CS435 Assignment 4 - Classification

Harry Chong

2 March 2021

1 Theory Questions

1. Given the following image pixel intensity values:

$$I = \begin{bmatrix} 1 & 2 & 5 & 0 \\ 2 & 2 & 4 & 1 \\ 2 & 2 & 5 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

   (a) For each iteration, assign each observation pixel in I to the reference vector it is closest to and update the reference vectors to be the means of their members.

   Iteration 1: Cluster assignment on I after assigning each value to the initial reference vector a_1 or a_2:

$$I = \begin{bmatrix} a_1 & a_1 & a_2 & a_1 \\ a_1 & a_1 & a_2 & a_1 \\ a_1 & a_1 & a_2 & a_1 \\ a_1 & a_1 & a_1 & a_1 \end{bmatrix}$$

   Sum for all values associated with a_1: 17
   Sum for all values associated with a_2: 14

   Updated reference vectors:
   $a_1 = \frac{17}{13} = \boxed{1.3077}$
   $a_2 = \frac{14}{3} = \boxed{4.6667}$

   Iteration 2: Cluster assignment on I after assigning each value to the updated reference vector a_1 or a_2:

$$I = \begin{bmatrix} a_1 & a_1 & a_2 & a_1 \\ a_1 & a_1 & a_2 & a_1 \\ a_1 & a_1 & a_2 & a_1 \\ a_1 & a_1 & a_1 & a_1 \end{bmatrix}$$

   Sum for all values associated with a_1: 17
   Sum for all values associated with a_2: 14

   Updated reference vectors:

$a_1 = \frac{17}{13} = \boxed{1.3077}$
$a_2 = \frac{14}{3} = \boxed{4.6667}$

Since the cluster assignments did not change, we are done.

(b) Compute the weight matrix W using the following formula:

$$w(a,b) = e^{-((a_i-b_i)^2+(a_x-b_x)^2+(a_y-b_y)^2)}$$

Weight Matrix (W) =

$$\begin{matrix}
0 & e^{-2} & e^{-20} & e^{-10} & e^{-2} & e^{-3} & e^{-14} & e^{-10} & e^{-5} & e^{-6} & e^{-24} & e^{-13} & e^{-9} & e^{-10} & e^{-13} & e^{-18} \\
e^{-2} & 0 & e^{-10} & e^{-8} & e^{-2} & e^{-1} & e^{-6} & e^{-6} & e^{-5} & e^{-4} & e^{-14} & e^{-9} & e^{-11} & e^{-10} & e^{-11} & e^{-14} \\
e^{-20} & e^{-10} & 0 & e^{-26} & e^{-14} & e^{-11} & e^{-2} & e^{-18} & e^{-17} & e^{-14} & e^{-4} & e^{-21} & e^{-29} & e^{-26} & e^{-25} & e^{-26} \\
e^{-10} & e^{-8} & e^{-26} & 0 & e^{-14} & e^{-9} & e^{-18} & e^{-2} & e^{-17} & e^{-12} & e^{-30} & e^{-5} & e^{-19} & e^{-14} & e^{-11} & e^{-10} \\
e^{-2} & e^{-2} & e^{-14} & e^{-14} & 0 & e^{-1} & e^{-8} & e^{-10} & e^{-1} & e^{-2} & e^{-14} & e^{-11} & e^{-5} & e^{-6} & e^{-9} & e^{-14} \\
e^{-3} & e^{-1} & e^{-11} & e^{-9} & e^{-1} & 0 & e^{-5} & e^{-5} & e^{-2} & e^{-1} & e^{-11} & e^{-6} & e^{-6} & e^{-5} & e^{-6} & e^{-9} \\
e^{-14} & e^{-6} & e^{-2} & e^{-18} & e^{-8} & e^{-5} & 0 & e^{-10} & e^{-9} & e^{-6} & e^{-2} & e^{-11} & e^{-17} & e^{-14} & e^{-13} & e^{-14} \\
e^{-10} & e^{-6} & e^{-18} & e^{-2} & e^{-10} & e^{-5} & e^{-10} & 0 & e^{-11} & e^{-6} & e^{-18} & e^{-1} & e^{-13} & e^{-8} & e^{-5} & e^{-4} \\
e^{-5} & e^{-5} & e^{-17} & e^{-17} & e^{-1} & e^{-2} & e^{-9} & e^{-11} & 0 & e^{-1} & e^{-13} & e^{-10} & e^{-2} & e^{-3} & e^{-6} & e^{-11} \\
e^{-6} & e^{-4} & e^{-14} & e^{-12} & e^{-2} & e^{-1} & e^{-6} & e^{-6} & e^{-1} & 0 & e^{-10} & e^{-5} & e^{-3} & e^{-2} & e^{-3} & e^{-6} \\
e^{-24} & e^{-14} & e^{-4} & e^{-30} & e^{-14} & e^{-11} & e^{-2} & e^{-18} & e^{-13} & e^{-10} & 0 & e^{-17} & e^{-21} & e^{-18} & e^{-17} & e^{-18} \\
e^{-13} & e^{-9} & e^{-21} & e^{-5} & e^{-11} & e^{-6} & e^{-11} & e^{-1} & e^{-10} & e^{-5} & e^{-17} & 0 & e^{-10} & e^{-5} & e^{-2} & e^{-1} \\
e^{-9} & e^{-11} & e^{-29} & e^{-19} & e^{-5} & e^{-6} & e^{-17} & e^{-13} & e^{-2} & e^{-3} & e^{-21} & e^{-10} & 0 & e^{-1} & e^{-4} & e^{-9} \\
e^{-10} & e^{-10} & e^{-26} & e^{-14} & e^{-6} & e^{-5} & e^{-14} & e^{-8} & e^{-3} & e^{-2} & e^{-18} & e^{-5} & e^{-1} & 0 & e^{-1} & e^{-4} \\
e^{-13} & e^{-11} & e^{-25} & e^{-11} & e^{-9} & e^{-6} & e^{-13} & e^{-5} & e^{-6} & e^{-3} & e^{-17} & e^{-2} & e^{-4} & e^{-1} & 0 & e^{-1} \\
e^{-18} & e^{-14} & e^{-26} & e^{-10} & e^{-14} & e^{-9} & e^{-14} & e^{-4} & e^{-11} & e^{-6} & e^{-18} & e^{-1} & e^{-9} & e^{-4} & e^{-1} & 0
\end{matrix}$$

(c) To find the minimum graph cut, first compute the diagonal matrix D using the weighted matrix W. Take the sum of each row and set the diagonal to the sum.

Diagonal Matrix (D) =

$$\begin{matrix}
0.3299 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.6691 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.1537 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1426 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.1515 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.3167 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.2829 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5405 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0724 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.1410 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1537 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8940 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5808 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9556 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9511 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7752
\end{matrix}$$

After using the *svd* function in MATLAB, we can get the following eigenvalues:

Eigenvalues =

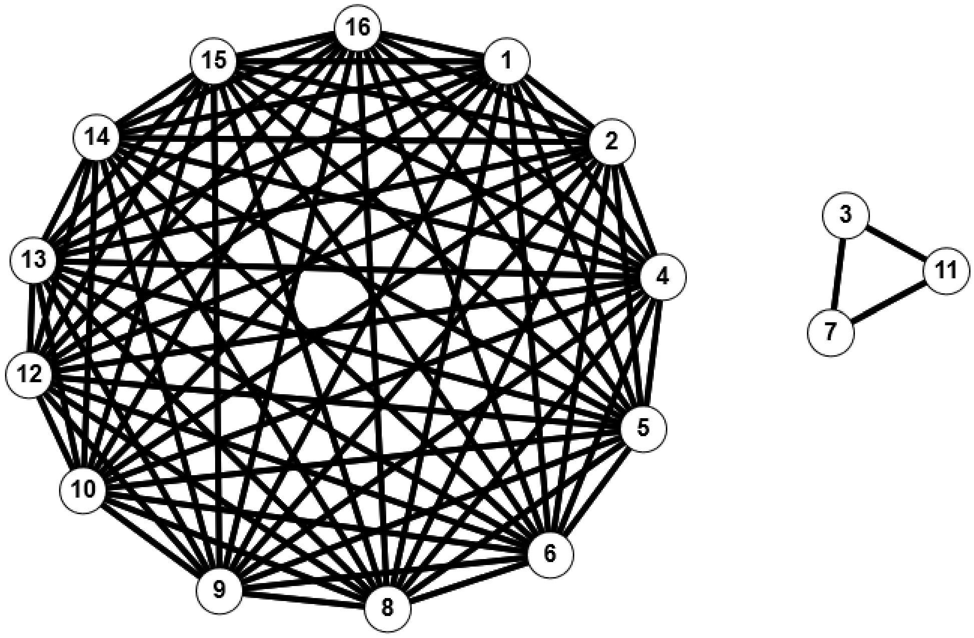
$$\begin{matrix}
1.8206 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.4857 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.4186 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.3221 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.2792 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9628 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.7264 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5648 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4393 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4140 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2882 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1720 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1555 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0566 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0049 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000
\end{matrix}$$

Chosen Eigenvalue = $\boxed{0.0049}$

Chosen Eigenvector =

$$\boxed{\begin{bmatrix} -0.1112 \\ -0.1082 \\ 0.5263 \\ -0.1372 \\ -0.1110 \\ -0.1083 \\ 0.5076 \\ -0.1327 \\ -0.1132 \\ -0.1124 \\ 0.5263 \\ -0.1305 \\ -0.1197 \\ -0.1214 \\ -0.1257 \\ -0.1288 \end{bmatrix}}$$

Cut Graph:



2 Classifying an Image using Grayscale Histograms

Performing the K-NN classification using grayscale histograms on the dataset resulted in **229** correctly labeled images out of **350** images (for a seed of 0).

$$Accuracy = (\frac{229}{350})100\%$$

$$\boxed{Accuracy = 65.428571\%}$$

Figure 1: Image Correctly Labeled as a car

Figure 2: Image Correctly Labeled as a not a car

Figure 3: Image Incorrectly Labeled as a car

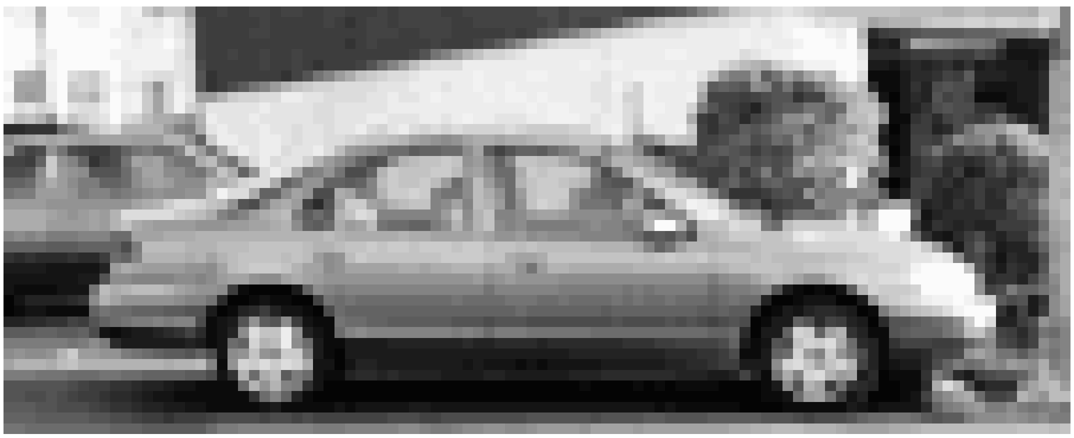


Figure 4: Image Incorrectly Labeled as a not a car

3 Classifying an Image using Gists

Performing the K-NN classification using HOGs on the dataset resulted in **324** correctly labeled images out of **350** images (for a seed of 0).

