TASK 1
s1898238

Task 1.1

Figure 1.1.1: Class 0

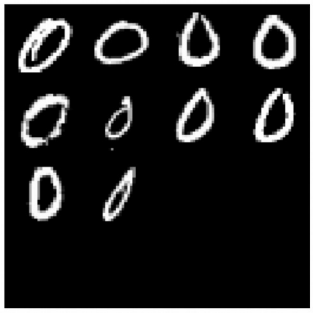


Figure 1.1.2: Class 1

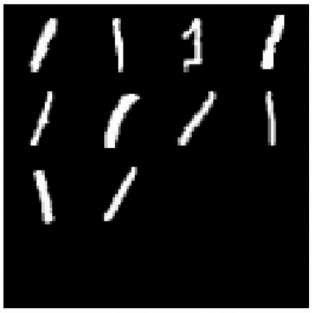


Figure 1.1.3: Class 2

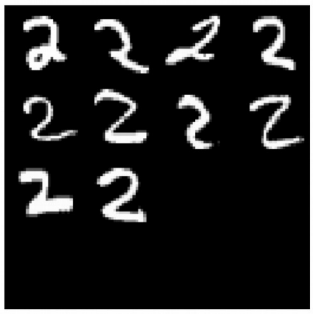


Figure 1.1.4: Class 3

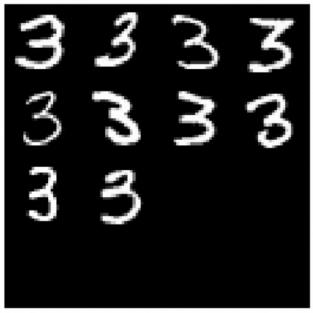


Figure 1.1.5: Class 4

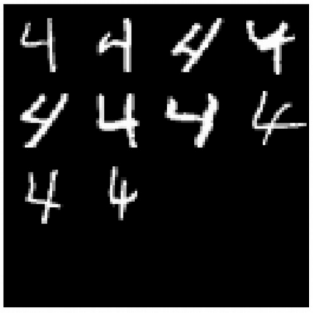


Figure 1.1.6: Class 5

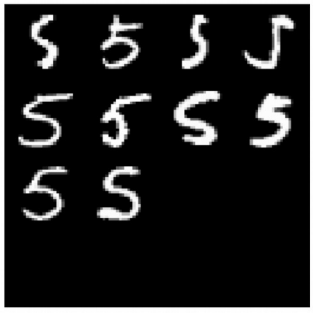


Figure 1.1.7: Class 6

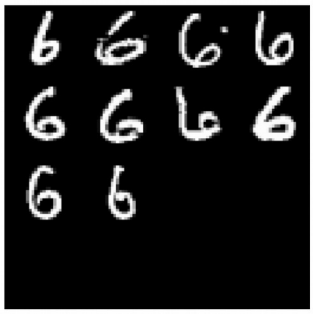


Figure 1.1.8: Class 7

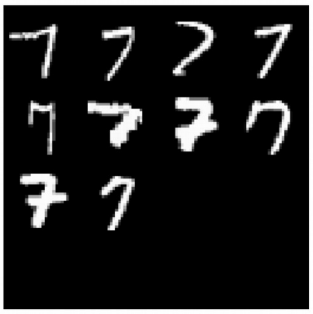
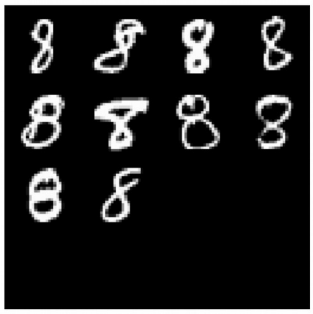
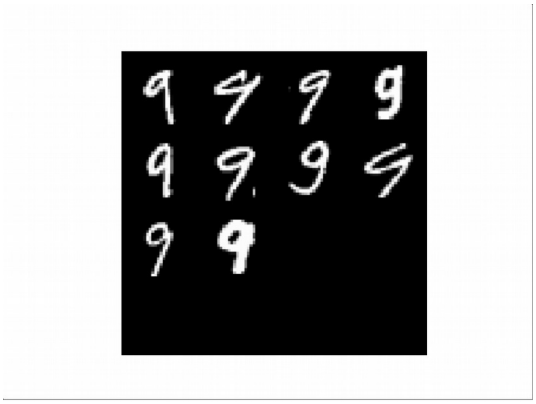


