Erasmus University Rotterdam
MSc. Business Analytics and Quantitative Marketing

Recreating Image by Pixels Clustering with K-means Clustering, Agglomerative Hierarchical Clustering, and Spectral Clustering

Course: Machine Learning

Coordinator: Professor Ilker Birbil

Student: Linh Dinh

1 Introduction

Cluster analysis is widely applied in business, science and other fields. For example, we can use it to classify the customers' group in marketing analytics, or it can be used to get the trajectory of human evolution based on clustering human's gene sequence in genetics. Hence, clustering in the very first step to analyze cluster is crucial. This report provides three methods to group the pixels into clusters. The methods are K-Means clustering, Agglomerative Hierarchical clustering and Spectral clustering and the implementation is based on the image pixel by clustering the pixels' RGB value.

This report contains three main sections. First, it provides a definition of the the objective problem. Then, it presents the intuition and detailed explanation of each method. In the last section, the results of each clustering method and the comparison of the results within and between each method will be presented.

2 Problem definition

The goal of this report is to solve a clustering problem for an image as shown in Figure 1. The clustering is based on the RGB value (between 0 and 255) of each pixel. All of the implementation of the clustering are under values K=2, 4, 8, 16. It means that we will try to cluster the image's pixels into 2, 4, 8 and 16 clusters by each method.

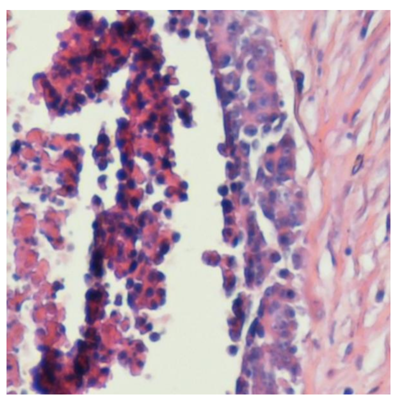


Figure 1: Image that used for the clustering problem

3 Methodology

3.1 K-Means Clustering

k-means clustering algorithm is one of the simplest unsupervised machine learning algorithms. To perform k-means algorithm, it is necessary to define the number of clusters K before the iteration. Then, it randomly assign the cluster number from 1 to K each observation. Next, compute the centroid of each cluster and assign observations again to the closest centroid. Distance between centroid and observation is based on Euclidean distance. Repeat this iteration until the clusters do not change.

k-means algorithm works when when the variables are large as it is computationally faster than other clustering algorithms in general. However, the result is locally optimal and possibly not match global optimum since the results heavily depend on the initial starting points. Furthermore, unlike other clustering algorithms, it is necessary to specify the number K beforehand. Although elbow method can be used to determine K if the prior knowledge of dataset is limited, it can be ambiguous to determine. Last drawback of k-means algorithm is that it only works for linear boundaries. k-means algorithm assumes that the boundaries are linear therefore, it fails when the boundaries are non-linear.

Next, we will explain the objective function J for the k-means method. As we presented before, the core idea of k-means method is to minimise the distance of each observation to their cluster centroid, we can express it by the objective function $J = \sum_{k=1}^{K} \sum_{i=1}^{n} \|x_i - \mu_k\|^2$, which x_i represent each observation i and μ_k is the centroid of cluster k.

To prove that the algorithm decreases the objective at each iteration, we need to define the 'decreases' first. The decreases means the objective is decreased or remain the same, or we can say not increased. Then, let's look at the objective function $J = \sum_{k=1}^{K} \sum_{i=1}^{n} \|x_i - \mu_k\|^2 z_{ik}$ and we simplified it. For each cluster k, the abbreviated objective function is $J_k = \sum_{\vec{x} \in \omega k} \|\vec{x} - \vec{\mu}_k\|^2$, ωk means the set of the samples in cluser k. As we know, the intuition of the k-means algorithm is assign all the samples to the closest centroid, so the sum of the distances between (the objective function) of samples and their centroid are minimized. That is, in every iteration the reassignment of each sample will reduce $\sum_{\vec{x} \in \omega k} \|\vec{x} - \vec{\mu}_k\|^2$ and it will contributes the reduction of the objective J. Also, when we try find a new iteration in every iteration, we minimized the objective by replace the old centroid to the new. Hence under each iteration, we can only reduce the objective function rather than increase.

In our implementation, we choose the initial centroids based on method k-means++ method instead of random. Random method selects initial centroids randomly by choosing k observations. However, k-means++ method selects initial centroids based on specific probabilities. This initialization is simple and computationally fast. According to **?**, The steps of k-means++ method is as below.

1. Choose the first initial centroid randomly
2. For x_i in our sample set, find the minimum distance between x_i and the chosen centroid(s), which are named r, $(x_i) = arg\ minf\ \|x_i - r\|^2, r \in 1, 2, ..., k$.
3. Next, we taken the x_i with highest (x_i) value as the next centroid.
4. Repeat step 2 and 3 until all K centroids are chosen.
5. Then we use all the chosen centroids as our initial centroids.

The stopping criterion is both set the maximum iterations the minimize the objective. As we use the python function *'sklearn.cluster.KMeans'*, the iterations will stop when we find the minimum objective or reach the maximum iteration times which is 300 times by default. After we get the results, we use elbow method to select the best K.

