**Electrical and Computer Engineering, Texas Tech University**

**ECE 5332-011: Deep Learning for Medical Signal/Image data**

Alexandre Soares da Silva, R11485685

Rishi Das, R11485682

Jude Inwumoh, R11610597

**1. Basic Image processing**

Matlab script: **DL\_1\_basic\_img\_proc.m**

**1. a. Loading and saving an image**

From the MATLAB image dataset, we were able to load the image onion.png to obtain its RGB display.



Figure 1: RGB Onion.png

This image in figure 1 was converted to gray scale using the **rgb2gray()** function.



Figure 2: onion\_gray.png

**1.b. Introduction to the 2D Fourier space**

The image ‘cameraman.tif’, loaded from MATLAB’s image database, is transformed using the 2D discrete Fourier transform.



Figure 3: 2D Fourier

With the aid of **fftshift()** function, low frequency content found in cameraman.tif with the 2D DFT is shifted to the center of the image, while higher frequency content expands from the image center to its border. The pixel intensities are scaled down for display using **log().**

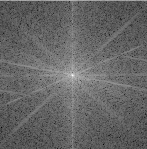


Figure 4: 2D Fourier Space

Two different binary masks are applied to the frequency-domain image: a low pass mask and high pass mask, each using radii between 5 and 50 pixels. The cameraman.tif image is reconstructed from the masked (filtered) frequency-domain images.

It can be observed that the low pass filter with the largest radius produces the least blurring, whereas the low pass mask with the smallest radius, the most blurring. Notice, however, the artifacts (ripples) caused the ideal filters, which happens both for the low pass and the high pass mask.

When a high pass mask is applied, the image loses low frequency content (slower changes in pixel intensity), while what remains is mostly sharp changes in pixel intensity (edges). Finally, the high pass mask has the most extreme effect when it is used with its largest radius (50 pixels), with no more than shapes remaining on the grey almost solid background, with some artifacts again produced by the idealness of this filter (binary masks instead of smooth progression of intensities).

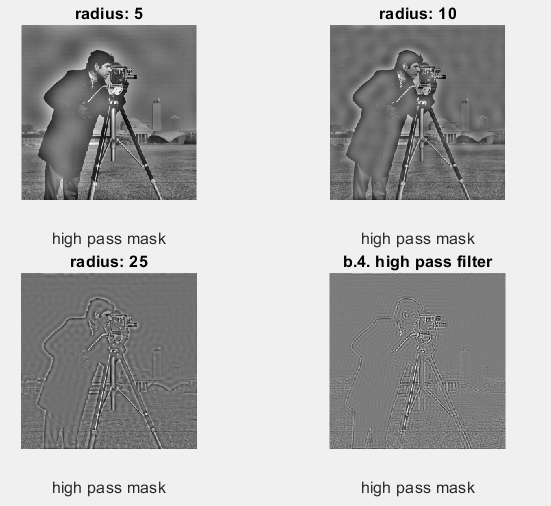
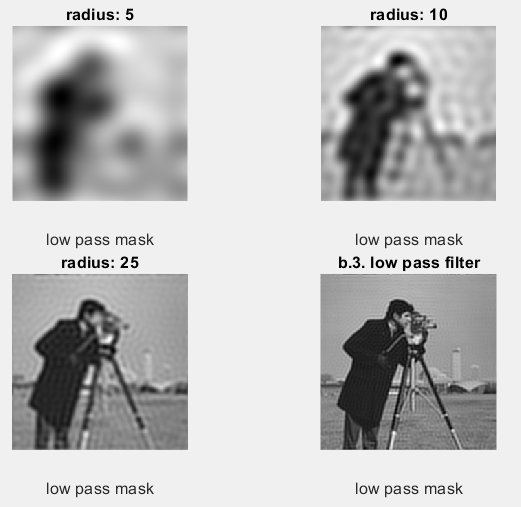
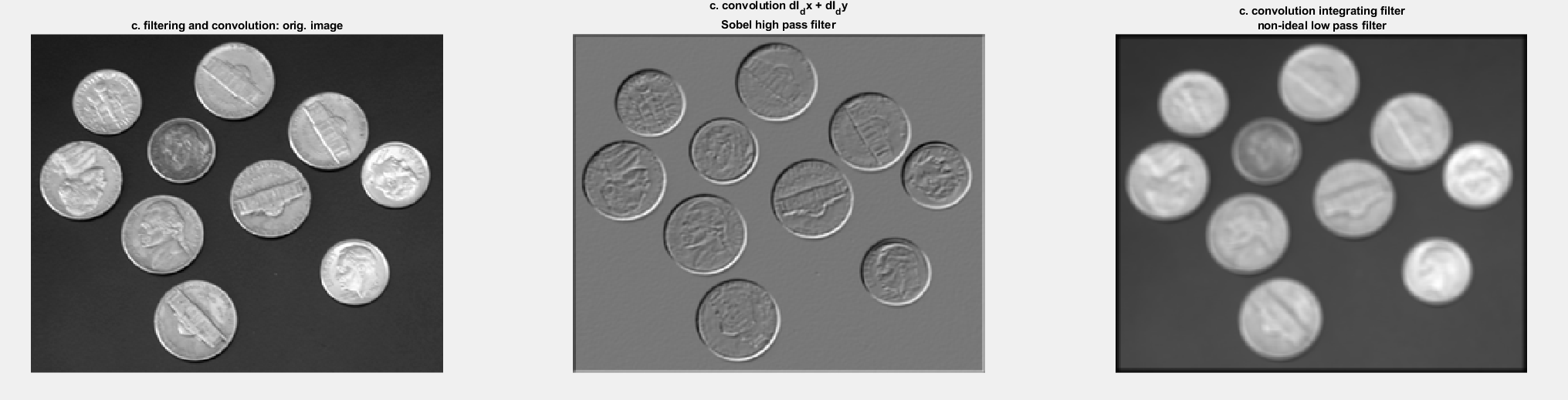