Figure 1.1.9: Class 8



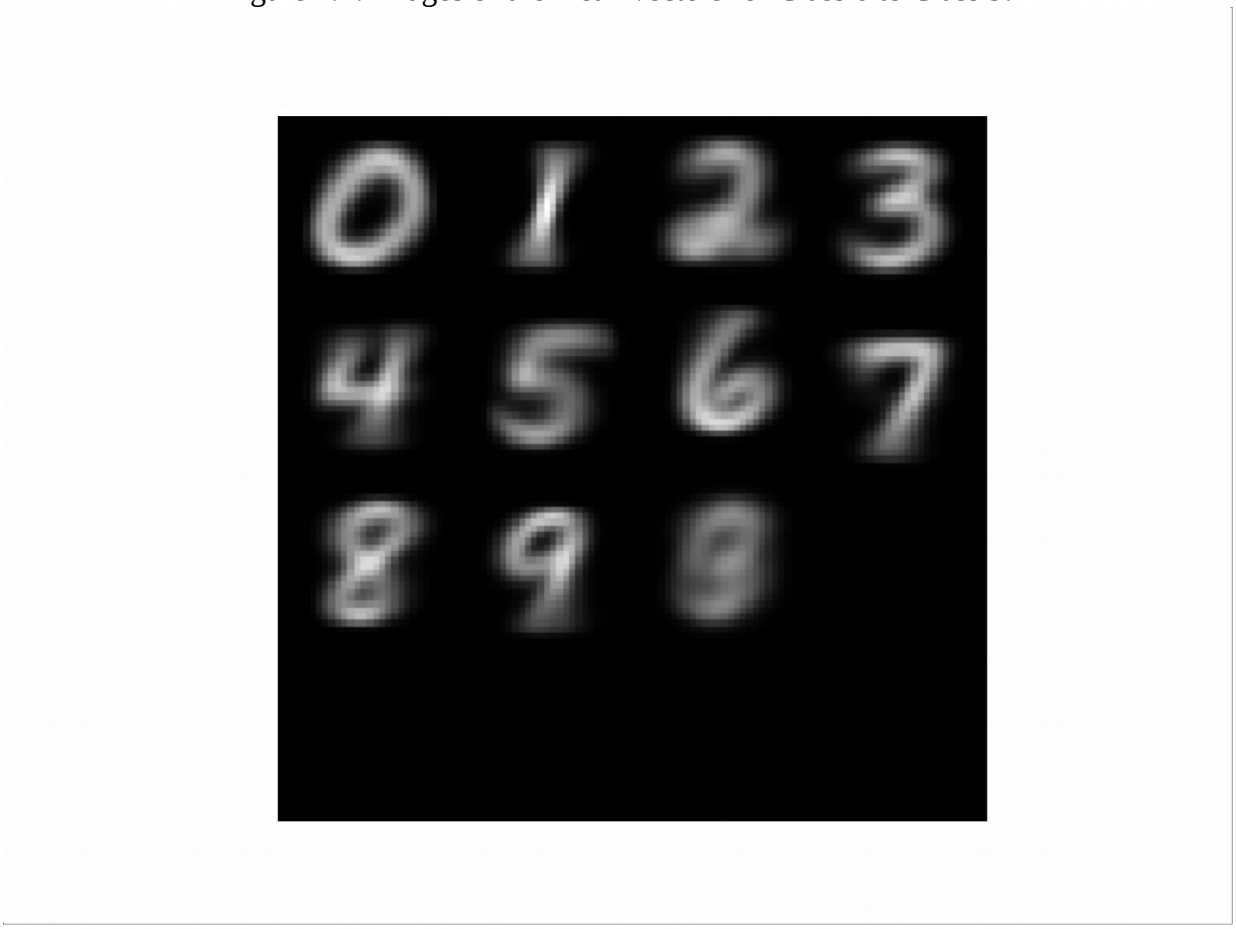
STILL TASK 1.1

Figure 1.1.10: Class 9



