PROBLEM 1

Texture Analysis

(a)

Texture Classification

1. **Abstract and Motivation:**

   Texture is basically pixels with high frequency variations but with much less randomness than noise. Texture classification is important since to distinguish texture from the noise, which is also high frequency, helps interprets the image better. Texture classification is the classification of texture based on type by bringing out the characteristic of the texture pixel intensity variation. In this section, we use Laws filters for feature extraction and use the extracted features to classify the twelve textures into four classes.

2. **Approach and Procedures:**

   Laws filters are initially 5D filters and are of the following 5 types:

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]

   The 5 filters have their own responses. The L5 filter is a zero phase low pass filter. The S5 filter is a bandpass filter. The R5 filter is a high pass filter. The E5 filter is a band pass filter with a phase change of 90 degrees.

   These filters are convolved to obtain 2D Laws filters which have a combination of the individual responses with certain modifications. The L5 filter computes the local average, the E5 filter detects the edge and the S5 filter detects the spots. The R5 filter detects diagonal edges.

   Algorithm:

1. The tensor products for every 5 combinations of the 5 Laws filters is computed to determine 25 2D filters.
2. These 2D filters are convolved through the image for each pixel as the center pixel.
3. Each pixel as a datapoint, the 25D feature vectors are calculated.
4. Now the features are normalized using the 0 mean and variance 1 approach
5. PCA is then applied to reduce computation time.
6. It is then passed to k-means to classify the textures.
7. The PCA plot is plotted.

K-means clustering:

- unsupervised learning algorithm
- moves centroids with respect to density of datapoints
- stop when centroids converge on simultaneous iterations

PCA

- Used to speed up the algorithm
- Reduces dimensions based on correlation
- The limitation is that if there is no correlation, the components are ordered via variance which might not be as effective.

3. Results:

Ideal results:

The textures are numbered as provided.

An ideal result would imply that the following textures are grouped together:

1 5 7

2 8 10

3 9 11

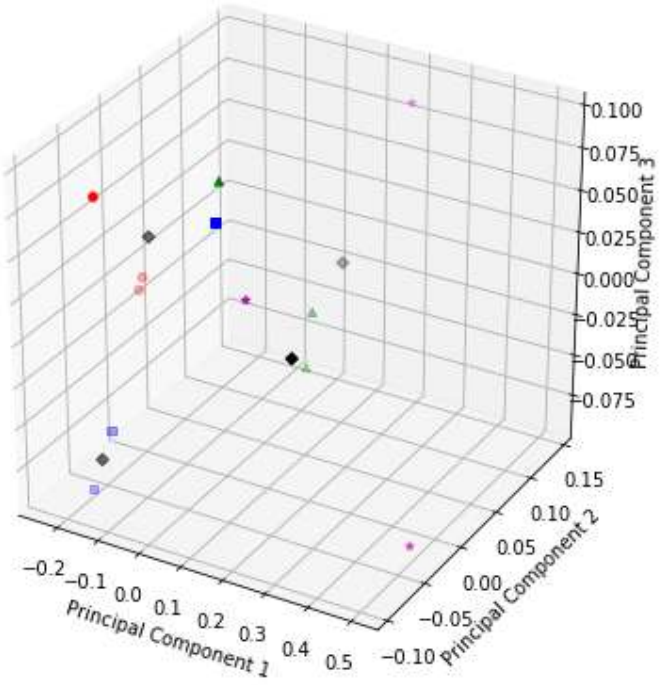
4 6 12

Note that the black diamond co-ordinates in the plots are the centroids. Similar texture co-ordinates have similar colour and shape. The labels in the result sections correspond to different groups and not necessarily to the sequential class label. However, the accuracy metric is also mentioned for reference.