Figure 5: Display of the low and high pass masks with different radius.

**1.c. Spatial filtering and convolution**

The 2D spatial filtering with nonideal high pass (Sobel) and low pass (integrating) filters performed better than the ideal masks, causing no visible artifacts in the images. The testing was done on a coins.png image using two 3x3 Sobel kernels to obtain the gradients on the x and y directions, and the outcome are the pronounced edges. The low pass spatial filter, a 5x5 integrating kernel ( 1/25\*[xi,k] where x=1 and i,j ranges are 1<=i, j<= 5).



**2. Clustering**

Matlab script: **DL\_2\_clustering.m**

**k-means Clustering**

k-means clustering is a type of unsupervised learning, where similar data is grouped based on their underlying structure into cohorts or clusters. K represents the number of clusters the data points are going to be classified into. This technique is defined by an objective function that tries to minimize the sum of all Euclidean distances within a cluster, for all **k** clusters.

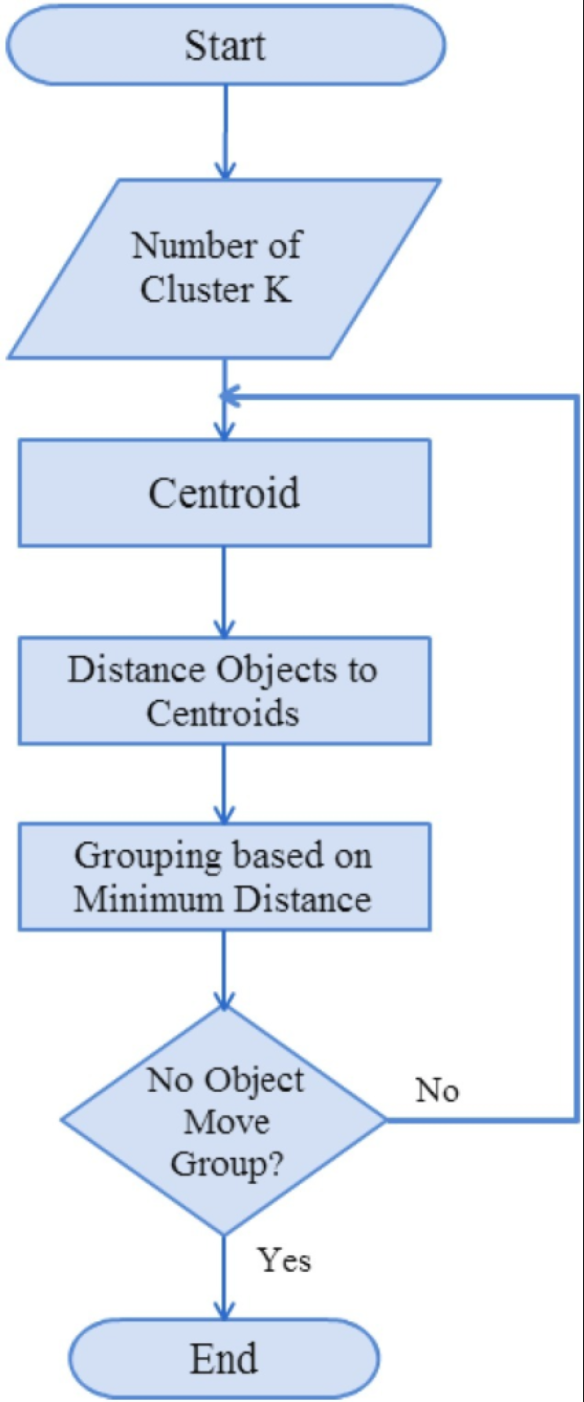


Figure 6: The figure above shows the flowchart that is followed to perform k-means clustering.

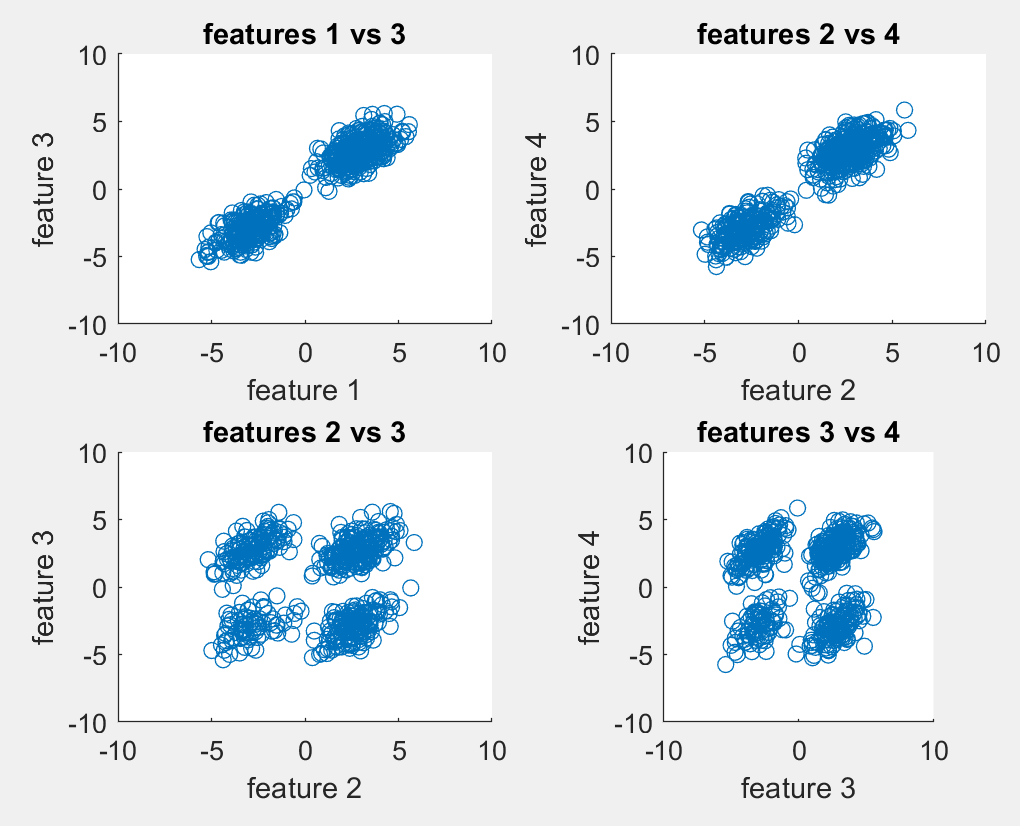


Figure 7: Estimation of number of clusters by plotting dimensions against each other.

From the dataset obtained from ‘kmeans.mat’, we observed 4 dimensions, each containing 560 feature vectors. After plotting all dimensions against each other, we can observe that it consists of 4 separate clusters. Hence, we apply the technique of kmeans and hierarchial on the data with number of classes defined as 4. Morever, to test individual cluster differences and performance, we increased the cluster numbers to 8 and measured the Silhoutte coefficients.