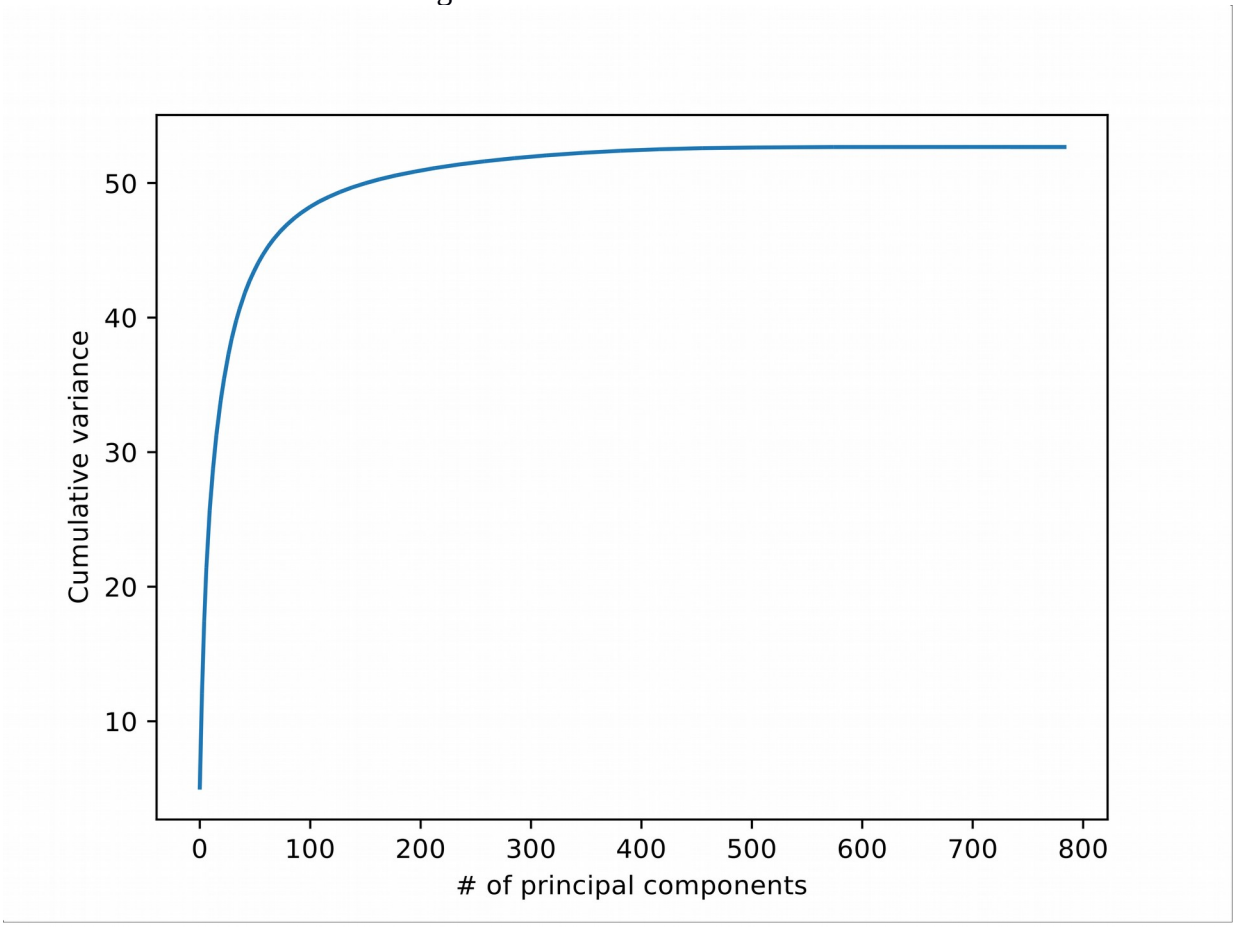
Task 1.2

Figure 1.2: Images of the mean vectors for Class 0 to Class 9.



