EE569 Project 4

Name: Shreyas Dinesh Patil
Email: shreyasp@usc.edu

Programming language used: Python
Problem 1: Texture Analysis

1(a) Texture classification

I. Abstract and motivation

Texture classification is one of the four major texture analysis tasks. The other three are texture segmentation, texture synthesis and shape from texture. In this problem texture classification is performed on 12 texture images using 5x5 Laws filters for feature extraction. PCA is used for dimension reduction and K-means clustering is done for classifying the images into bark, straw, brick, and bubbles type textures.

II. Approach

1D Kernel for 5x5 Laws Filters

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]

Method 1:

1) The boundary extension of image is done based on pixel replication.
2) Create 25 2D 5x5 Laws filters by taking the tensor product of the given 1D Kernels.
3) For each texture image
   I) Subtract mean of image pixel values from each pixel.
   II) Apply 25 5x5 Laws filters to image to get 25 filtered images.
   III) Take average of sum of absolute values of pixels for each of the 25 filtered images to get 25D feature vector.
4) Standardize each of the 25 features in the 12x25 feature matrix obtained. Also create another 12x15 feature matrix.
5) Use Principal Component Analysis to reduce 25 features to 3. (sklearn.decomposition.PCA used). Reduce other 15 features to 3 using PCA.
6) Apply K-means clustering algorithm with number of clusters equal to 4 and k-means++ centroids initialization to 12x3 data matrices obtained from 25 and 15 features. (sklearn.clusters.Kmeans used)
7) Display the array of labels for 12 images.

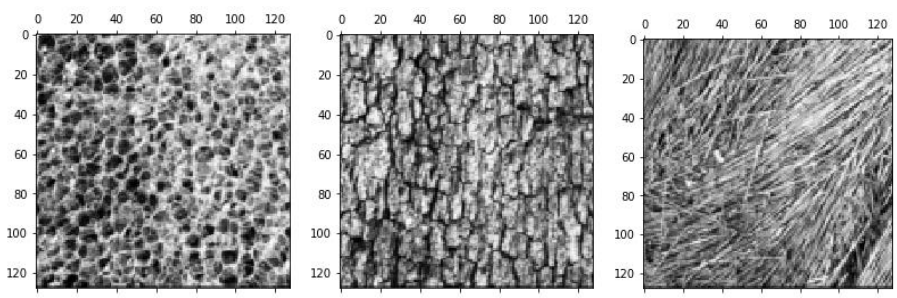
Method 2:

1) The boundary extension of image is done based on pixel replication.
2) Create 25 2D 5x5 Laws filters by taking the tensor product of the given 1D Kernels.
3) For each texture image
    IV) Subtract mean of image pixel values from each pixel.
    V) Apply 25 5x5 Laws filters to image to get 25 filtered images.
    VI) Take average of sum of absolute values of pixels for each of the 25 filtered images to get 25D feature vector.
4) Standardize each of the 12 data points in the 12x25 feature matrix obtained. Also create another 12x15 feature matrix.
5) Use Principal Component Analysis to reduce 25 features to 3. (sklearn.decomposition.PCA used). Reduce other 15 features to 3 using PCA.
6) Apply K-means clustering algorithm with number of clusters equal to 4 and k-means++ centroids initialization to 12x3 data matrices obtained from 25 and 15 features. (sklearn.clusters.Kmeans used)
7) Display the array of labels for 12 images.

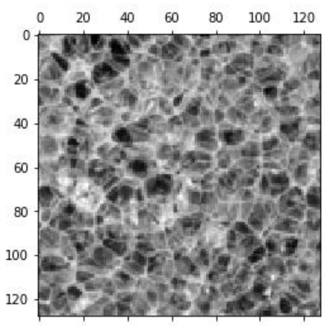
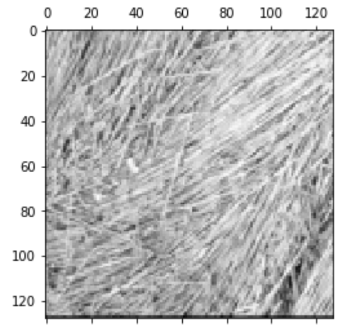
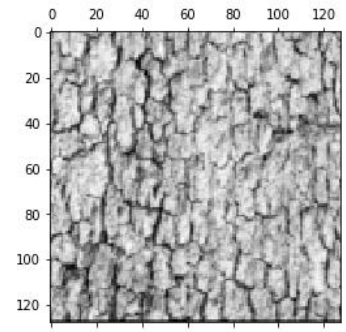
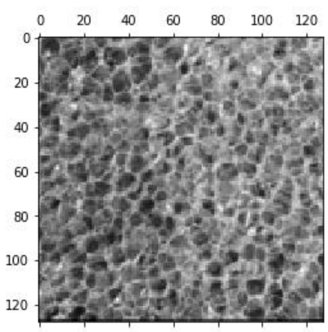
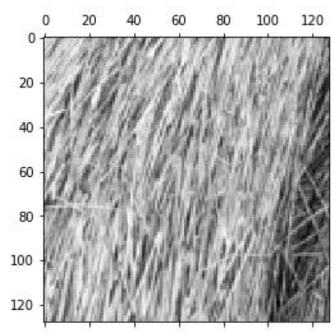
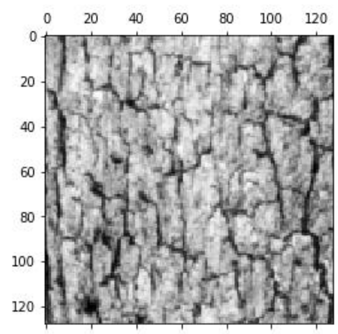
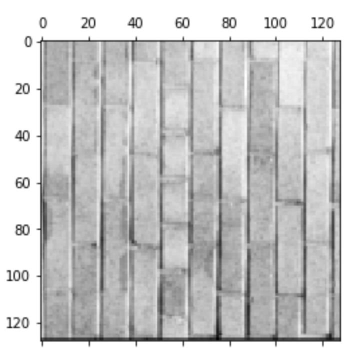
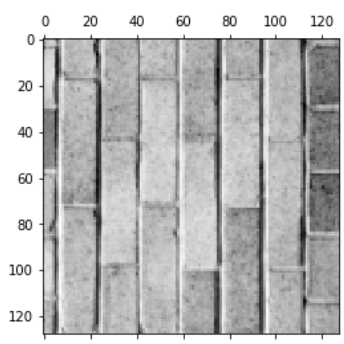
III. Experimental Results