**Silhouette Coefficient:** measures consistency in data within a cluster, by calculating how similar the data is in a cluster compared to other clusters. A coefficinet of 1 would mean the highest similarity. Visual inspection silhouette coefficient plots helps in evaluating the performance of our clustering method.

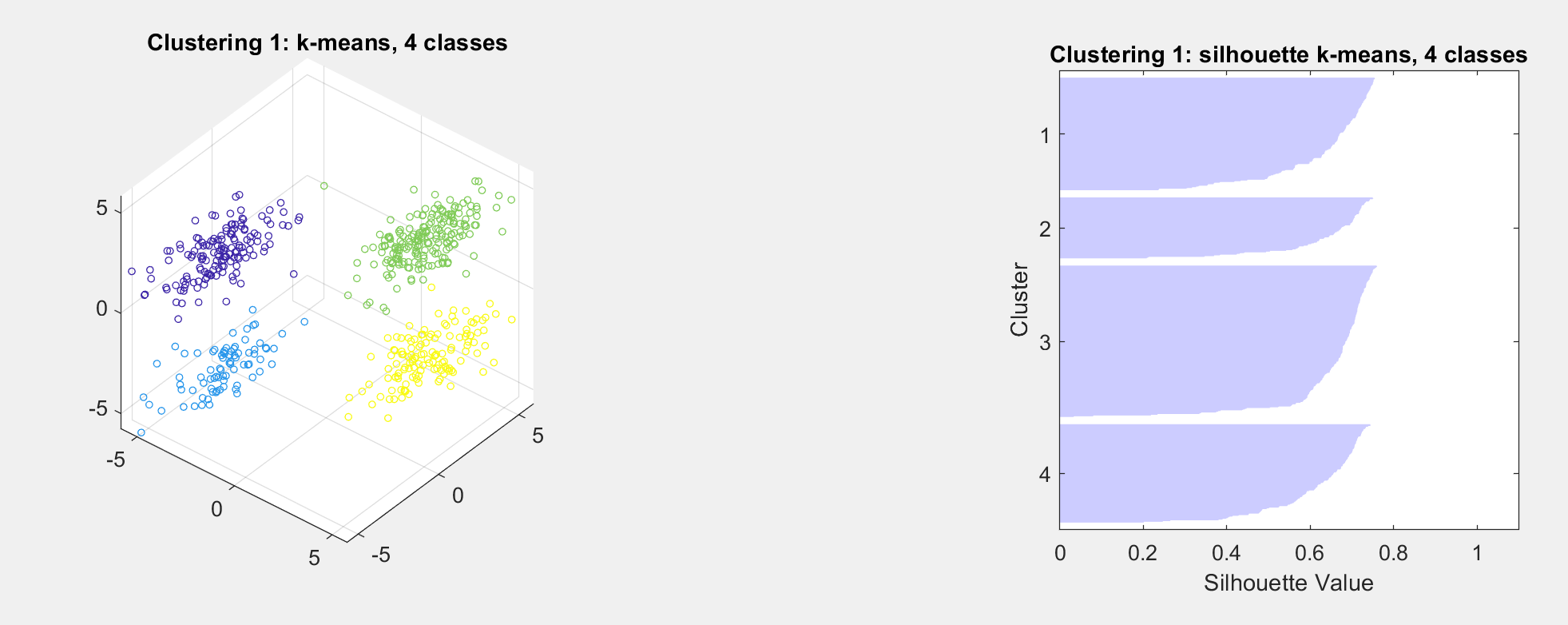


Figure 8:Showing k-means clustering using 4 clusters and its silhouette coefficients for each cluster