Task 1.3

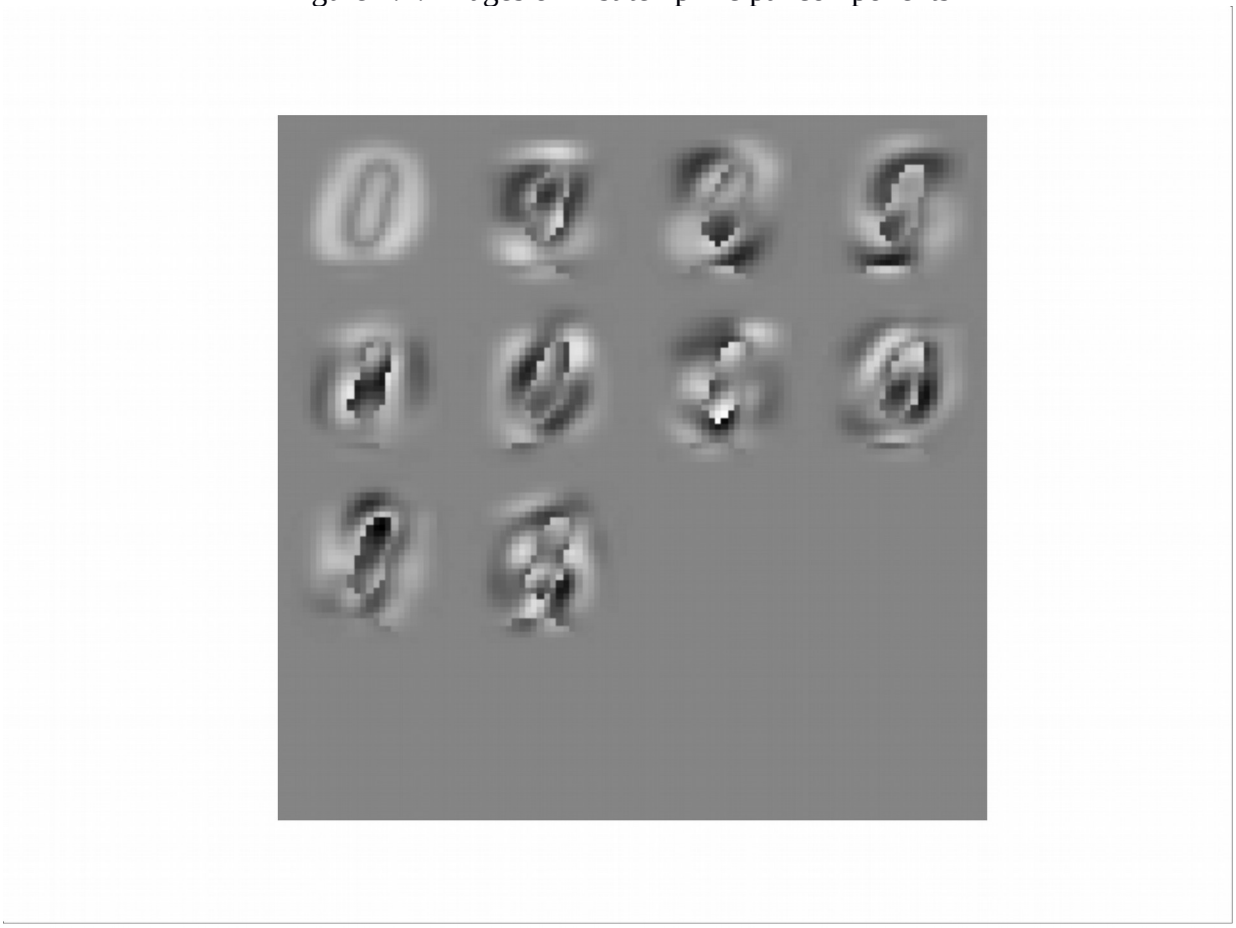
Figure 1.3: Cumulative Variance



MinDims values: 27, 44, 87, 154
Dimensions needed to cover 70% of variance: 27
Dimensions needed to cover 80% of variance: 44
Dimensions needed to cover 90% of variance: 87
Dimensions needed to cover 95% of variance: 154