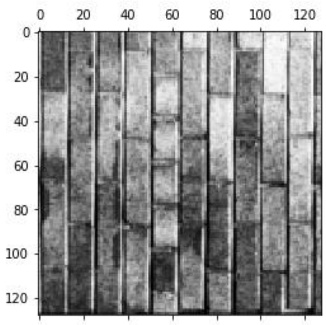
Method 1 results for 3D data:

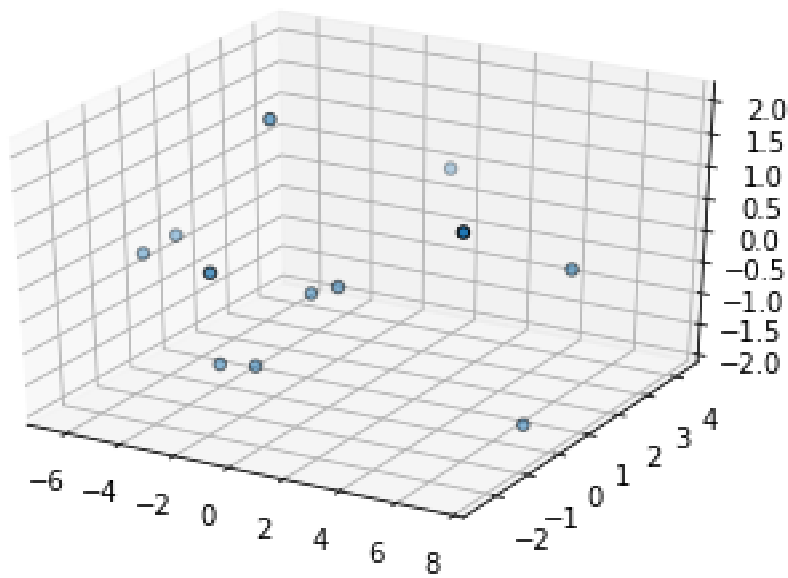
array([1, 1, 2, 1, 0, 0, 1, 0, 3, 1, 2, 1])
Here the index of the array+1 is the texture number. For eg. Texture1.raw is classified as label 1



//





Method 1 results for 25D data:

array([0, 0, 2, 0, 1, 1, 0, 1, 3, 0, 2, 0])

Here the index of the array + 1 is the texture number. For eg. Texture1.raw is classified as label 0.

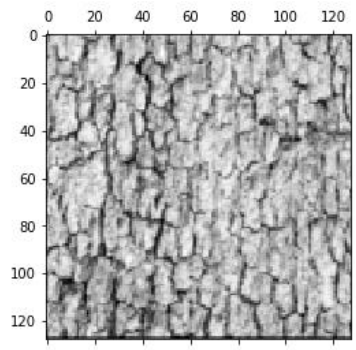
Method 2 results for 3D data:

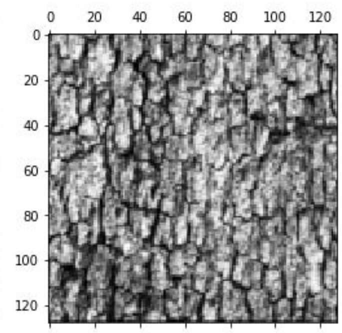
Classification result obtained using PCA:

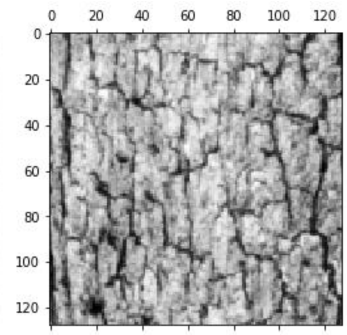
Array of labels after classification:

array([3, 2, 1, 0, 3, 0, 3, 2, 1, 1, 1, 0])
Here the index of the array+1 is the texture number. For eg. Texture1.raw is classified as label 3