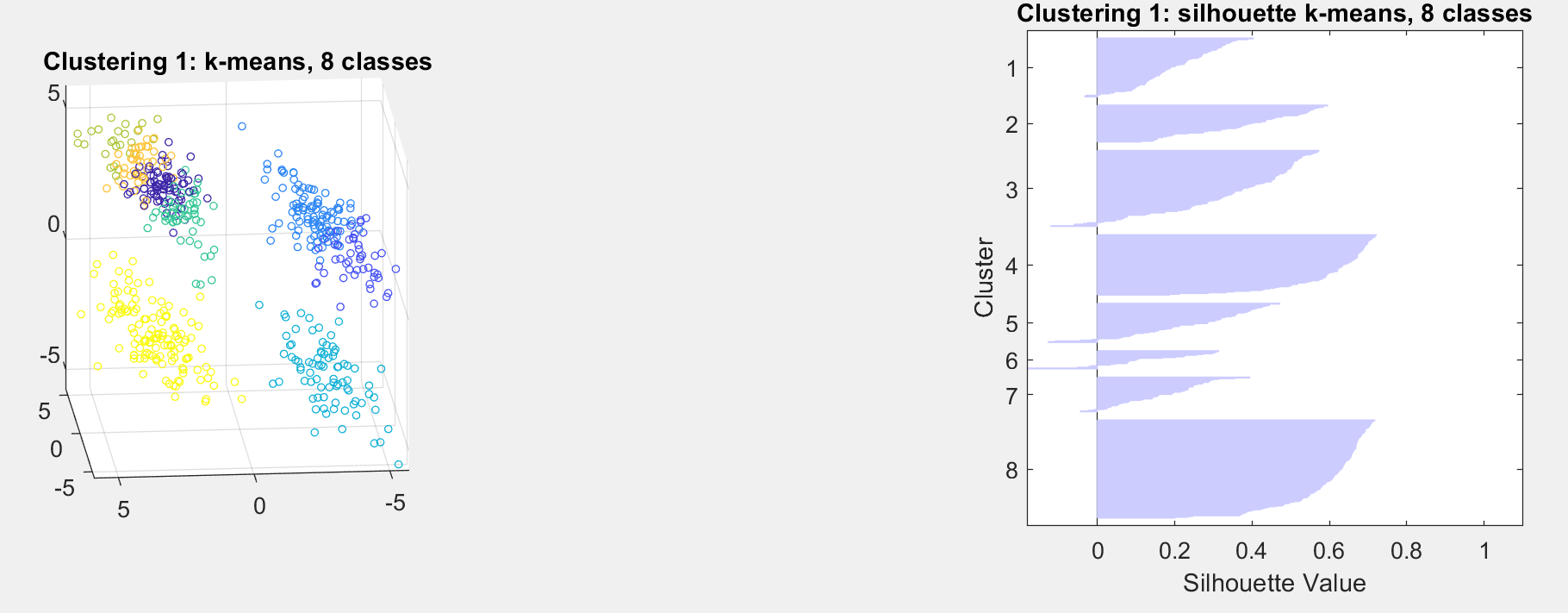
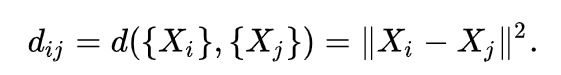


Figure 9:Showing k-means applied to the dataset with k value of 8. Silhouette coefficients are also shown

**Hierarchial clustering**

Hierarchial clustering is an agglomerative clustering technique, where each data points is considered as an individual cluster. At every iteration, each clusters is merged with other clusters, until a numer of clusters defined initially is formed. We calculate the proximity of new clusters and merge similar clusters to form a new cluster.

Ward's minimum variance criterion minimizes the total within-cluster variance.

[2]