Task 1.4

Figure 1.4: Images of first ten principal components



Task 1.5

Figure 1.5.1:
SSE vs iteration number for k=1: 0.63 seconds

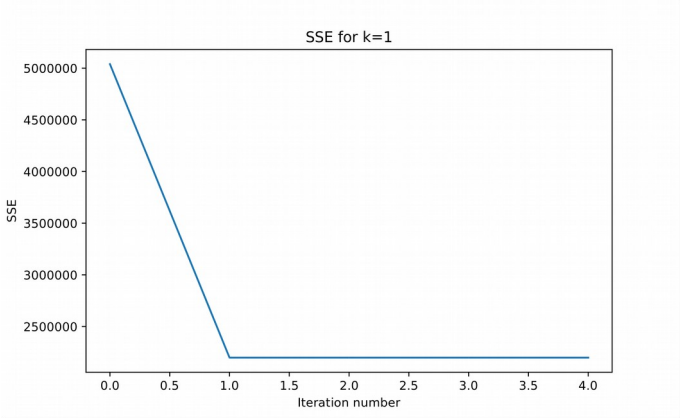


Figure 1.5.2:
SSE vs iteration number for k=2: 7.78 seconds

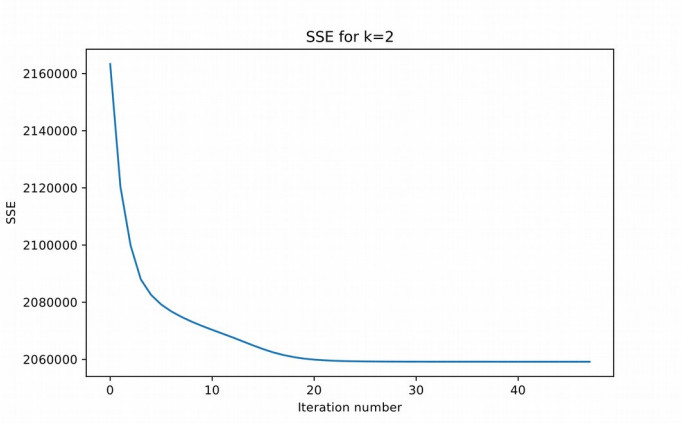


Figure 1.5.3:
SSE vs iteration number for k=3: 13.22 seconds

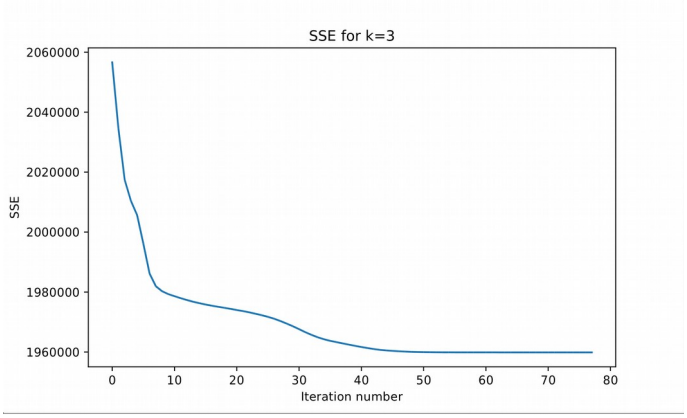


Figure 1.5.4:
SSE vs iteration number for k=4: 9.62 seconds

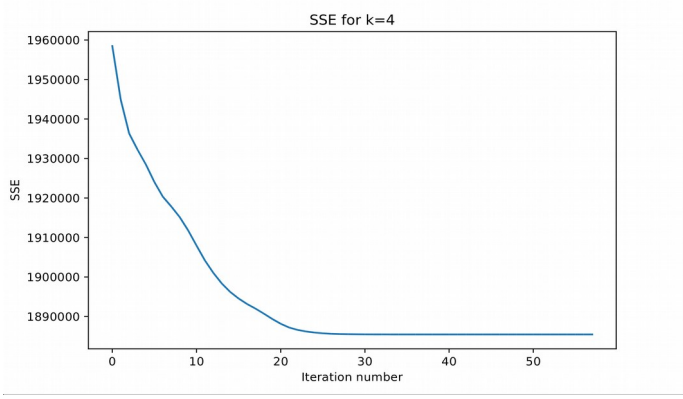


