Texture Classification and Representation using 2D Textons

Tanmay Randhavane, 110050010

Nilesh Kulkarni, 110050007

Deepali Adlakha, 11d170020

The Algorithm

- Classification
  Training
  - Input: One texture image (m X n) per texture class, q texture classes, a filter bank having p filters.
  - Output: Histogram Database
  - Algorithm:
    - For an image, for each pixel, convolute filters from filter bank to get a vector of size p.
    - Apply k-means algorithm on all m X n vectors, to get k vectors each of size p. These are called textons, which are particular to an image.
    - Generate k textons in the same way for all the texture classes. Thus, we get a texton database of qk textons.
    - Now, consider an image of a texture class. We have a p-length vector for each pixel in this image. Find the closest texton to this vector in the texton database using euclidean distance. Create a histogram of this mapping with the textons in the texton library on the x-axis.
    - Thus, we get a database of q histograms which we will use for classification.

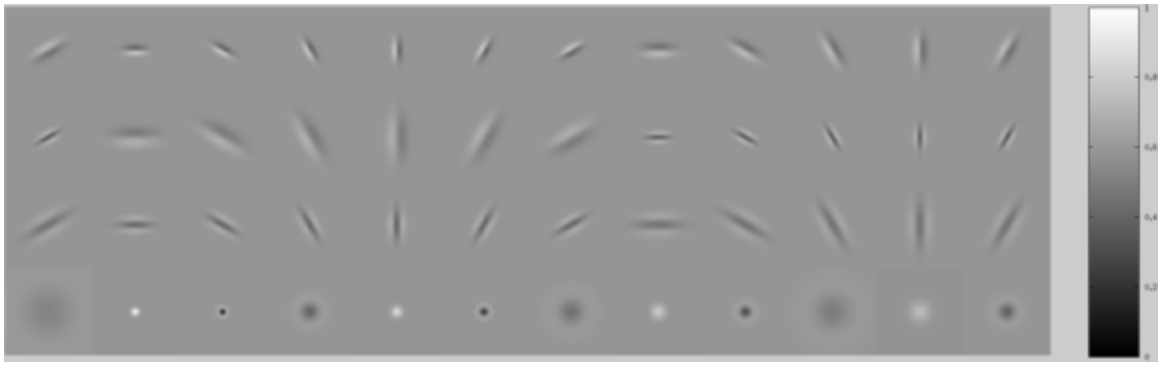
  Testing
  - Input: An image, texton database, histogram database, filter bank
  - Output: The texture class to which the image belongs
  - Algorithm:
    - For each pixel, apply filters from filter bank to get a vector of size p.
    - Find the closest texton to this vector in the texton database using euclidean distance. Create a histogram of this mapping with the textons in the texton library on the x-axis.
    - Find a histogram closest to test histogram from histogram database using chi-square distance. The image belongs to this texture class.

- Texture Reconstruction
  - Input: Textons generated for a texture image, mapping of each pixel to a texton, filter bank
  - Output: Reconstructed texture image
  - Algorithm:
    - Pseudo-inverse of filters from filter bank is applied on each texton to get a patch associated with that texton.
    - For each pixel, get the mapped texton (Using the mapping we get from K-Means). Assign the intensity of the center pixel in the patch associated to the texton at that pixel.