1. Average energy computed using: Mean of squares
   Classification algorithm: K-means
   Initial centroids passed: Principal Components for textures 5,8,9,6
   Accuracy: 75% (9/12)
   Results:

```
label 0 : [1, 5, 7, 12]
label 1 : [2, 8]
label 2 : [3, 9, 10]
label 3 : [4, 6, 11]
```

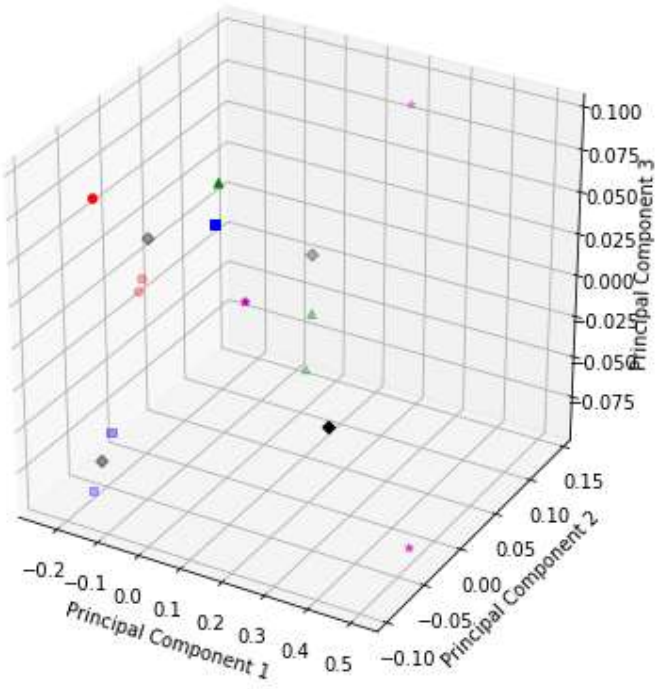
Plot:



2. Average energy computed using: Mean of squares
   Classification algorithm: K-means
   Initial centroids passed: Principal Components for textures 1,2,3,4
   Accuracy: 9/12
   Results:

```
label 0 : [1, 5, 7, 12]
label 1 : [2, 8]
label 2 : [3, 9]
label 3 : [4, 6, 10, 11]
```

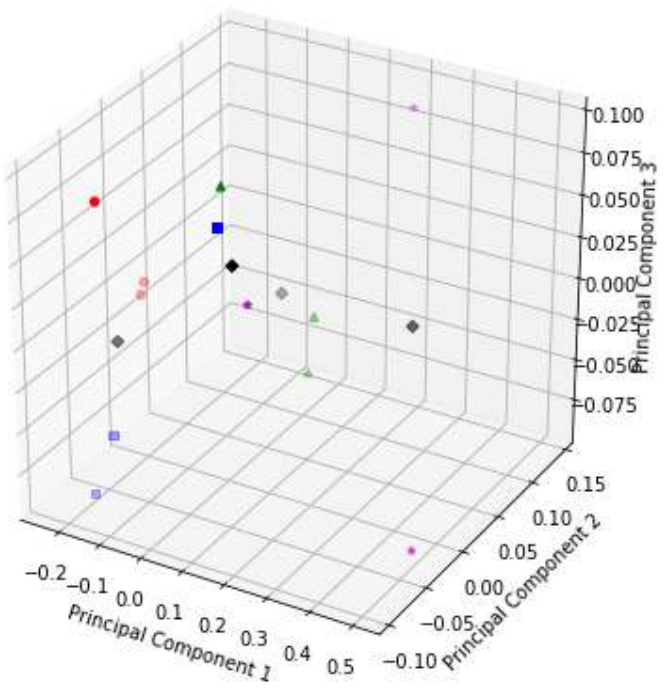
   Plot:



3. Average energy computed using: Mean of squares
   Classification algorithm: K-means
   Initial centroids passed: 7,10,11,12
   Accuracy: 7/12
   Results:

```
label 0 : [1, 2, 5, 7, 8]
label 1 : [9, 10]
label 2 : [3, 11]
label 3 : [4, 6, 12]
```