Figure 1.5.5:
SSE vs iteration number for k=5: 8.60 seconds

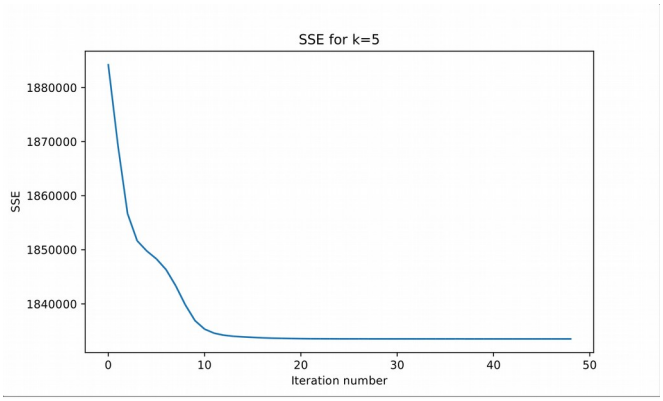
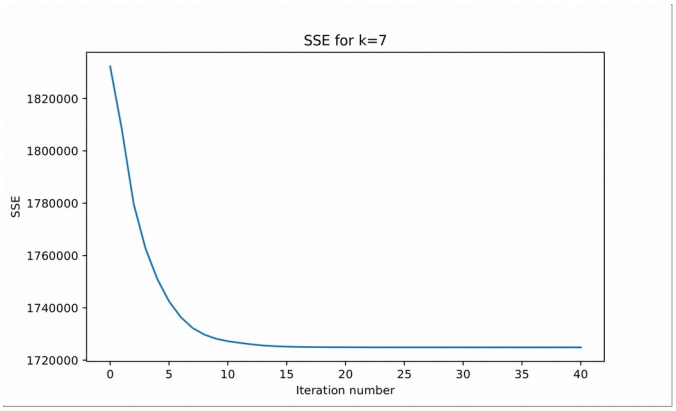


Figure 1.5.6:
SSE vs iteration number for k=7: 7.61 seconds



STILL TASK 1.5

Figure 1.5.7:
SSE vs iteration number for k=10: 15.71 seconds

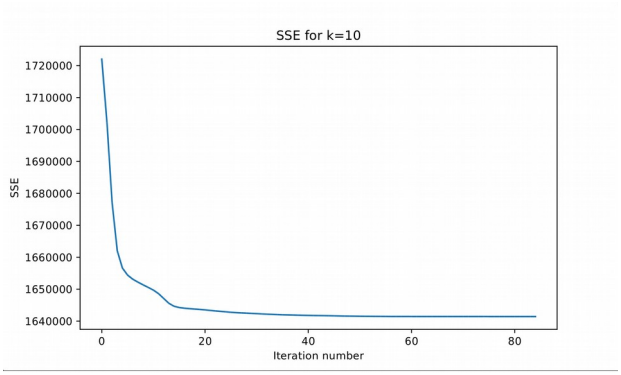


Figure 1.5.8:
SSE vs iteration number for k=15: 18.04 seconds

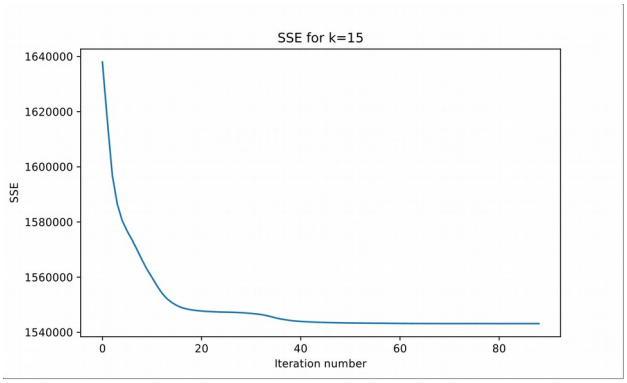
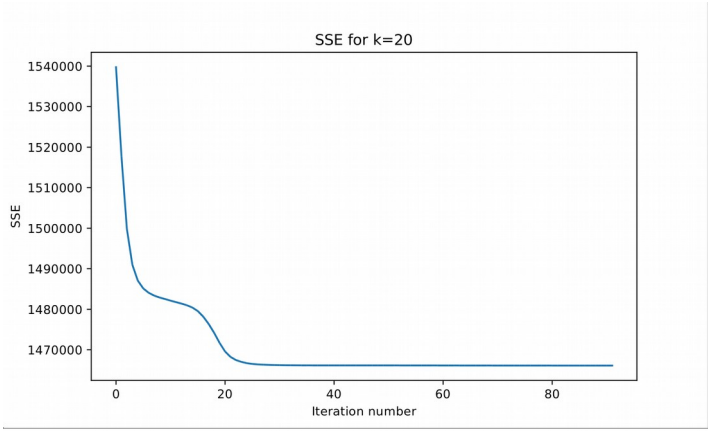