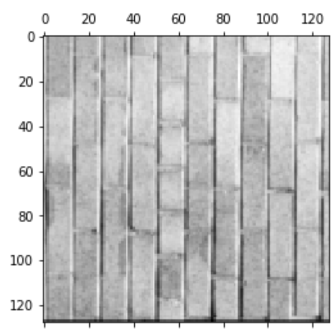
Class 0

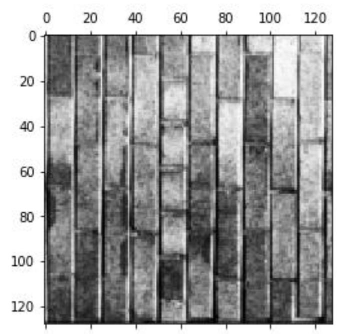


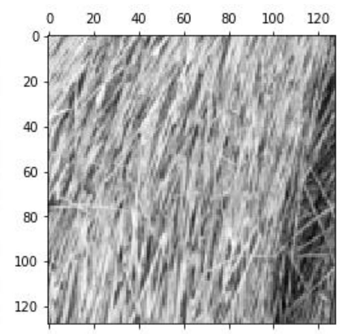


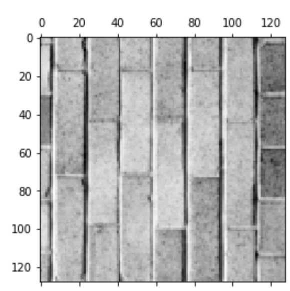


Class 1

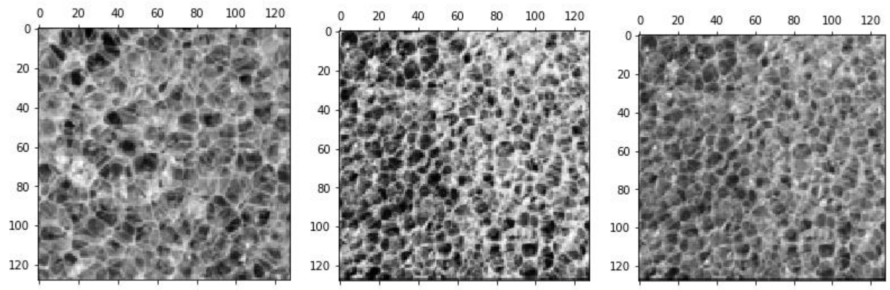




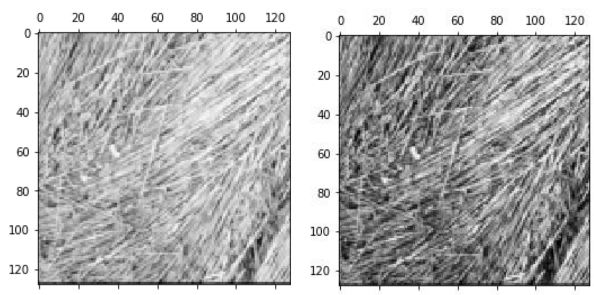




Class 3



Class 2



Classification result obtained without using PCA:

Array of labels after classification:

array([1, 3, 0, 2, 1, 2, 1, 3, 0, 0, 0, 2])
Here the index of the array+1 is the texture number. For eg. Texture1.raw is classified as label 1

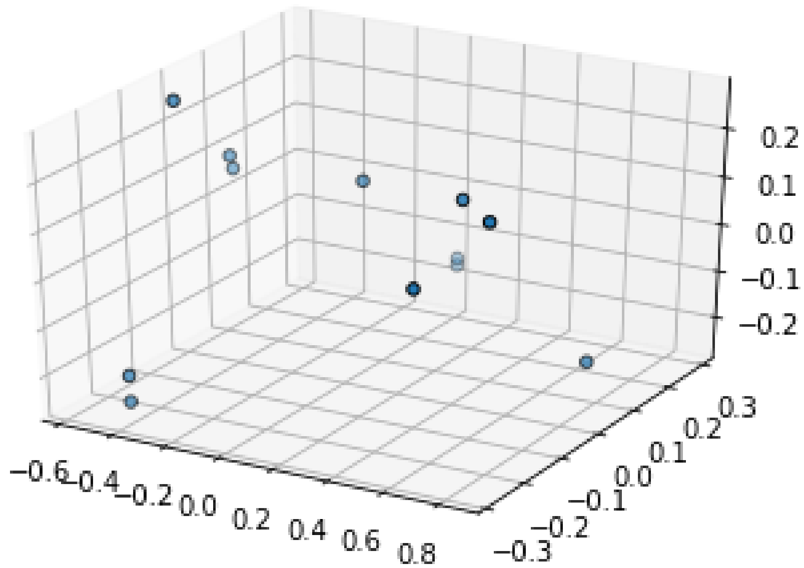


Figure: 3D plot of 12 texture points

Variance of data points for each feature vector

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
3942309.360	215575.15	75788.60	76286.63	255175.97	206659.76	16391.42	5798.79	5083.71	16104.91

F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
57882.58	5638.33	2262.89	2128.13	7071.10	41444.43	4966.52	2229.39	2283.72	8228.71

F21	F22	F23	F24	F25
122349.52	15663.30	7898.74	9216.27	37067.59

IV. Discussion

- There are many misclassifications in the results obtained using method 1. This may be due to less number of samples being available for k-means clustering.
- Reducing 25 features to 15 does not improve the classification accuracy. So, I have kept all 25 features.
- Method 2 gives only 1 misclassification of texture10.raw with 25 features and there is no change in accuracy with using 3D feature vector of PCA
- Laws filters are used for extracting features from images. L5 filter gives center weighted local average, E5 detects edges, S5 detects spots, W5 detects waves and R5 detects ripples textures in image.

- From the variance table we can see that the feature L5T L5 has the highest variance of data points compared to other features and feature S5T S5 has the lowest. So, feature S5T S5 has the highest discriminating effect on the k-means clustering algorithm and feature L5T L5 lowest.
- Feature dimension reduction does not help in increasing the accuracy of classification. But it reduces the time required for clustering by k-means algorithm because the dimension of the data is reduced from 25D to 3D. There is no change in results obtained from changing dimensions 25D to 3D.

1(b) Texture segmentation

I. Abstract and motivation

Textures can be valuable features for image segmentation. Partitioning image into small number of homogeneous regions helps us to extract important features. Image segmentation methods can be divided into 2 kinds; region-based and edge-based. Here region based texture segmentation is used with Laws filters for extracting features from each image pixel. K-means is used for clustering similar pixels together.

II. Approach

1D Kernel for 5x5 Laws Filters

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]

1) Create 25 2D 5x5 Laws filters by taking the tensor product of the given 1D Kernels.
2) For each texture image

   Subtract mean of image pixel values from each pixel.

   Apply 25 5x5 Laws filters to image to get 25 filtered images.
3) For 25 filtered images
   I) Take window of size 5x5 for each pixel and calculate absolute sum of pixel values in that window to get energy.