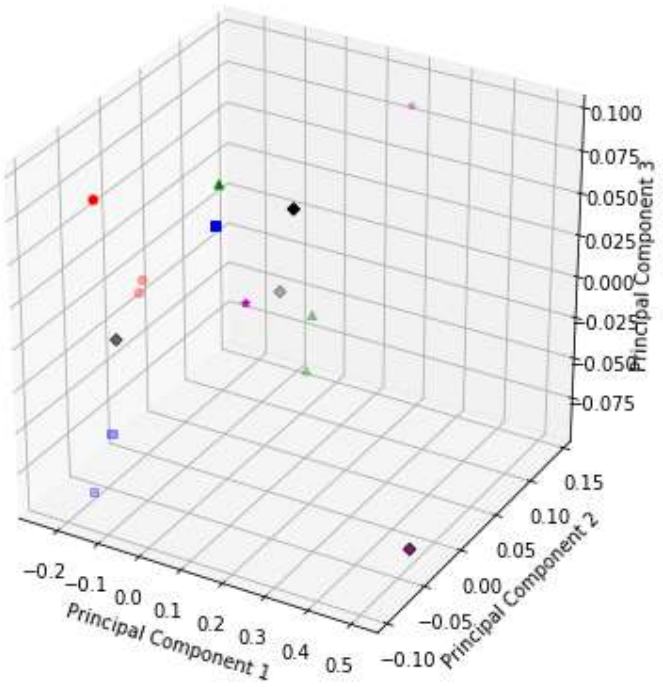
   Plot:



4. Average energy computed using: Mean of squares
   Classification algorithm: K-means
   Initial centroids passed: None
   Accuracy: 8/12
   Results:

```
label 0 : [9, 10, 11]
label 1 : [4, 6, 12]
label 2 : [3]
label 3 : [1, 2, 5, 7, 8]
```

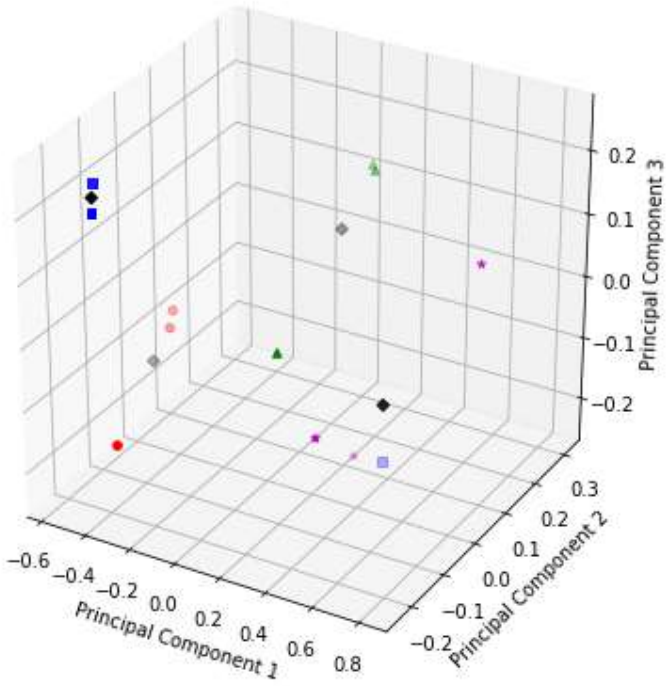
   Plot:



5. Average energy computed using: Mean of Absolute Values
   Classification algorithm: K-means
   Initial centroids passed: None
   Accuracy: 11/12
   Results:

```
label 0 : [4, 6, 12]
label 1 : [3, 9, 10, 11]
label 2 : [2, 8]
label 3 : [1, 5, 7]
```

   Plot:

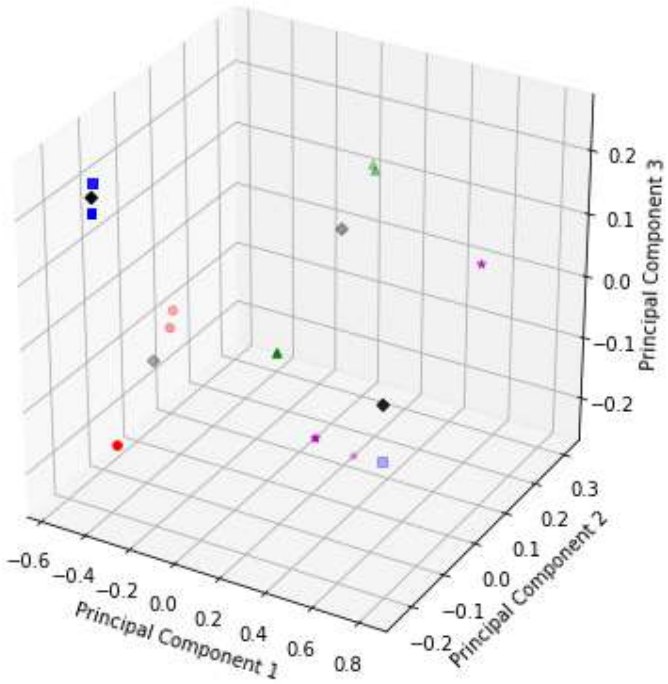


6. Average energy computed using: Mean of Absolute Values
   Classification algorithm: K-means
   Initial centroids passed: 1,2,3,4
   Accuracy: 11/12

   Results:

```
label 0 : [1, 5, 7]
label 1 : [2, 8]
label 2 : [3, 9, 10, 11]
label 3 : [4, 6, 12]
```

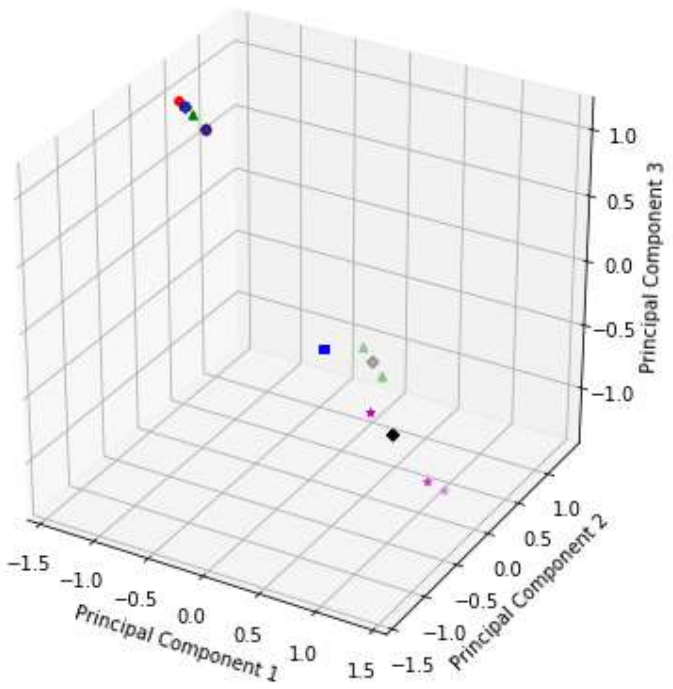
   Plot:



7. Average energy computed using: Mean of Square Values
   Additional normalization before k-means too
   Classification algorithm: K-means
   Initial centroids passed: 5,8,9,6
   Accuracy: 7/12
   Results:

```
label 0 : [2, 5, 7]
label 1 : [1, 8, 12]
label 2 : [3, 9, 10, 11]
label 3 : [4, 6]
```

   Plot:



4. Discussion:

- The best result for the mean of squares approach is by using k-means with initial centroids for textures 5,8,9,6 or textures 1,2,3,4. As can be seen from the plot these two groups of textures act as pairs plotted in 3D space and can be easily clustered if the initial centroids are either points from these pairs. A total of 9 out of 12 were obtained.
- Textures 1,2,3,4 have the original contrast whereas in textures 5,6,8,9, the dark is made darker and the light is made lighter. The remaining however are made darker. Therefore, centroids passed from the above-mentioned quartets have a much better effect than the last quartet.
- The mean of absolute approach is used to reduce the computation time while closely approximating the mean of squares approach. A better result (11/12) is also obtained since by taking the absolute value, the accuracy does not get stretched by a non linear factor.
- The lack of sufficient data points to ensure a good accuracy also affects the output classification. The training data is not completely representative of its class.
- Accuracy is also affected by uneven changes in contrast which imbalances the equilibrium of illumination (the histogram for all textures of one class become skewed towards the darker or brighter side)
- The **discriminating power** of the feature is given by the variance of all elements in that feature vector. The lesser the variance, the more the discriminating power. Maximum variance is for L5L5. Therefore, it has the least discriminant power. The