Figure 1.5.9:
SSE vs iteration number for k=20: 19.17 seconds



Task 1.6

Figure 1.6.1:
Image of cluster centres for
k = 1

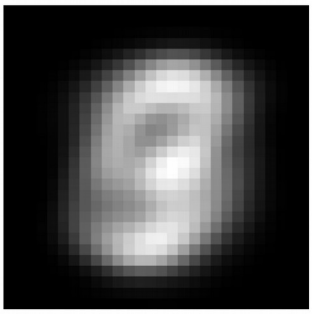


Figure 1.6.2:
Image of cluster centres for
k = 2

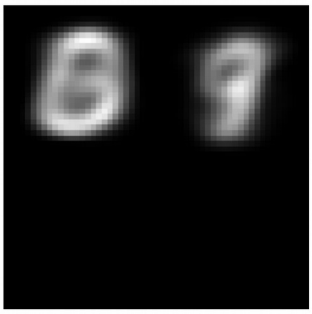


Figure 1.6.3:
Image of cluster centres for
k = 3

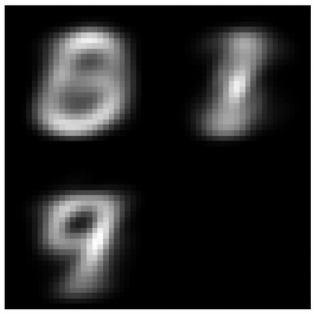


Figure 1.6.4:
Image of cluster centres for
k = 4

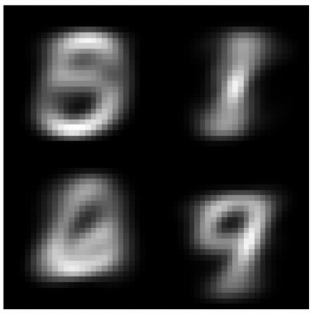


Figure 1.6.5:
Image of cluster centres for
k = 5

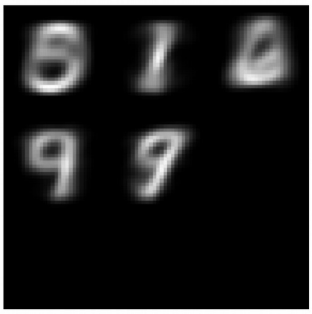


Figure 1.6.6:
Image of cluster centres for
k = 7

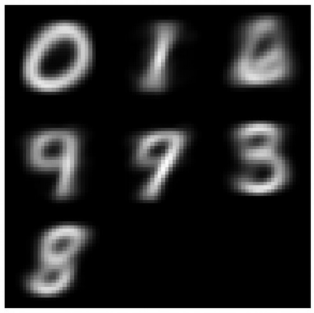


Figure 1.6.7:
Image of cluster centres for
k = 10

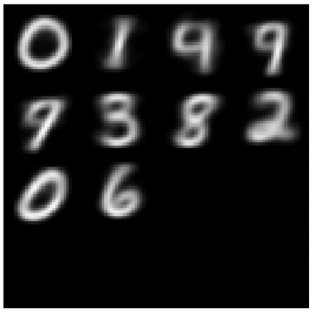


Figure 1.6.8:
Image of cluster centres for
k = 15

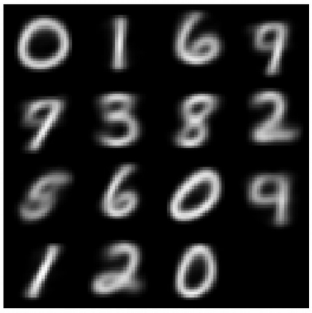
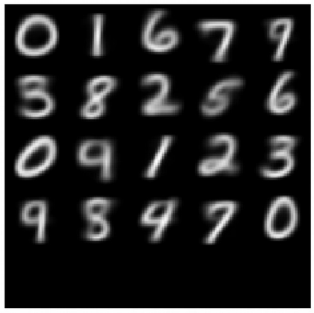