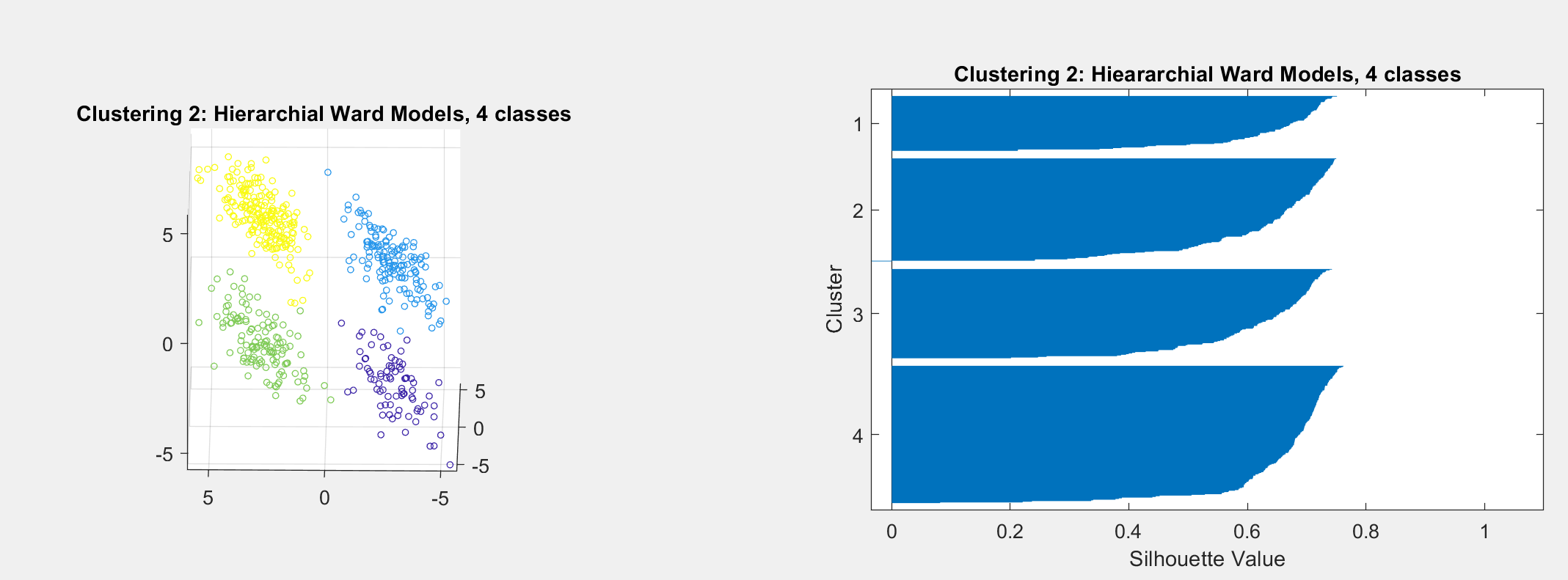


Figure 10: Shows the hierarchial ward cluster for 4 different clusters and Silhouette coefficients.

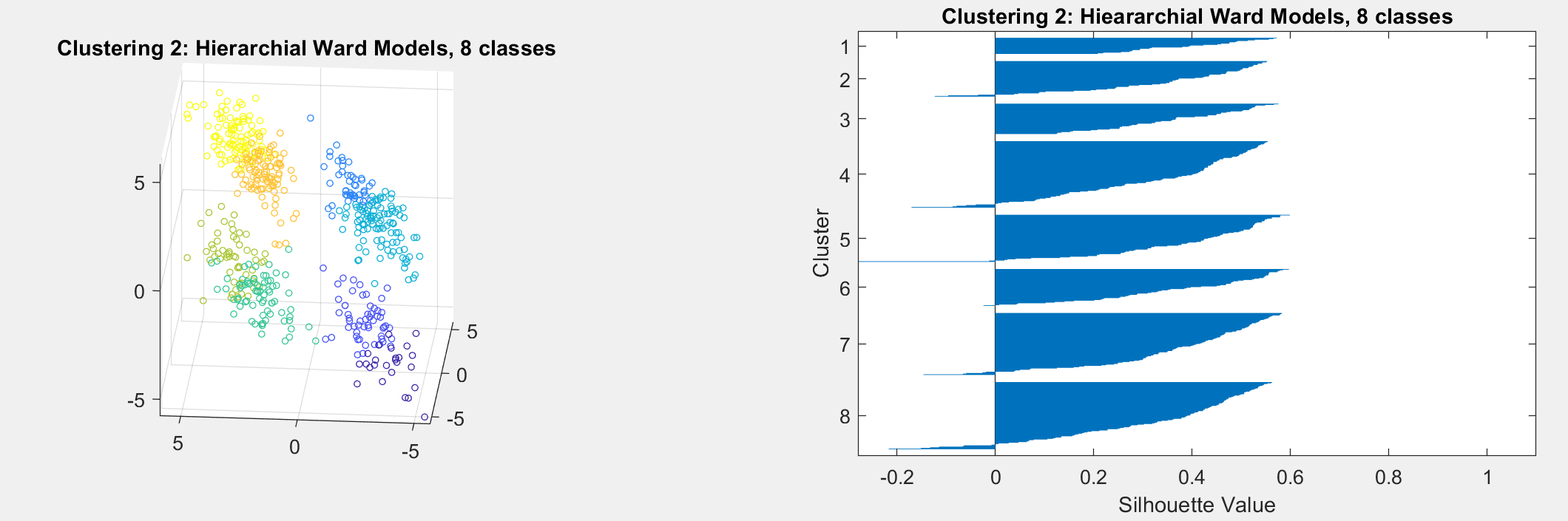


Figure 11: Shows the hierarchial ward clustering for 8 clusters with it’s Silhouette coefficient.

