CSCI 5521: Introduction to Machine Learning

Homework 3

Saksham Goel | goelx029 | 5138568

1. **Apply PCA to the digits data set to reduce to dimensions needed to capture 90% of the variance.**

   I applied the PCA algorithm over the whole dataset. By whole dataset I mean all the 3000 samples containing all three digits (0, 8, 9) and both types of sample (training and test). After applying PCA on that I found out that 73 dimensional space is enough to capture 90% variance.

2. **Write your own K-means algorithm and apply it to the Digits data set, after reducing the dimension using the PCA in the previous step. Use k = 6 clusters and initial centers equal to the elements # 1, 1000, 1001, 2000, 2001, 3000 in the original data set. Print out a confusion matrix showing how many 0's, 8's, 9's there are in each cluster. If there are k = 6 clusters, this matrix should be 6 × 3. If each cluster were assigned a class based on the majority label among members of the cluster, what would be error rate be?**

   The confusion matrix that I got after running K-Means with the given initialization over the dataset after reducing it to a 73 dimensional space is as follows:

Cluster # \ Digit	**0**	**8**	**9**
1	18	439	14
2	54	489	14
3	5	20	427
4	7	36	528
5	484	6	4
6	432	10	13

   To find the Error rate, we can just find the sum of the smallest two entries in each row of the table above and divide it by the total number of samples. For this part the error rate we got was:

   Error Rate: 0.0670 = 6.7%

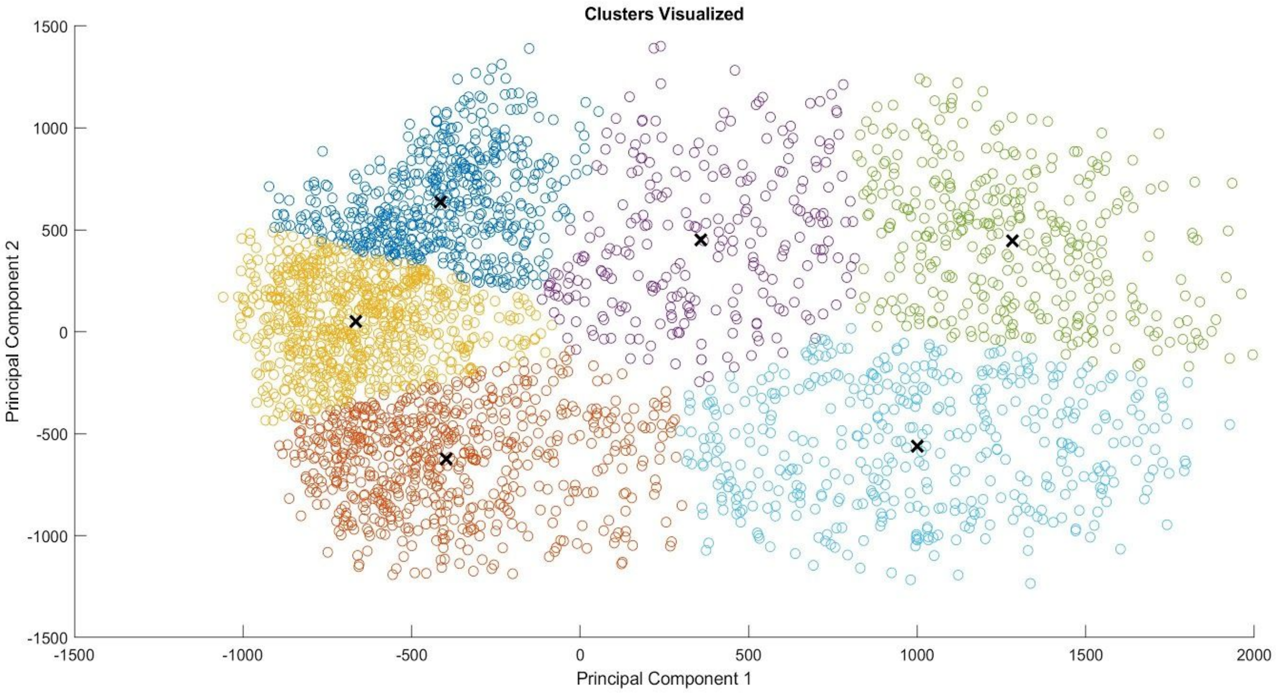
3. **Repeat the above, but start by using only 2 principal components, followed by k = 6 clusters. Initialize K-means using the same 6 data samples (projected onto the first two principal components).**