   From step 3 we get 25D feature vector for each pixel in the image.
4) Normalize the feature vector using the energy of feature extracted by $L5T\ L5$.
5) Apply K-means clustering algorithm with number of clusters equal to 7 and k-means++ centroids initialization to the 510x510x25 feature matrix. (sklearn.clusters.Kmeans used)

III. Experimental results

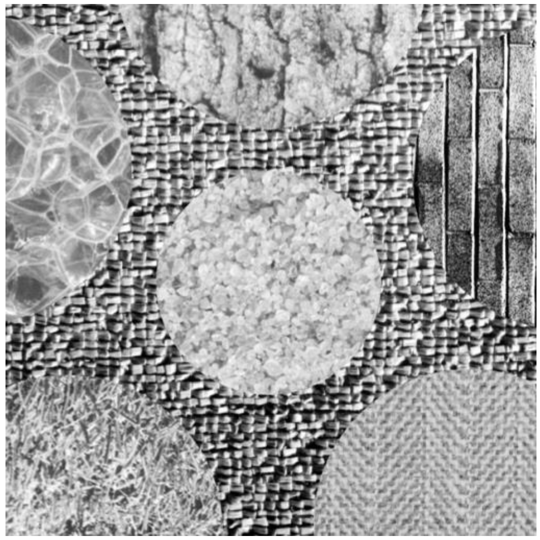


Figure1: comb.raw

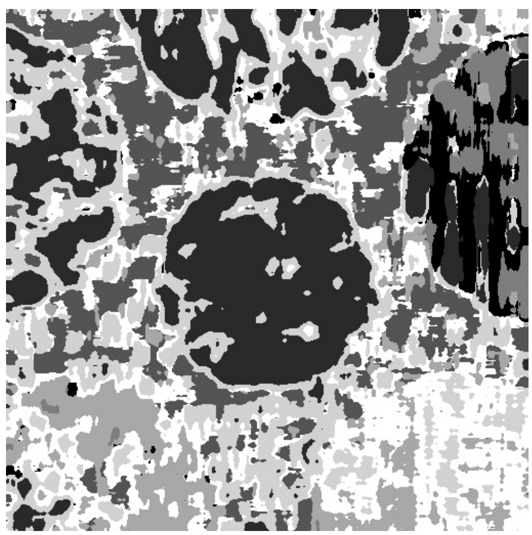


Figure1: Result for window size 17 and L5T L5 energy feature normalization

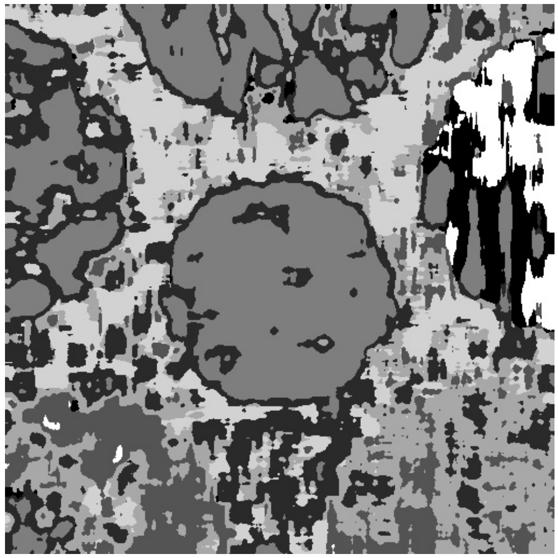


Figure1: Result for window size 19 and L5T L5 energy feature normalization

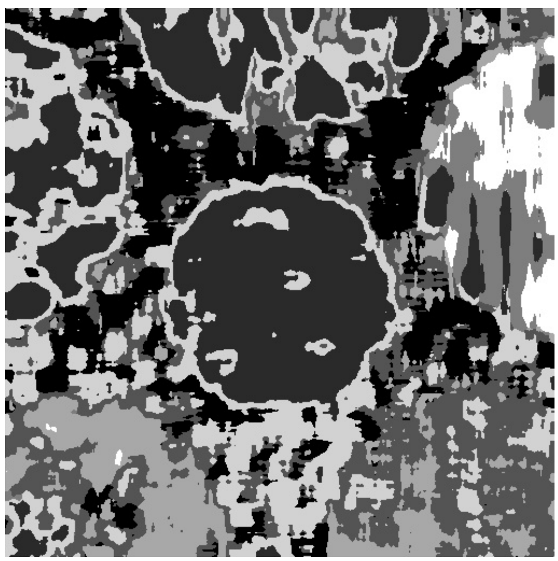


Figure1: Result for window size 21 and L5T L5 energy feature normalization

IV. Discussion

- From experimental results we see that increasing the window size has not much effect on improving the result. But after certain window size different regions start connecting and results in bulgy output.
- Segmentation result get better with window size but after certain point the there are connecting regions observed between the texture regions.

1(c) Advanced Texture Segmentation Techniques

I. Experimental results

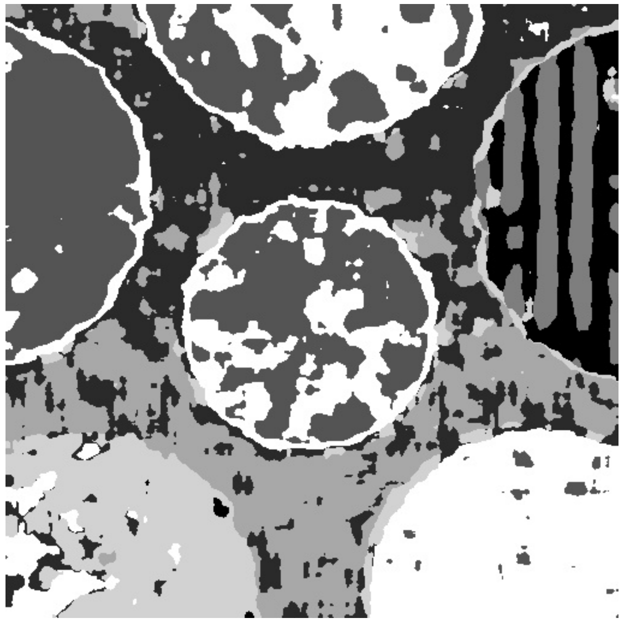


Figure1: Result for window size 17 with standard normalization and PCA (components 3)