minimum variance is for S5S5 (S5W5 has second least too). So it has most discriminant power.

(b)

Texture Segmentation

1. **Abstract and Motivation:**

   Texture segmentation is the process of identifying the different textures in an image and segmenting them as per type. Whereas in classification, we classify different texture samples, here we segment different textures within the same sample. Previous approaches included checking images for change in pixel intensity using a window-based approach and trying to use histogram based methods but both these methods did not yield the best results since they included the locality of pixel variation was not wide enough to aid distinguishability. Here, we use the Law filters to extract the features again and then use PCA and k-means to segment the image *pixel-wise,* as opposed to texture wise in classification.

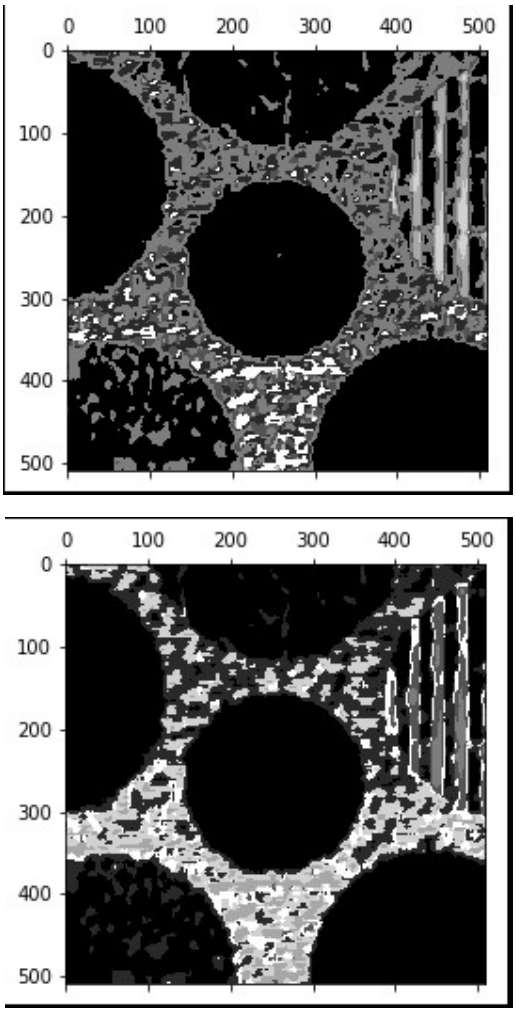
2. **Approach and Procedures:**

   In this case, the segmentation is pixel wise whereas in the classification case, classification is image wise.

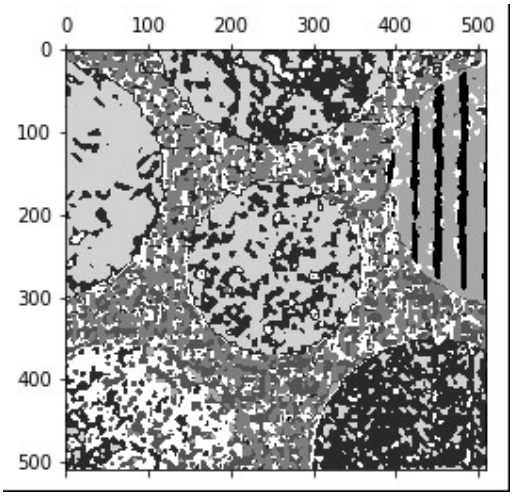
   Algorithm:

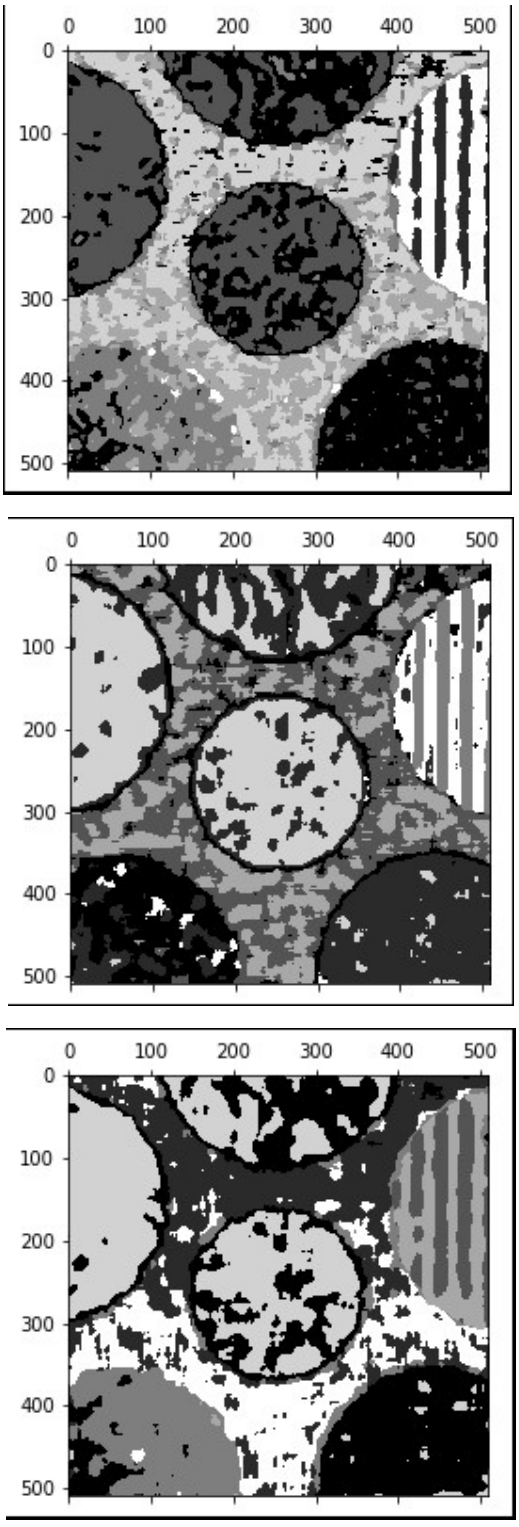
   1. Neighbour pad the input image by 2.
   2. Apply Law filter to get filtered image.
   3. Boundary pad as per window size.
   4. Get the average energy using the window method.
   5. Once the average energy is obtained for all pixels, flatten the array and stack it to a sample x feature array.
   6. Repeat for all 25 dimensions.
   7. Normalize the array using L5L5 or feature normalization.
   8. Pass the matrix to k-means for classification.
   9. Reshape the labels and assign arbitrary gray level intensities.

3. **Results:**



Using squares method to compute average energy: 9,13. Normalization using L5L5.





Alternate energy computation.*: 9,11,15,17. Normalization using L5L5 and additional feature normalization.*

4. **Discussion:**

Effect of window size:

- In this case, the segmentation does not yield best results without an appropriate window size.
- A smaller window would mean a smaller neighbourhood in consideration. This will hamper with the segmentation labels. Too large a window would mean consideration of other texture pixels. Therefore an optimum size is needed, which in this case would be around 25.

Computational Complexity:

- Window sizes affect computational complexity greatly when the window is implemented using two for loops inside the iterating for loops.
- In this case, it taken a long time to implement a window of greater than 13.
- However, computational complexity reduces by almost half when broadcasting and slicing operations are used instead of iterating loops.

The dataset:

- The data set is a heavy 510 x 510 data points with 25 dimensional features.
- Although there are enough data points for training, they might not be correlate depending on the window size in consideration.
- Therefore, this might affect the segmentation result.