Explanation: From the above figure shown in this paper, for implementing k-means clustering we needed to have an idea for the desired number of clusters we had to pick, we also observe how k-means gives us an unintuitive result as the data in the cluster is not well seperated. It is also important to pick the right k value for our data and how the cluster.

In contrast, hierarchial clustering requires few assumptions when it comes to distribution of data. It requires to measure the distance for a pair of data points. It joins the neighboring similar data into a cluster, and then adds nearby points to the nearest group. The resulting data distribution can be visulaized in sort of a connectivity plot.

**3. Simple Image classification problem**

Matlab script: **DL\_3\_simple\_img\_class.m**

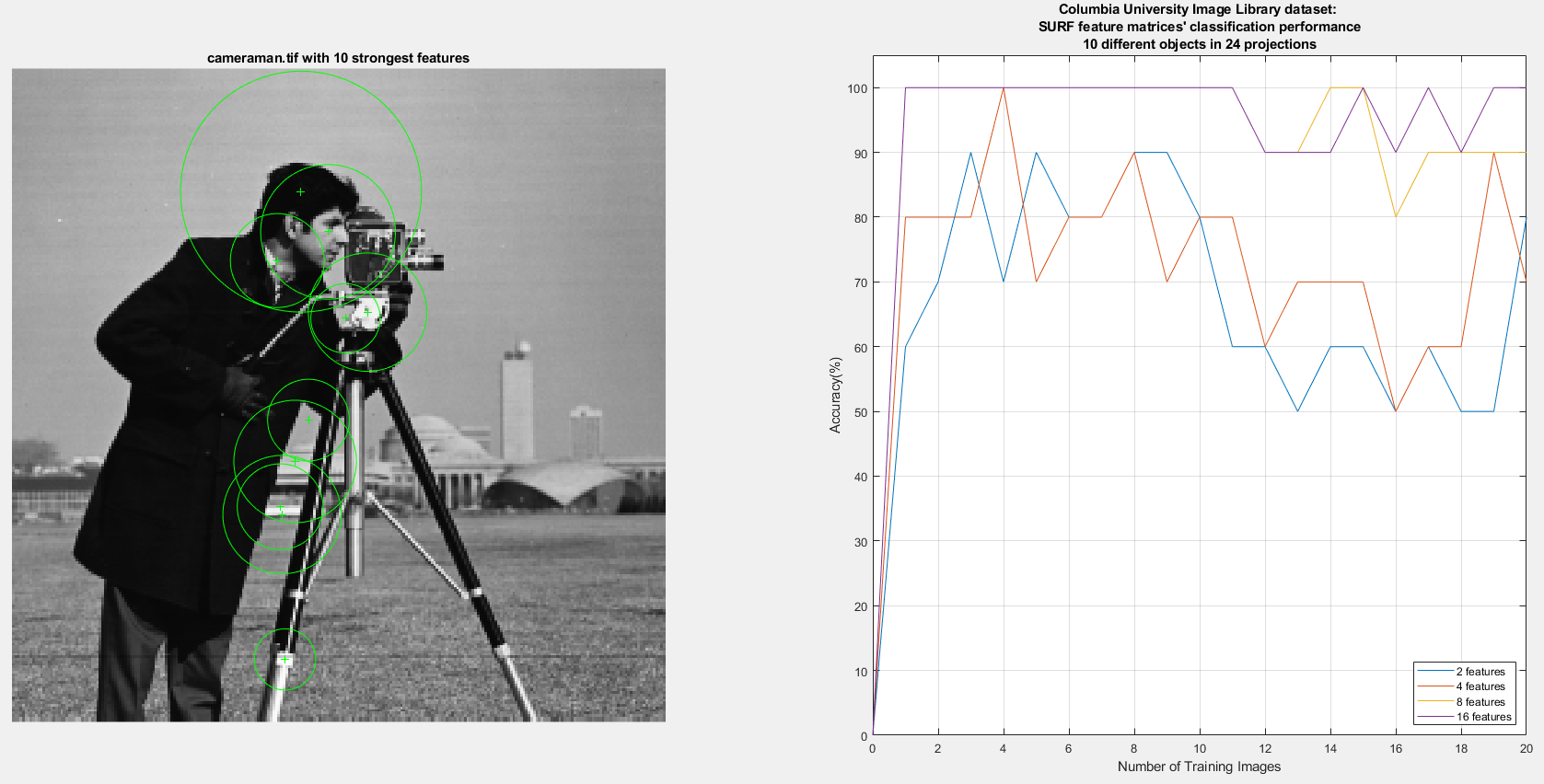
****

Figure 12: SURF features plotted on their regions of detection

The problem statement required the use of the SURF (Speeded-up Robust Features) algorithm to match and classify objects from different perspectives. The assignment’s goal was to evaluate the accuracy of such classification engine, testing different features and varying dataset sizes for training and testing. The dataset used was the Columbia University Image Library (COIL-20) dataset, with selection of 10 objects (3, 4, 5, 6, 9, 10, 12, 13, 14, and 19) in 24 different projections (from 0 to 11 and from 60 to 71).

SURF is a computationally efficient algorithm that approximates Hessians (matrices of second partial derivatives) of Gaussians by using box filters (convolution). The descriptors produced by this algorithm are N x 64 matrices, which are calculated with wavelet responses in x and y directions (gradient components in vector form).