Figure2: Result on applying closing operation on Figure1

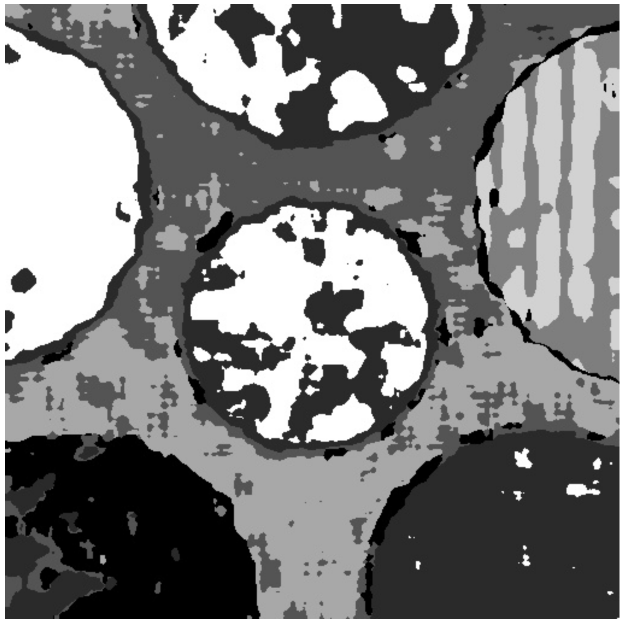


Figure3: Result for window size 19 with standard normalization and PCA (components 3)

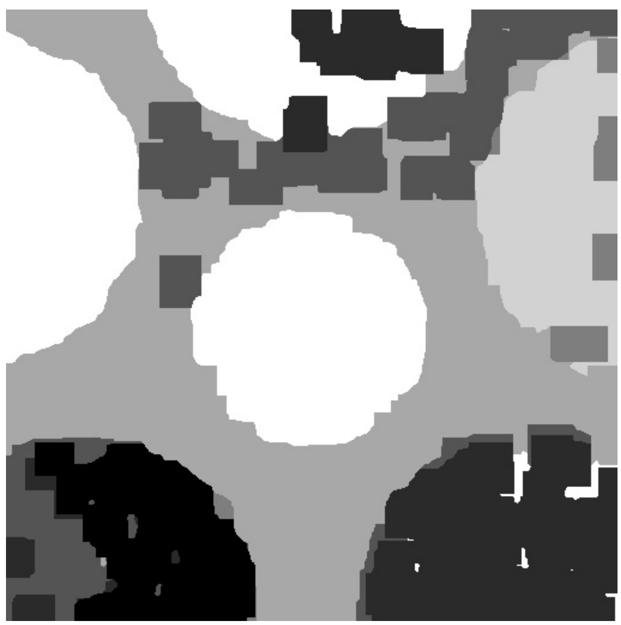


Figure4: Result on applying closing operation on figure3

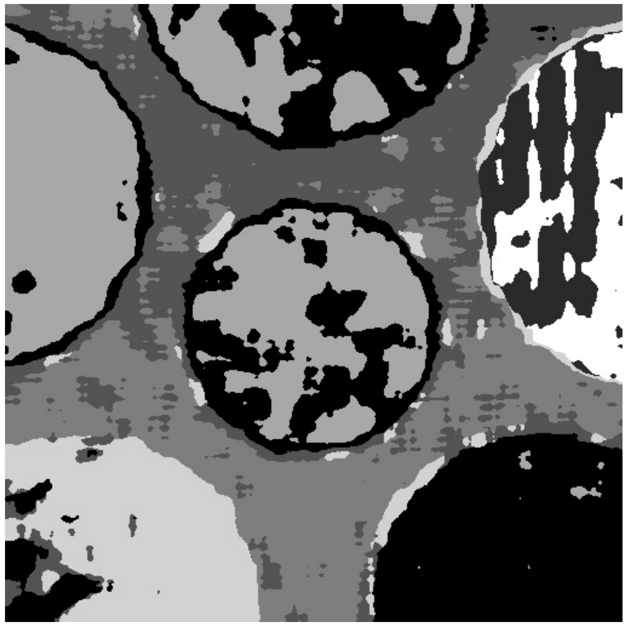


Figure5: Result for window size 21 with standard normalization and PCA (components 3)

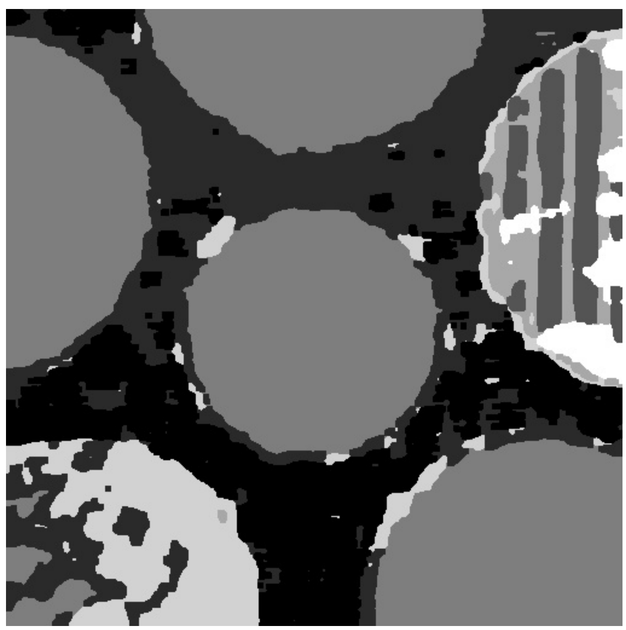


Figure6: Result on applying closing operation on figure5

II. Discussion

- We can see that the results improve if we use standard normalization for features along with PCA.