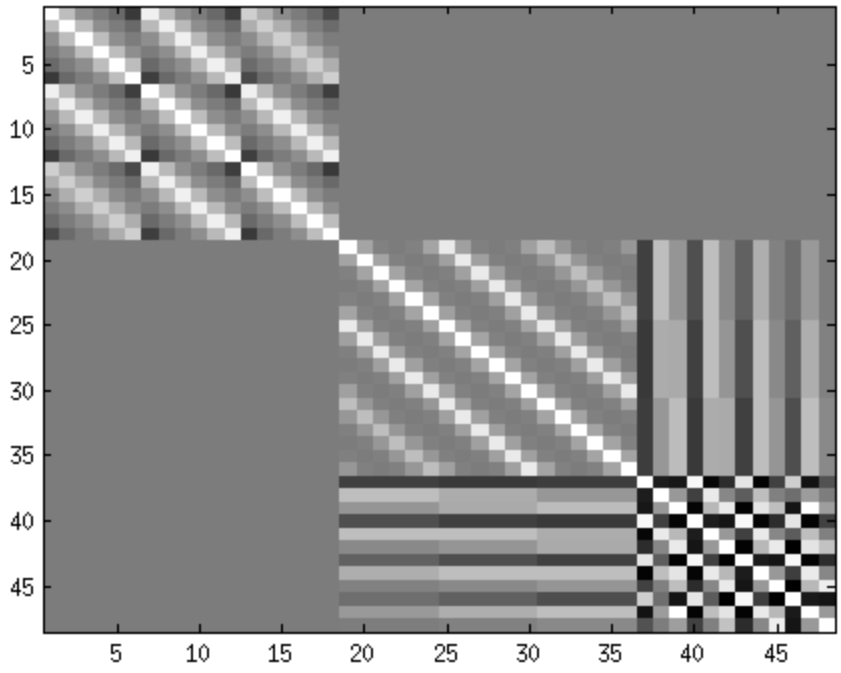
Results

- Visualization of the three filter banks

LM Filter Bank

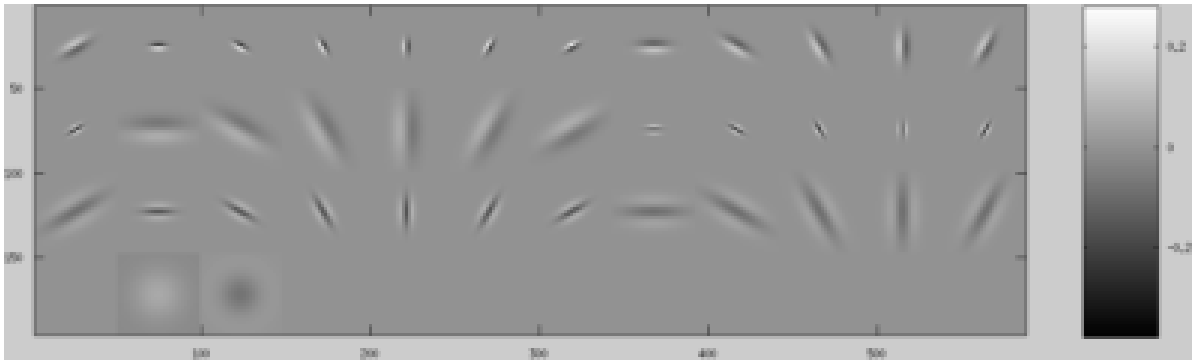


Filter Visualization

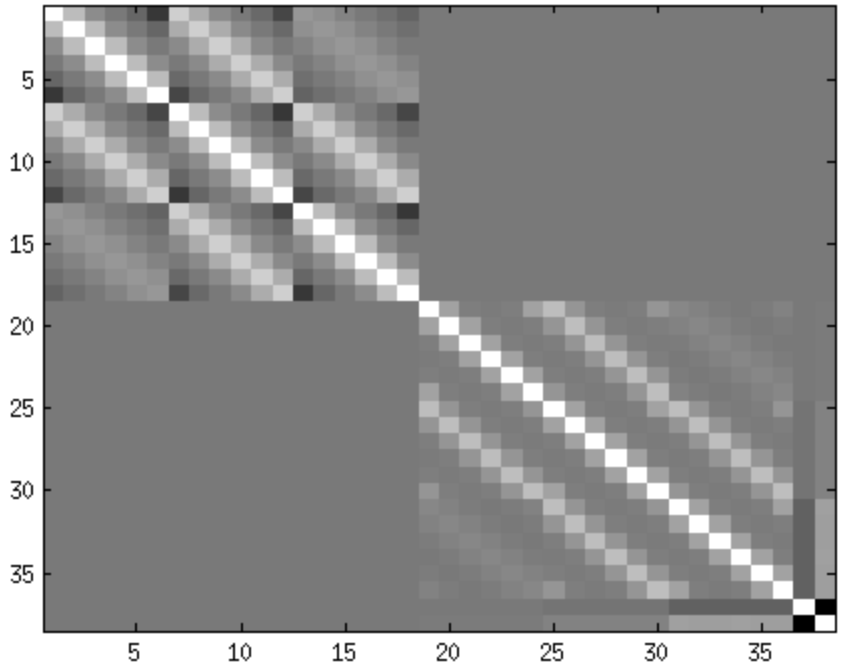


Filter Correlation Matrix

RFS Filters (MR8 is derived from RFS)