   The confusion matrix that I got after running K-Means with the given initialization over the dataset after reducing it to a 2 dimensional space is as follows:

Cluster # \ Digit	**0**	**8**	**9**
1	24	368	145
2	37	259	363
3	9	288	455
4	173	71	16
5	355	0	0
6	402	14	21

To find the Error rate, we can just find the sum of the smallest two entries in each row of the table above and divide it by the total number of samples. For this part the error rate we got was:

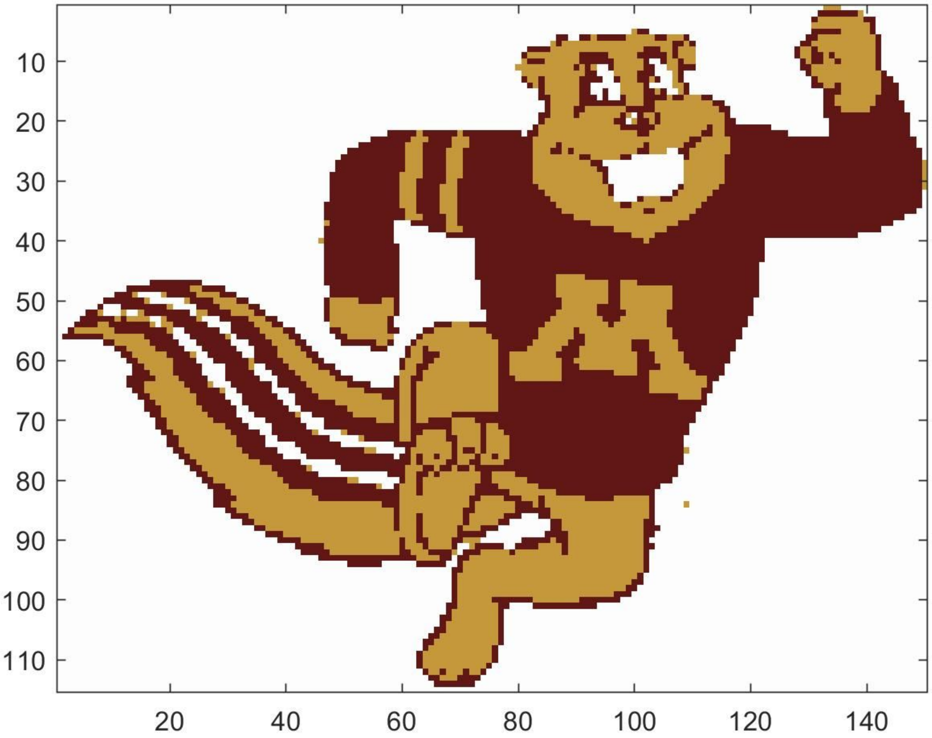
Error Rate: 0.2947 = 29.47%



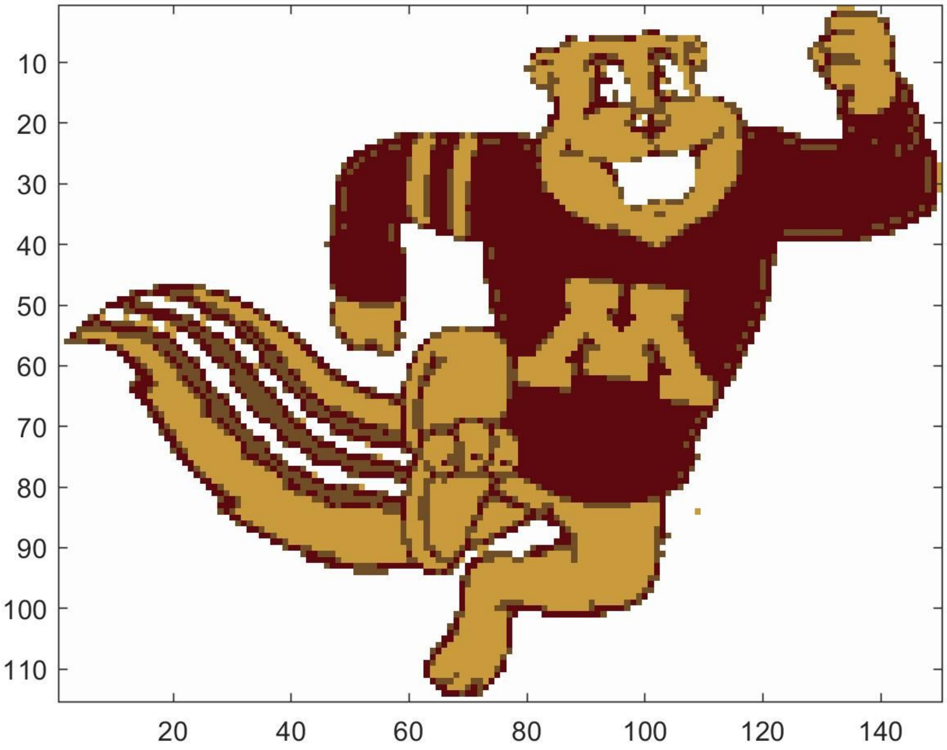
4. **Apply the k-means algorithm to the colored pixel values in image goldy.ppm and stadium.ppm. The data in this case are the RGB pixel values: points in R^3 . Try k = 3, 4, 7. Replace each pixel RGB contents with its corresponding cluster centroid, and re-form the image using the newly substituted pixel values. Redraw the resulting pictures using the modified pixel values. In Matlab, you can read in the picture using the imread function, and display it with imagesc. You can use a combination of reshape, permute to map the 3D array to a n × 3 array of pixel values (where n is the number of pixels in the image), and back again**

Goldy Images:

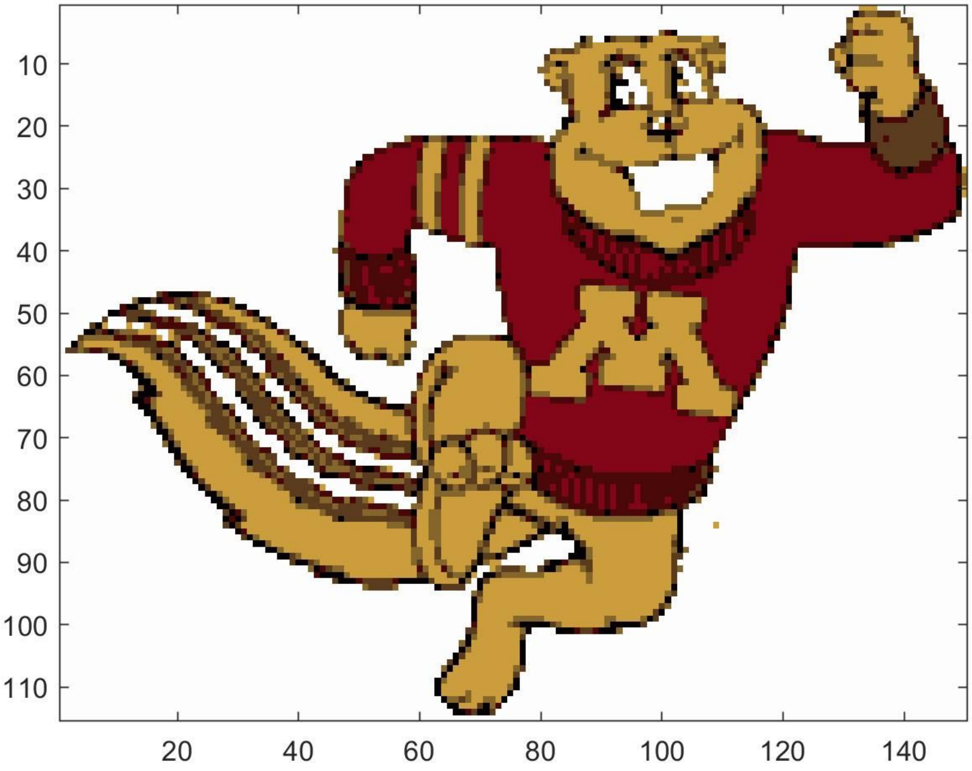
a. K = 3



b. K = 4

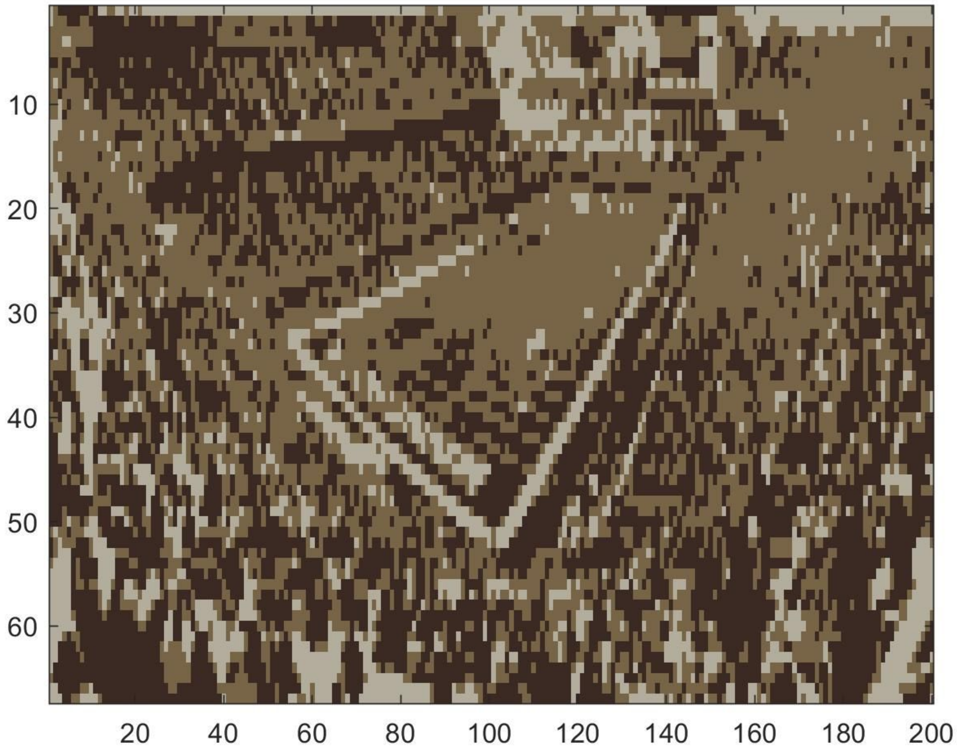

10
20
30
40
50
60
70
80
90
100
110