- Applying morphological closing operation on obtained PCA result improves result by small margin. We can see that some holes have been removed in the texture regions
- Using PCA for feature dimension reduction after the step of obtaining 25D energy feature vector for each pixel does not have much effect on improvement of results.
- Closing morphological operation improves the segmentation results by removing the holes in the textures.

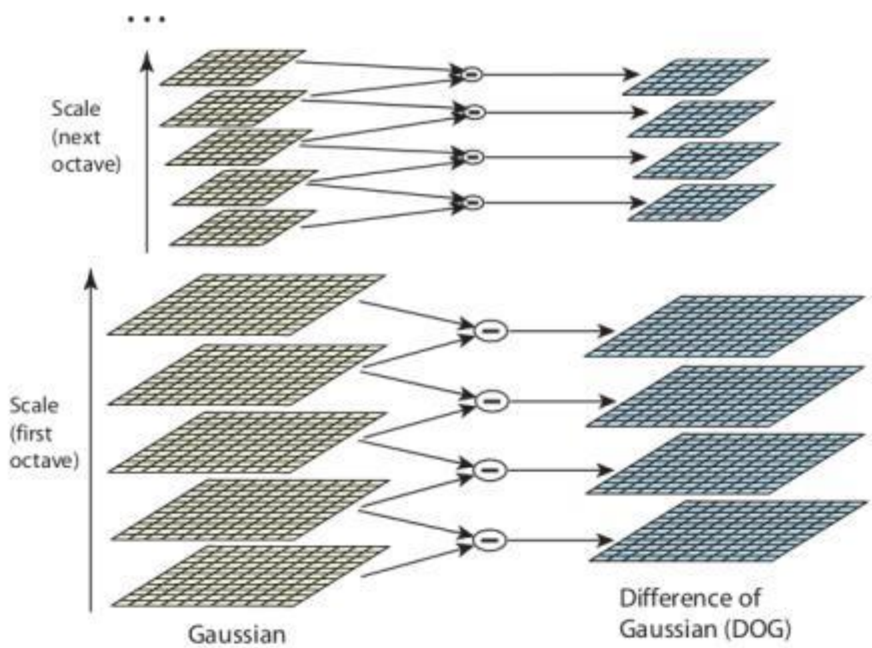
Problem 2: Image Feature Extractor

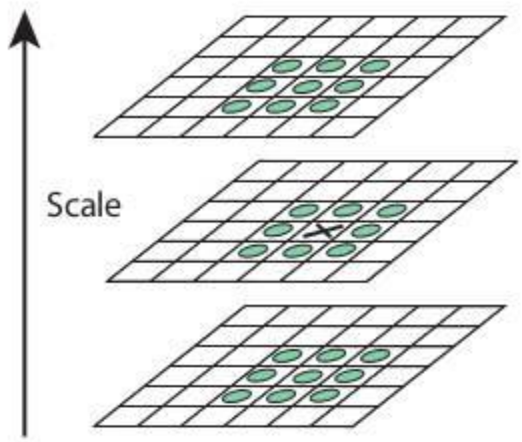
2(a) SIFT

I. SIFT is robust to image scaling, translation and rotation and partially robust to illumination changes and affine or 3D projection.

II. A) Scale space is the space of gaussian filtered images with different sizes and sigma values. Images are stacked over each other as octaves in pyramid form. Each octave contains 4 images (as per paper) with same size and different sigma filtered images.

So, scale space has 3 coordinates; x, y and sigma. Sigma acts as the scaling parameter. The following image shows how difference of gaussian is calculated.





Once this Difference of Gaussian are found, images are searched for local extrema over scale and space. These local extrema are the potential key locations.

This scale space pyramid brings invariance to scaling of images.

B) In SIFT scale space is created which causes invariance to scaling. In this step original image is progressively blurred using gaussian filters with different sigma values and then the original image is resized to half and the process is repeated. Each key point is assigned orientation based on local image gradients. Gradients at key points are calculated using difference of pixel values. Magnitude and gradient are calculated for every pixel in the neighboring region of the key points. Histogram of orientations is created with 36 bins. So, each bin covers 10 degrees. Each pixel in the neighborhood of the key point is weighted in the histogram by its gradient magnitude and circular window with sigma that is 1.5 times that of the scale point. The final orientations are the bins with the highest value and 80% of the highest value in the histogram. This orientation to each key point brings invariance of features to image rotation.

III. By computing descriptor vector for each detected key point SIFT achieves robustness to variations such as illumination, 3D viewpoint etc. In this process a 16x16 region is selected around the key point and orientation histograms are created from samples in 4x4 neighborhood sub regions with 8 bins each. The descriptor is the vector of all the values of these histograms. The vector size becomes 128 since 16 histograms in 16x16 neighborhood region around key point each with 8 bins. This vector is normalized in order to make it invariant to affine changes in illumination. For countering non-linear illumination effects a threshold of 0.2 is used and the vector is again normalized.

IV. LOG is computationally costly so DOG which is approximation of LOG is used. Difference of Gaussian is computed efficiently by resampling of image

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

2(b) Image Matching

I. Abstract and Motivation

Image matching is aspect of object or scene recognition, motion tracking etc. in computer vision. In image matching generally a query image is matched against a database of images. This is done by extracting features of interest points in the query image and matching them against the interest points features of images in the database. The image whose interest point features are close to that of query image is matched.
In this problem we use SIFT (Scale Invariant Feature Transform) for detecting interest points and generating features in the given 2 river images. We then match the interest point with largest scale in one image to nearest neighbor key point in the other.