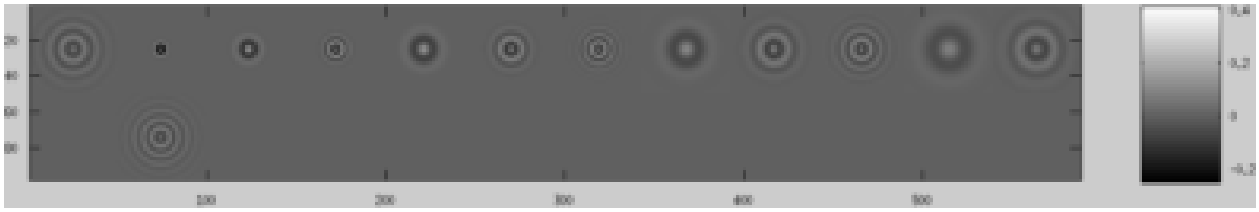


RFS Filter Bank Visualization

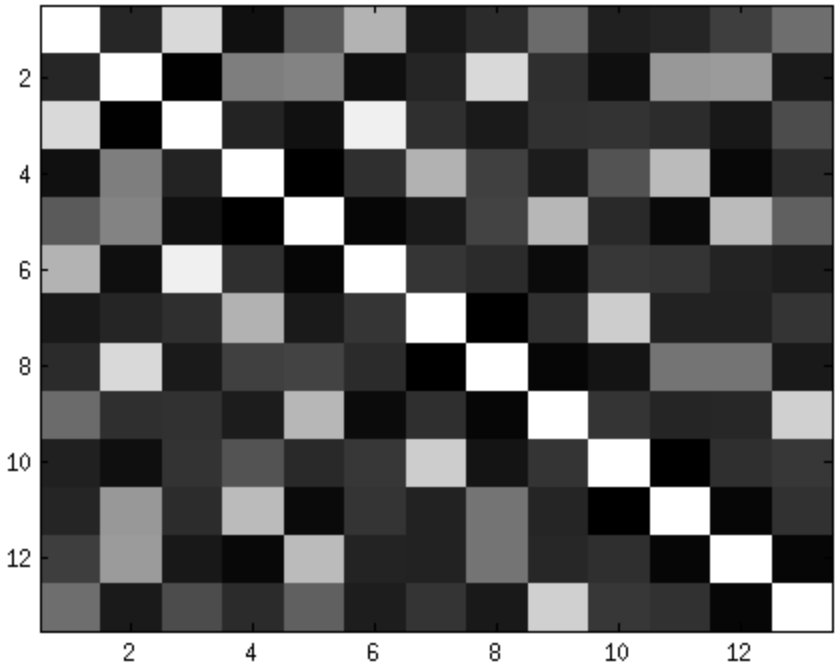


RFS Filter Correlation Matrix

S Filter

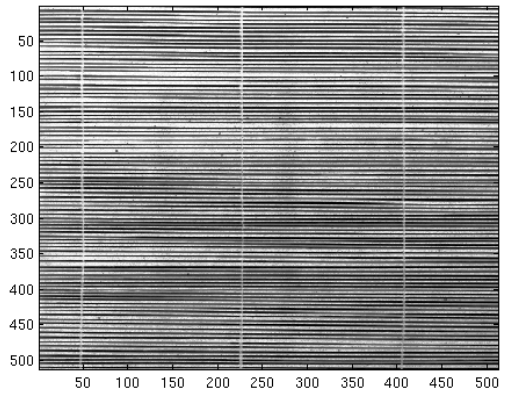
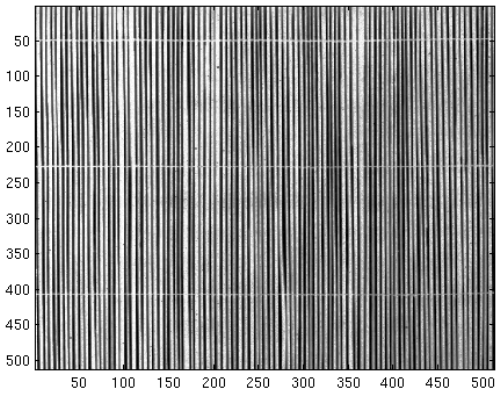