To build the feature matrix used by the algorithm, the script selects up to 20 perspectives of one object to create a train feature matrix. Those can be combined with the selection of 2, 4, 8, or 16 strongest features to obtain the same matrix.

For instance, if the selected number of perspectives of an object is 5 and the number of features selected is 4, functions will retrieve the first 4 SURF descriptors for each of the first 5 images, creating a train feature matrix of 20 rows x 64 columns. The descriptors’ rows are concatenated vertically.

The test feature matrix is comprised of descriptors from the remaining 19 perspectives of the object (from a total of 24). In this case, the test feature matrix is a 76 x 64 SURF descriptor matrix. The train and test feature matrices are matched with the function **matchFeatures()**, and the length of the indexPairs returned by that function corresponds to the number of matches found between train and test. For every combination of train object, number of features, and number of perspectives, feature matrices are matched, and the number of matches is saved. In the end, accuracy is based on train and test objects with the largest number of features matched, and if the predicted object matches the ground truth object.

For our project, this is the plot that reproduces the tests offered in the assignment sheet:

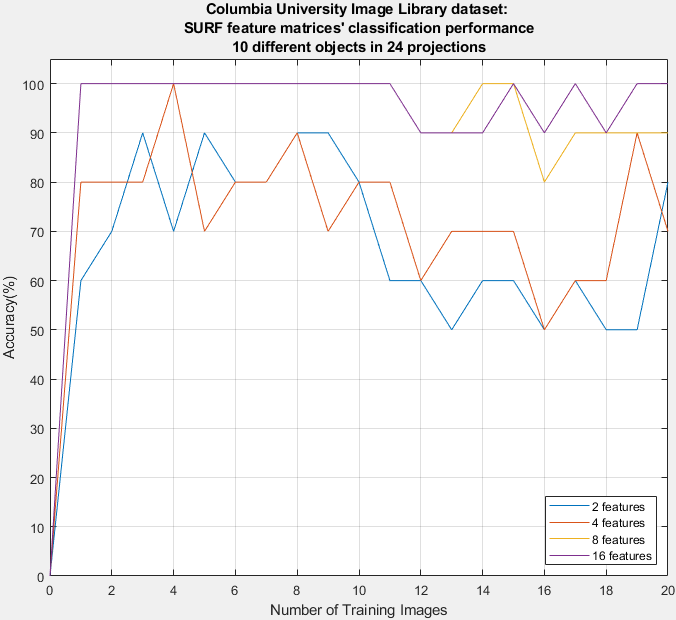


Figure 13: Accuracy curve

**References**

1. S. Aranganayagi and K. Thangavel, "Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure," International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, Tamil Nadu, 2007, pp. 13-17.
2. Bay, H., Tuytelaars, T., & Gool, L.V. (2006). SURF: Speeded Up Robust Features. ECCV.
3. <https://en.wikipedia.org/wiki/Ward%27s_method>