II. Approach (OpenCV SIFT implementation used)

1) Detect and compute key points and descriptors using SIFT in both images.
2) Find the key point and descriptor corresponding to largest scale in river image 1.
3) Find descriptor matching that from 2) in river image 2. (used flann.knnMatch())
    i) Select best matching keypoints based on the ratio test mentioned in paper with threshold = 0.75.
4) Use FLANN based matcher to match the key point from step 2) to nearest neighbor key point of river image 2 using the matches obtained in step 3. (used drawMatchesKnn())
5) Display the image with key points and matching lines.

III. Experimental results

Figure1: river1 image with key points

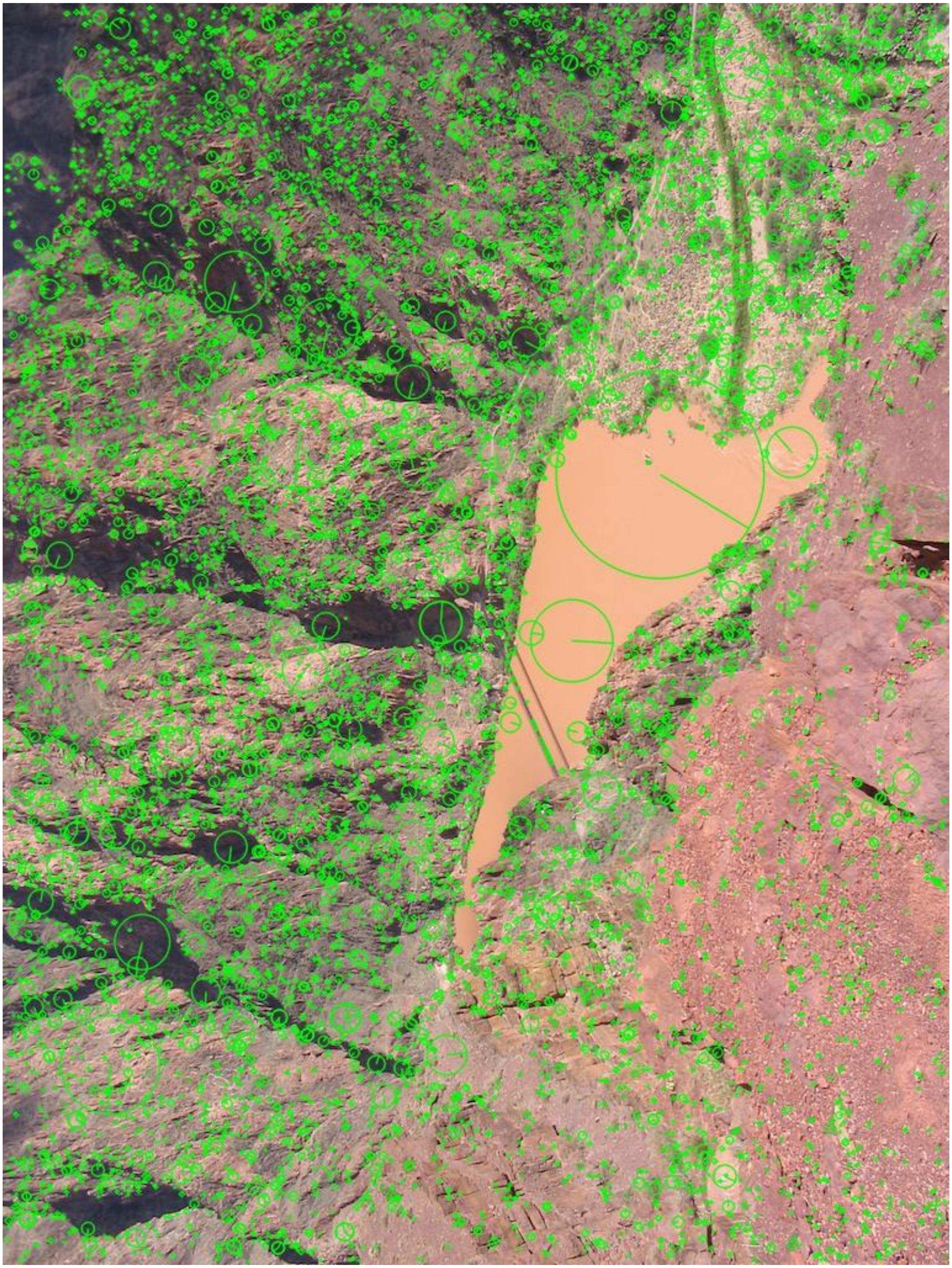


Figure2: river2 image with key points

Figure3: key point with largest scale (highest l2 norm) shown in green.

Figure4: Matching key point with largest scale (highest l2 norm) in river1 to river2(shown in white line)

Orientation of key point in river image1 206.17718505859375
orientation of matching key point in river image2 129.88003540039062

IV. Discussion

- The diameter of the circle around key point in image determines meaningful neighborhood. The line inside the circle determines orientation of gradient of the key point.
- The orientation of key points is measured w.r.t. image coordinate system in degrees.
- From the orientation values of key point in river image1 we can infer that the majority of pixels in the neighborhood of the key point are oriented in 206.177 degrees and similarly for matching key point in river image 2 are oriented in 129.88 degrees.