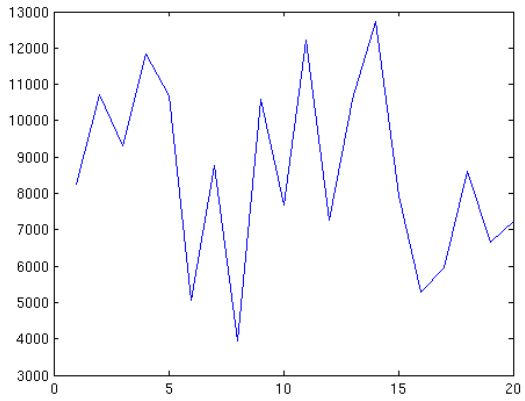
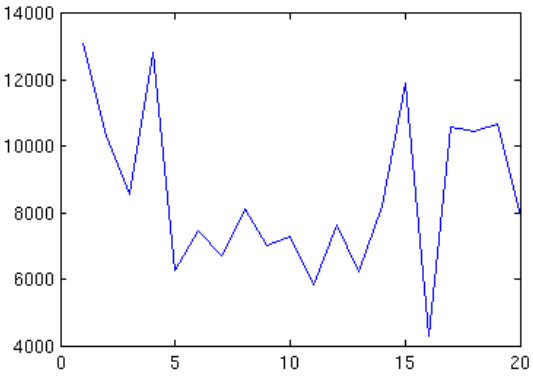
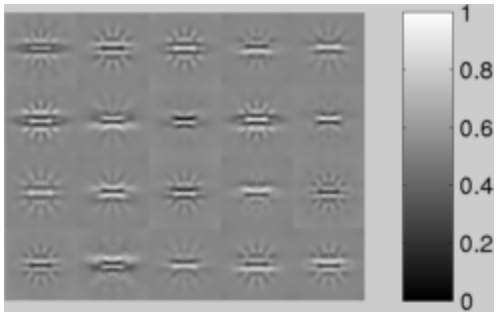
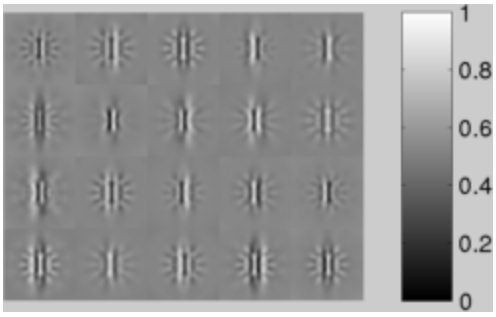


S Filter Visualization

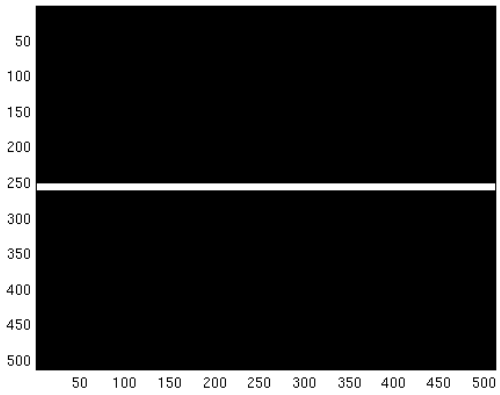
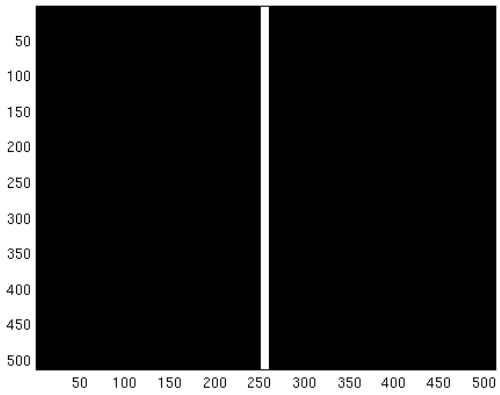