20
40
60
80
100
120
140

c. K = 7

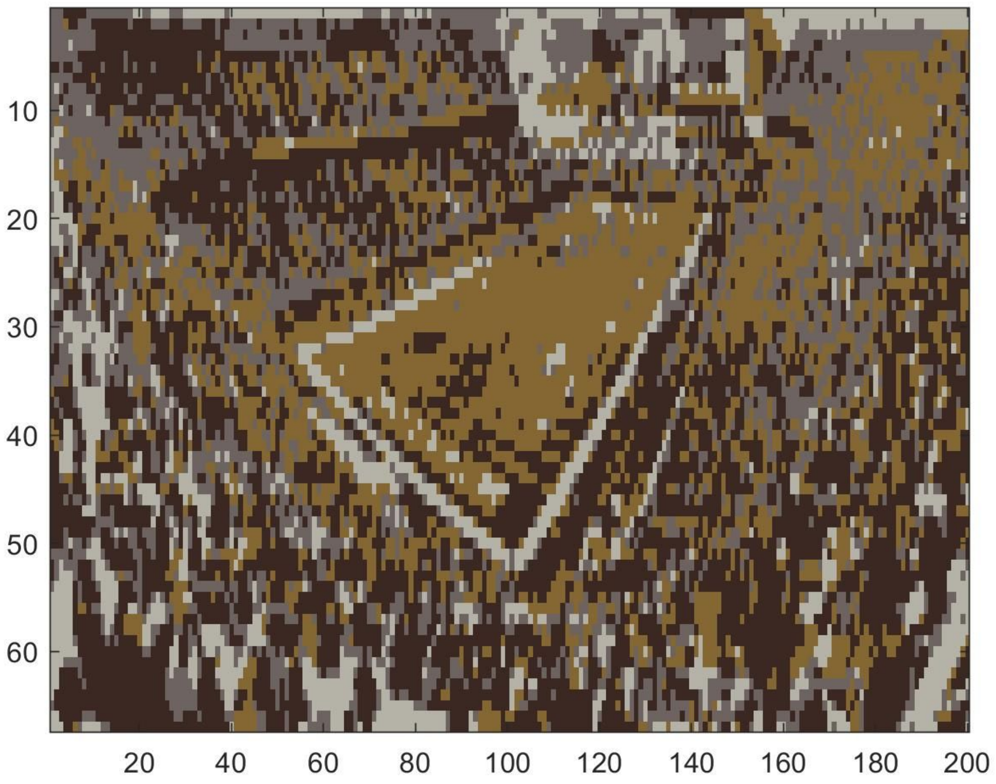

10
20
30
40
50
60
70
80
90
100
110
20
40
60
80
100
120
140