Figure 1.6.9:
Image of cluster centres for
k = 20



Task 1.7

Figure 1.7.1:
Cross-section image of cluster regions for k = 1

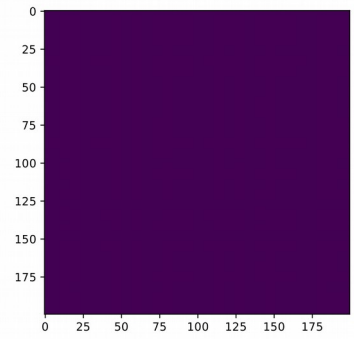


Figure 1.7.2:
Cross-section image of cluster regions for k = 2

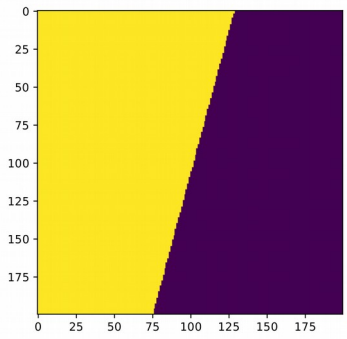


Figure 1.7.3:
Cross-section image of cluster regions for k = 3

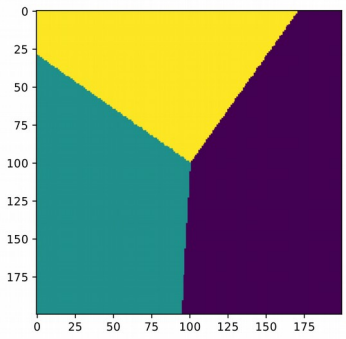


Figure 1.7.4:
Cross-section image of cluster regions for k = 5

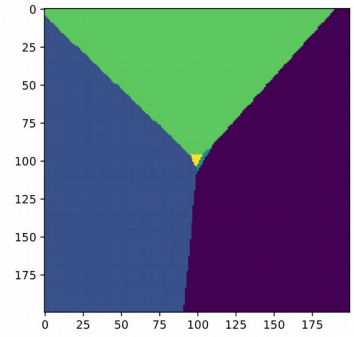
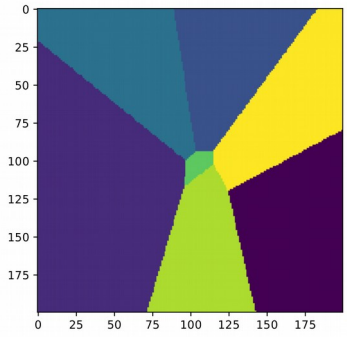


Figure 1.7.5:
Cross-section image of cluster regions for k = 10



The regions were visualised by passing the Dmap for each value of k from the task1_7 function to matplotlib's imshow function.

Task 1.8

Introduction: The purpose of these experiments was to determine the best among three methods for selecting initial cluster centers for k-means clustering. Three methods were tested with various values of k, and the SSE for each clustering experiment was recorded.

Procedure: Three methods for determining initial cluster centers were tested. The first method used the mean vector of the data set, with each dimension randomly scaled. This produced a fairly random set of points in the general vicinity of the data set. The second method used the first k samples from the data set as the initial cluster centers. The third method used the mean vector, adjusted by 1 in 1 dimension for each class to produce the necessary number of centers. This essentially placed all the centers extremely close together in the center of the data set. Each method was tested with 500 iterations maximum for k values of 1, 2, 3, 4, 5, 7, 10, 15, and 20, and the SSE for each experiment was recorded.
Data:

Figure 1.8.1: SSE for each value of k for each selection method

	Mean Randomly Scaled	First k Vectors	Start Near Mean
K = 1	2.199E6	2.199E6	1.661E6
K =2	2.059E6	2.059E6	1.626E6
K = 3	1.960E6	1.959E6	1.599E6
K = 4	1.885E6	1.885E6	1.573E6
K = 5	1.833E6	1.833E6	1.555E6
K = 7	1.976E6	1.724E6	1.512E6
K = 10	1.667E6	1.641E6	1.484E6
K = 15	1.622E6	1.543E6	1.474E6
K = 20	2.458E6	1.466E6	1.475E6

Results and Discussion: The third method was the best, producing the smallest SSE for every value of k except k = 20. Notably, the first method produced many empty clusters. The success of the third method makes sense, especially if the data set is not sorted in any particular manner, for placing the centers near the mean allows them to be pulled in the proper direction by the data. However, if the general vicinities of the clusters were known beforehand, one could place the initial centers in even better points.

Conclusion: Three methods for determining the initial cluster centers to be used in k-means clustering were examined. The method of slightly altering the mean vector to place all centers extremely close to the mean proved the most successful, producing the lowest SSE in almost all cases. There are surely even better methods of selection which could be determined by further experimentation.