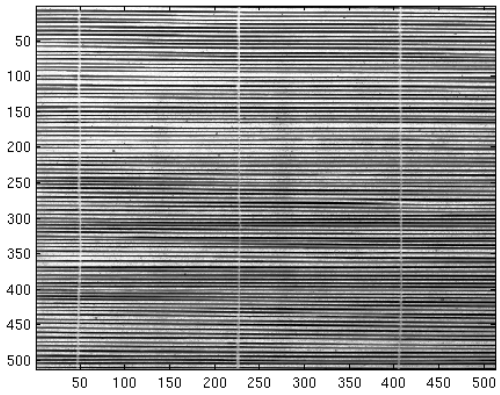
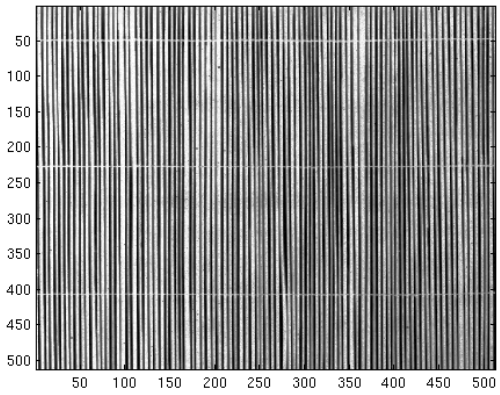
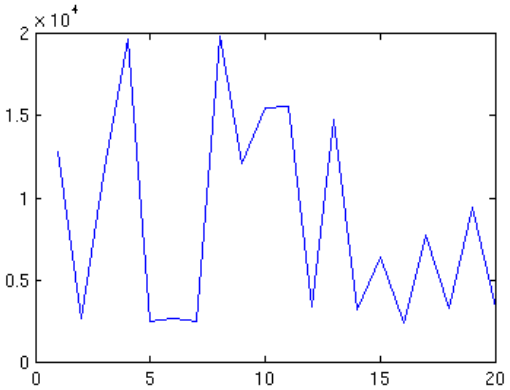
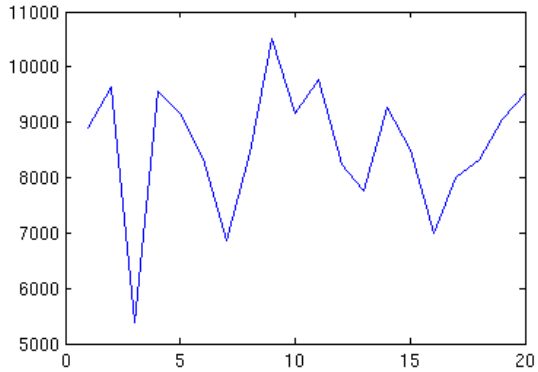
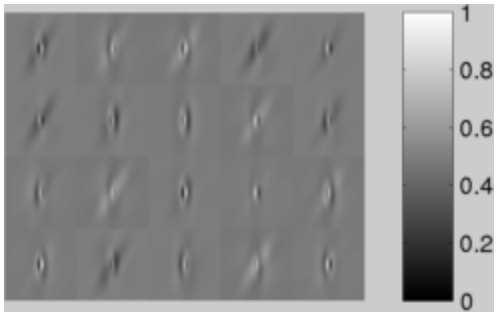
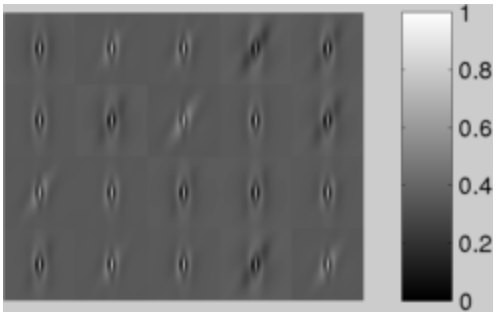
Correlation Matrix S Filter