For Stadium

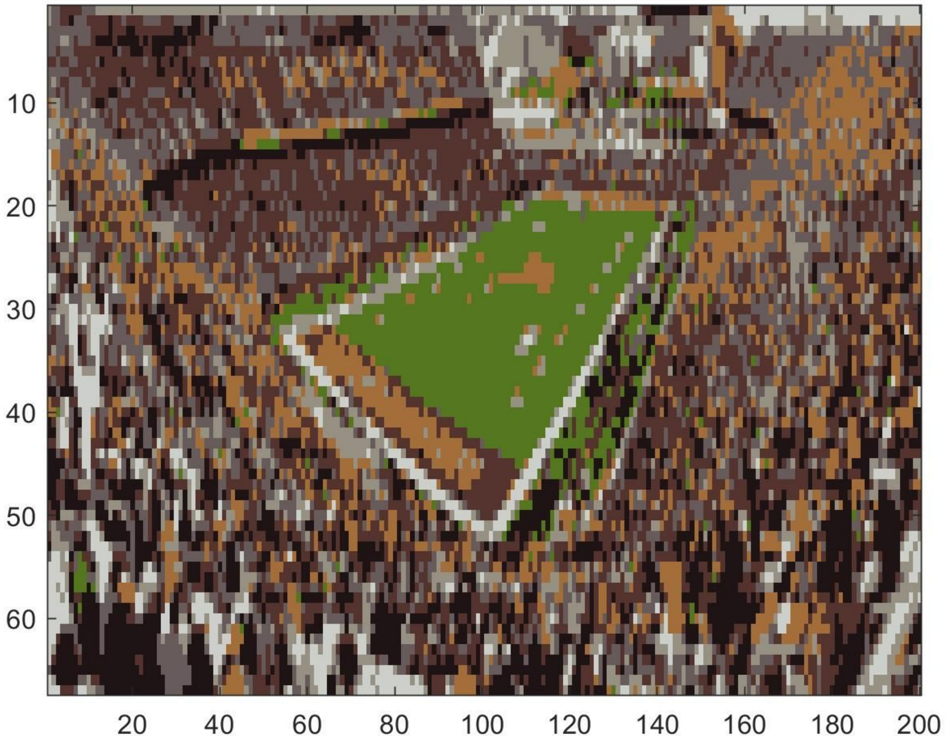
a. K = 3



b. K = 4



c. K = 7


10
20
30
40
50
60
20
40
60
80
100
120
140
160
180
200