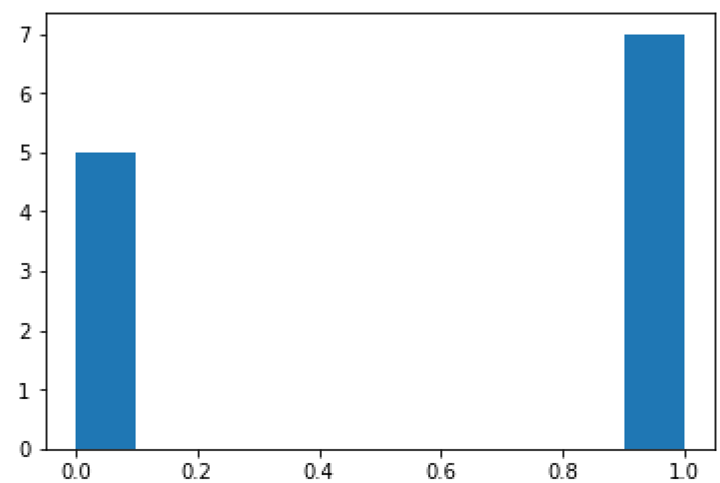


Figure 23 :Histogram for eight images with two bins

- The min_max ratio for 8 and 0 was 0.5681 and the min_max ratio for 8 and 1 was 0.5974
- The eight image is now classified as 1.

General Discussions:

- The codebooks here are the centroids of the clusters formed using the descriptors of 0 and 1.
- Same kmeans model was used in predicting the clusters for eight image as was used in forming the clusters for zero and one images.

Akash Mohan Das
USC ID: 7247503828
USC email: amohanda@usc.edu
Submission date: 03/19/2019

- There is no descriptor found in one_5 image without resizing and hence the model in this case might suffer from imbalanced data sets (4 ones and 5 zeros).
- As the resize factor was increased the min/max ratio became more biased towards 1.