- Rotational Variance of LM filters

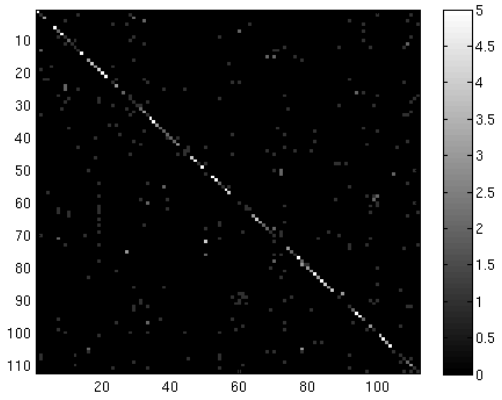
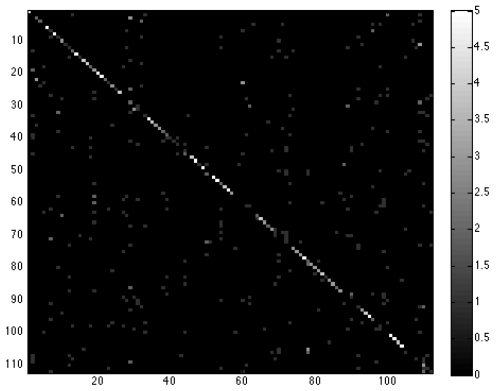
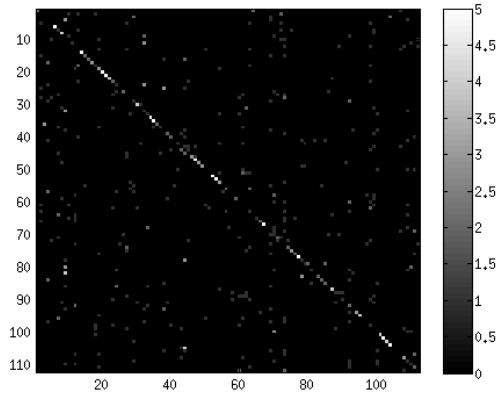
Horizontal	Vertical
	
	
	

- Rotational Invariance of MR8 filters

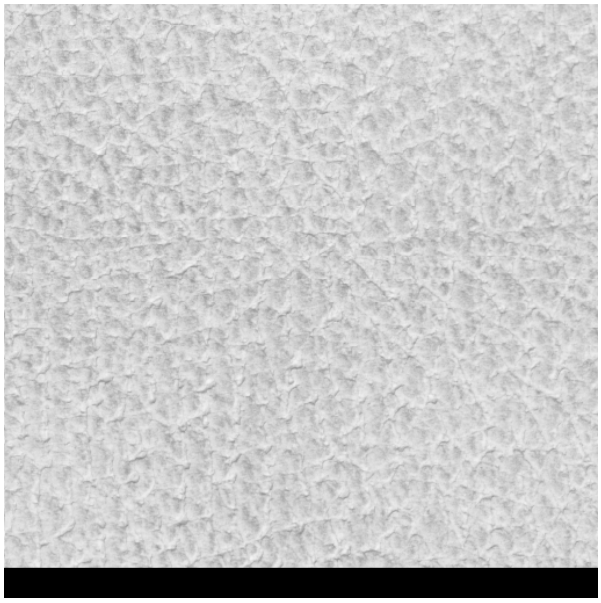
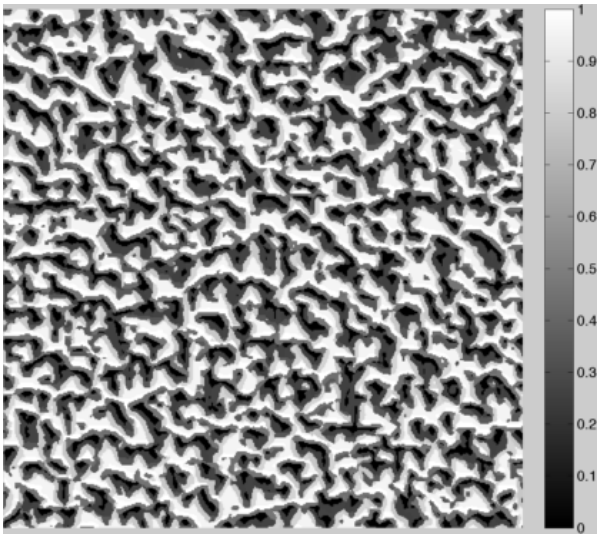
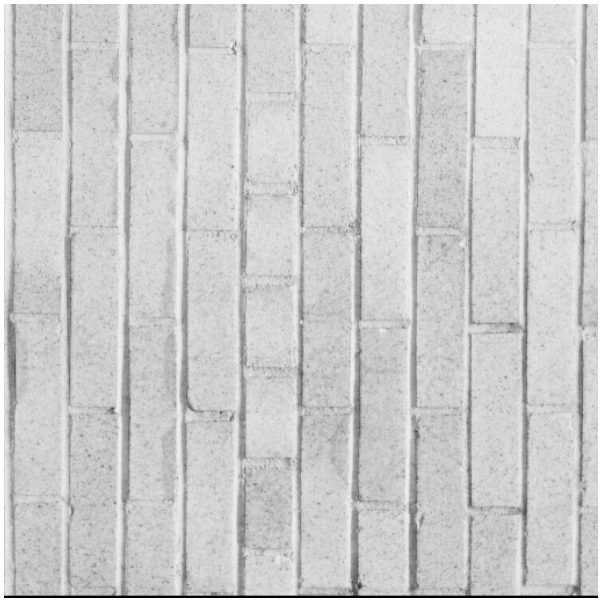
Horizontal	Vertical
	