$$Accuracy = (\frac{324}{350})100\%$$

$$\boxed{Accuracy = 92.571429\%}$$

Figure 5: Image Correctly Labeled as a car

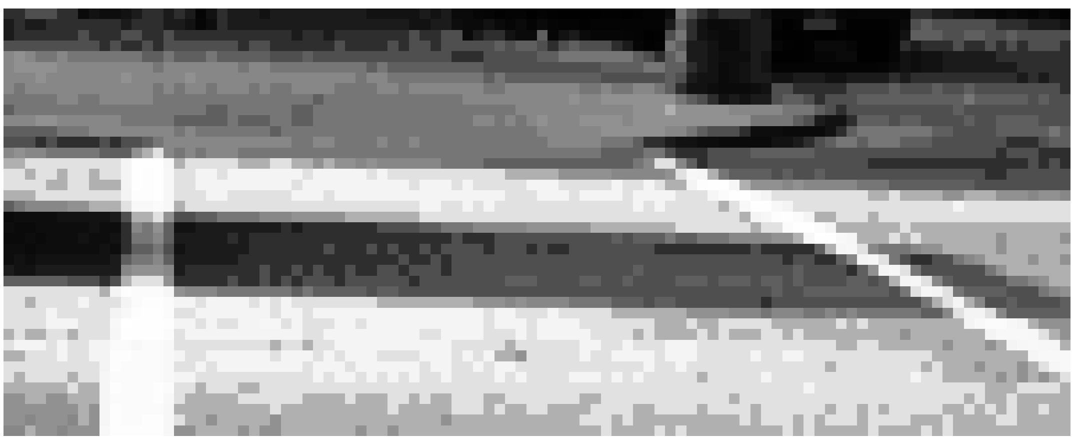


Figure 6: Image Correctly Labeled as not a car

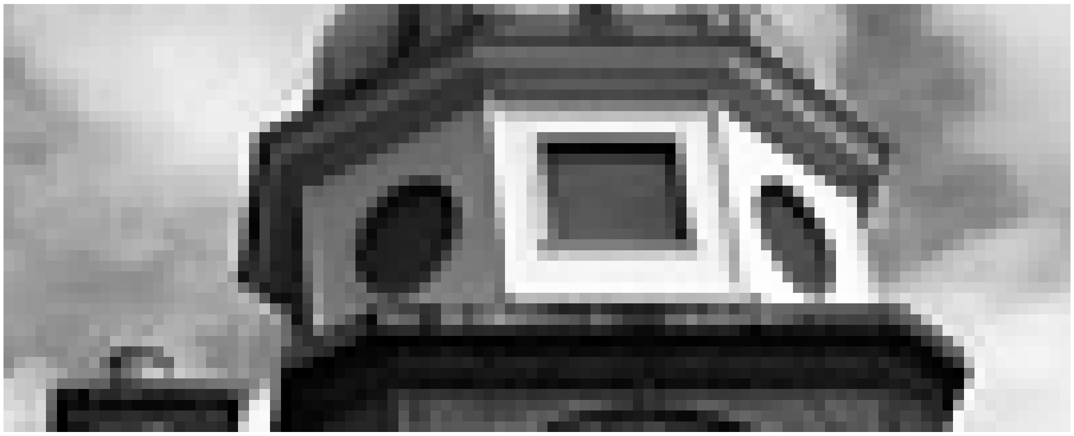


Figure 7: Image Incorrectly Labeled as a car

Figure 8: Image Incorrectly Labeled as not a car