2(c) Bag of Visual Words

I. Abstract and Motivation

Bag of visual words is a technique which is used in image classification. In this technique the words represent the image features that are unique patterns in image. Each image is represented as frequency histogram of features in image. In this problem SIFT algorithm is used to get key points and descriptors (which are vectors representing the detected key point features). These vectors are converted to codewords using k-means clustering. That is, codewords are defined as the centers of learned clusters.
Here there are only 2 clusters corresponding to 'ones' and 'zeros' images. Then the histogram of codewords is created. In this problem the histogram for zeros and ones images combined is obtained and compared against the histogram of 'eight' image to identify the class of the image.

II. Approach (OpenCV SIFT implementation used)

1) Read the given images.
2) Find key points and descriptors corresponding to the 4 'one', 4 'zero' and 'eight' image using sift.detectAndCompute().
3) Use k-means clustering on the descriptors of 4 'one' images and 4 'zero' images with number of clusters 2 combined.
4) Cluster the descriptors for image 'eight' with number of clusters 2.
5) Plot the histogram of train data labels and also labels obtained for descriptors of 'zero' and 'one' image descriptors.
6) Plot the histogram of labels for descriptors of image 'eight'.
7) Use histogram intersection method to classify image 'eight'.
   i) Add the (min/max) value of each bin of histogram of image 'eight' with the corresponding bin in 'zero' images histogram.
   ii) Add the (min/max) value of each bin of histogram of image 'eight' with the corresponding bin in 'one' images histogram.
   iii) Compare the values obtained in i) and ii) and classify image 'eight' to the class having highest value among zero and one.

III. Experimental results

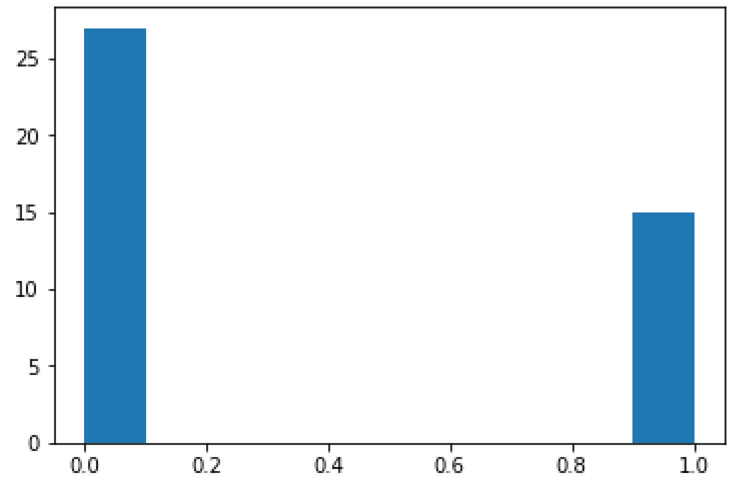


Figure1: Histogram of ‘zero’ images

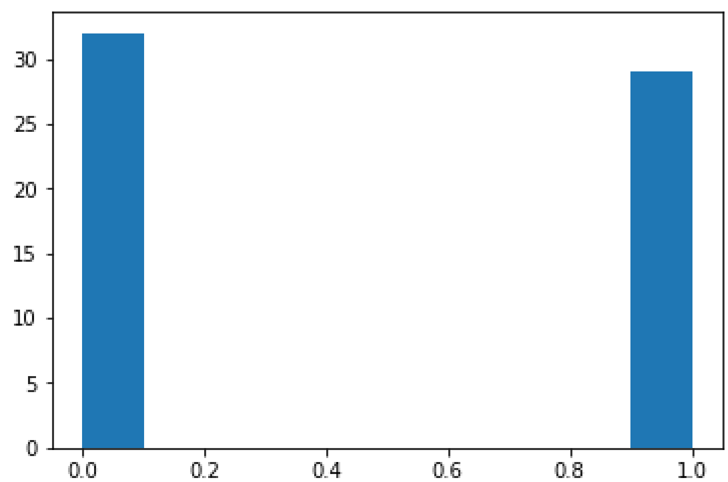


Figure2: Histogram of ‘one’ images

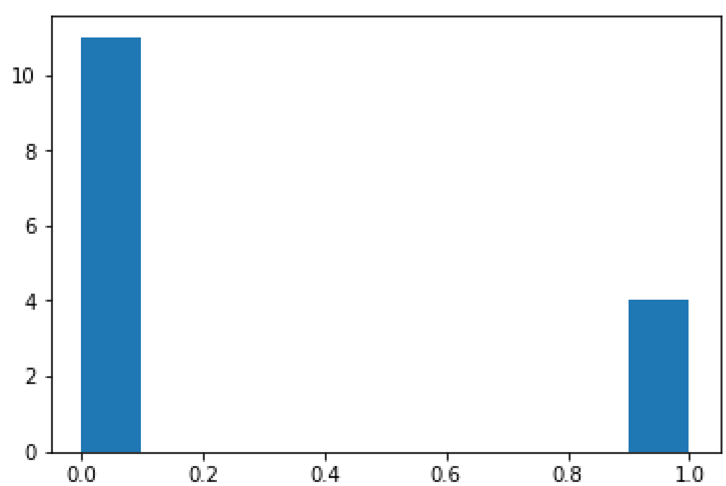


Figure3: Histogram of ‘eight’ image

Intersection with zeros histogram 0.5813397129186603
Intersection with ones histogram 0.5595238095238095

IV. Discussion

- We use histogram intersection method to classify image ‘eight’. Numerically histogram intersection method gives intersection value for ‘zero’ images histogram higher than for ‘one’ images.
- SIFT here is used for feature extraction because the original given images are rotated in any direction. As SIFT calculates key points invariant to rotation it is very effective in this problem.
- Bag of visual words is not effective because it does not use discriminative features and each codeword is independent of others. Therefore, model does not capture global information in the image.

References

- https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb
- https://courses.cs.washington.edu/courses/cse576/book/ch7.pdf
- https://www.cs.ubc.ca/~lowe/papers/iccv99.pdf
- https://people.eecs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf
- https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html