171396	171396
	

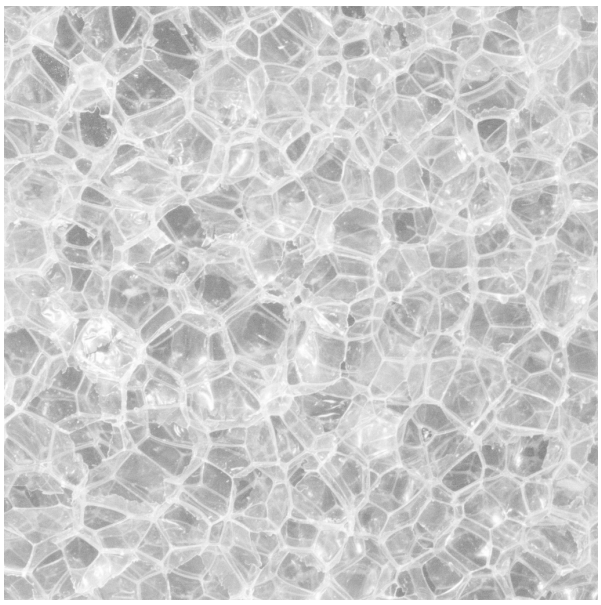
Horizontal	Vertical
	
	
	

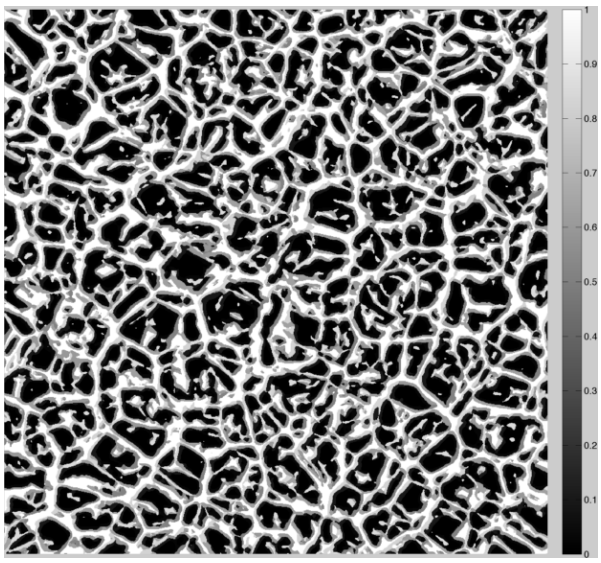
- Confusion Matrix for three filter banks

LM Filter Bank: Accuracy = 43.3929%	MR8 Filter Bank: Accuracy = 41.6071%
	
S Filter Bank Accuracy: 31.7857	
	

Texture Reconstruction

Original	Reconstructed
	
	




1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

References

http://www.cs.berkeley.edu/~malik/papers/LM-3dtexton.pdf
(We implemented the 2D texton classification problem from the following paper)

Datasets & Filter Banks

1) Dataset to train and test texture model : Brodatz Texture Database
(http://sipi.usc.edu/database/database.php?volume=textures)
2) Filterbank for 2D texton extraction : The Leung-Malik Filterbank
(http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html)