Segmentation:

- The dataset is segmented based on 7 texture labels.
- The segmentation algorithm used is k-means with no initial centroids passed.
- The problems faced in segmentation were that although the segmentation was happening, there seemed to be a problem in the classification of the segments as the patterns were distinct but similar and correlated.
- Therefore, a slight adjustment to the energy computation and additional normalization before k-means was incorporated into the algorithm resulting in an efficient change in segmentation.

(c)

Advanced Segmentation Techniques

1. **Abstract and Motivation:**

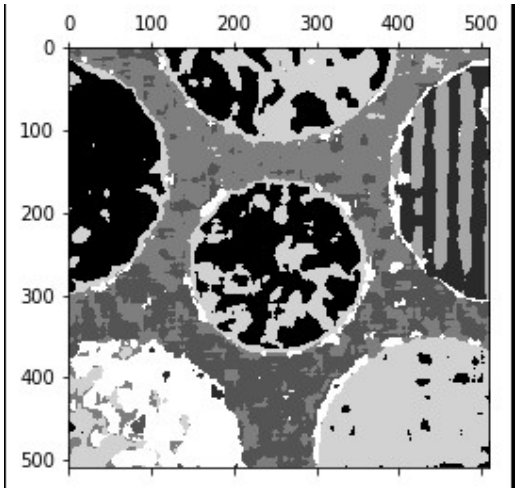
The previous result, while providing a decent visual aid to differentiate the textures, was not conclusive evidence pixel wise. In this section, the goal is to develop pre processing and post processing techniques to augment the result so as to eliminate any ambiguity in the pixel wise segmentation.

2. Approach and Procedures:

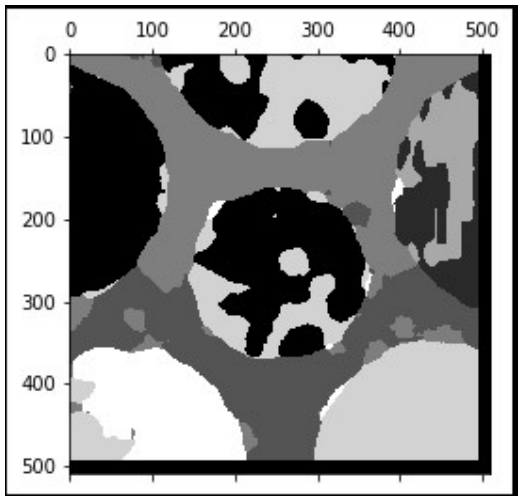
In addition to the output in the last stage, the following two approaches were considered:

1. PCA: this can be used to convert the sparse matrix of data into a dense matrix (cleaning) and thus classify according to principal components.
2. Morphological mask: count the frequency of pixels in the window. Assign that pixel the maximum frequent element.

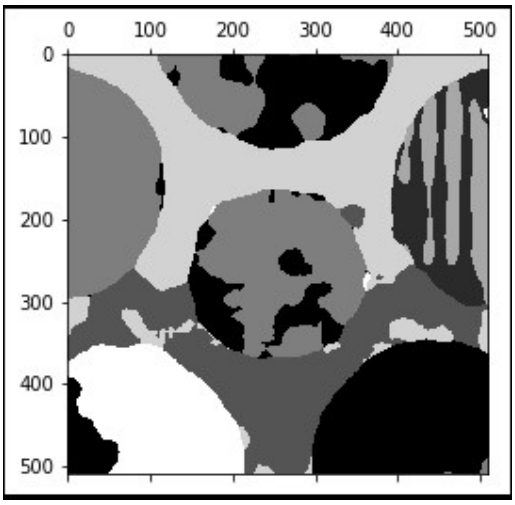
3. Results:



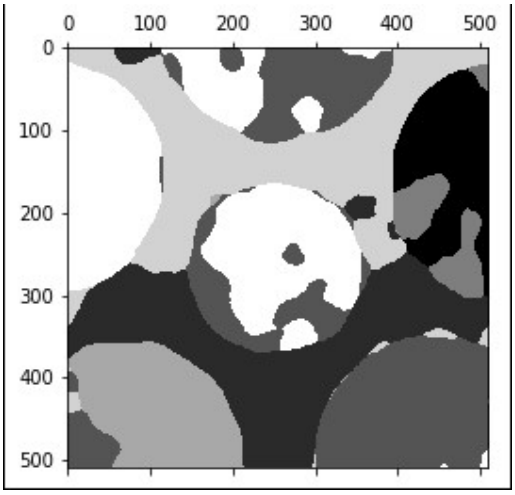
Window size 15, after PCA



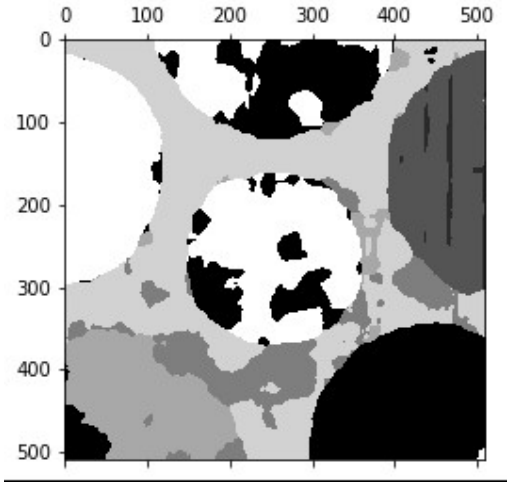
Morphological mask window is 15. Window size for segmentation is 15.



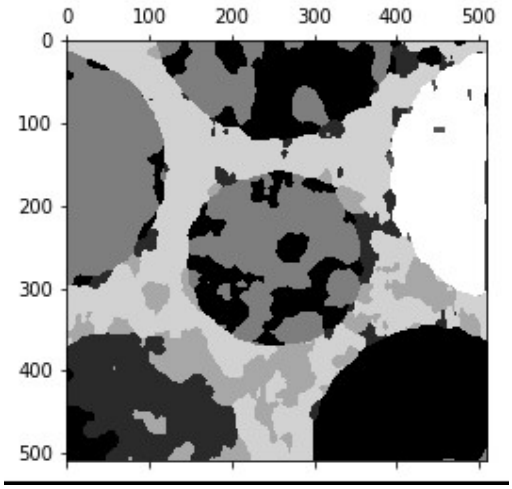
Too large window size for the morphology mask.



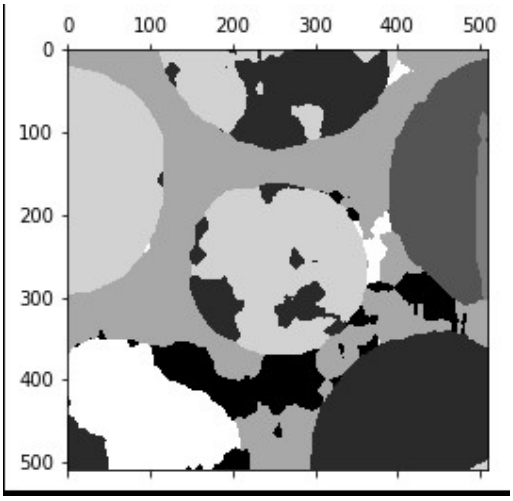
Segmentation window 25 and morphology window 19



*Segmentation window 9 and morphology window 19 (**Best case**)*



Segmentation window 9 and morphology window 13



Segmentation window 9 and morphology window 25

Segmentation Accuracy : 5/7 best case

4.Discussion:

- Here, two window sizes are taken into consideration: The window size for segmentation and the window size for morphology.
- The window sizes are not exactly dependent on each other but a larger segmentation window would mean greater consideration of the neighbourhood, although it has to be kept optimum.
- The window size for the morphological mask ensures:
    1. The boundaries between the textures are smoothened

2. The textures are more uniformly segmented.
3. Stray colors inside native colors are eliminated.