3.2 Agglomerative Hierarchical Clustering

Hierarchical clustering method is based on a dendrogram which is tree-based representation of data points. Two types of hierarchical clustering are agglomerative clustering algorithm and divisive clustering algorithm. Agglomerative clustering starts with each observation as its own cluster and merge with other clusters by going up the hierarchy. Whereas, divisive clustering considers all observations as a single cluster and divide into two and so on in each iteration until all observations are assigned to one cluster. In this report, agglomerative hierarchical clustering method will be discussed and applied to the cell image data.

Iteration of agglomerative hierarchical clustering starts at the bottom of the dendrogram. Each data point is considered as an individual cluster and merge with the most similar cluster. By repeating the iteration until the dendrogram is complete, the observation is assigned to one single cluster. Dissimilarity of each point is measured by Euclidean distance and there are various ways to define dissimilarity so called linkage methods. It is important however to decide which linkage method to use since the distance between clusters could be formulated and interpreted differently depends on each method.

In this report, we focus on complete linkage method and single linkage method as they are the most preferred ones compare to other linkage methods. Complete linkage method computes all pairwise dissimilarities between observations in each cluster and selects the two most distant objects. This method is preferred in studies as it yields rather balanced dendrograms than other methods. Single method whereas takes the minimal intercluster dissimilarity. It does not impose any constraints on the shape of clusters and hence flexible with irregular shaped clusters. Reason we focus on complete linkage and single linkage in this report is that since one is based on the similarity of their most similar members and another one is based on the similarity of their most dissimilar members, we can expect that clusters from each method would be very different from each other and get different results that can be compared.

Since agglomerative hierarchical clustering does not require to specify the number of clusters in the beginning, it is attractive to use when the given information of the dataset is vague or not sufficient. However, the running time of this algorithm is $O(n^2)$ and it becomes impractical when the dataset is large. Therefore, in the later part of report describes alternative way of applying agglomerative hierarchical clustering in large dataset.

3.3 Spectral Clustering

Spectral clustering is a method that uses the eigenvalues of a similarity graph to partition the nodes (or data points) into groups in which data points in the same group are similar and in the different groups are dissimilar. In the similarity graph $G = (V, E)$, each point $v_i \in V$ is a vertex and each edges represents the similarity between two vertices, the weight of an edge between v_i and v_j is defined as w_{ij} which comes from a similarity matrix W.
Given a dataset X = $x_1, \dots, x_n$ in R^p that we want to cluster into k subsets, the algorithm of the method is as following:

1. Form the similarity matrix $W \in R^{n\times n}$ with the below formula:

$$w_{ij} = \begin{cases} exp(\frac{-||x_i - x_j||^2}{c}) & ,i \neq j \\ 0 & ,i = j \end{cases}$$

2. Define D to be the diagonal matrix in which $d_{ii} = \sum_{j=1}^{n} w_{ij} \forall i$.

3. Compute the Graph Laplacian matrix : $L = D - W$, more oftenly, we compute $L^{norm} = I - D^{-1/2} W D^{-1/2}$, which is the normalized form of L.

4. Find $z_1, \dots, z_m$, the m eigenvectors of L corresponding to the m smallest eigenvalues, and form the matrix $Z = [z_1, \dots, z_m] \in R^{n\times m}$ by stacking the eigenvectors in columns.

5. Treating each row in Z as a data point and k-means clustering to group them into K clusters.

6. Assign the original point x_i to cluster j if and only if row i of the matrix Z was assigned to cluster j.

There are many methods to compute the similarity matrix, of which the most popular is Gaussian kernel with the Euclidean distance, $w_{ij} = exp(\frac{-||x_i - x_j||^2}{2\sigma^2}) \forall i \neq j$. Hence, given the affinity (similarity) W is the adjacency matrix of a graph, the simplest way to construct a partition is to solve the min-cut problem: Choose the partition $Z_i, \dots, Z_k$ that minimises $cut(Z_i, \dots, Z_k) = \frac{1}{2}\sum_{i=1}^{k} W(Z_i, \bar{Z_i})$ where $W(A, B) = \sum_{u\in A, v\in B} W(i, j)$. The normalized cut has been proposed by **?**, given The size of a cluster Z_i is measured by the weights of its edges $vol(Z_i)$, $Ncut(Z_i, \dots, Z_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{W(Z_i, \bar{Z_i})}{vol(Z_i)} = \frac{1}{2}\sum_{i=1}^{k} \frac{cut(Z_i, \bar{Z_i})}{vol(Z_i)}$. Here, a cut value between two clusters A and B is defined as: $cut(A, B) = \frac{1}{2}\sum_{u\in A, v\in B} w_{uv}$.
The spectral clustering method has many advantages. First, it does not make strong assumptions on the statistics of the clusters. Second, unlike k-means clustering, this method can find cluster with non-convex boundaries. However, this algorithm is sensitive to the choice of parameters and computationally expensive when dataset is large. Image segmentation based on spectral clustering improves the quality of image segmentation. Spectral method reduces dimensions by using the eigenvalues of the similarity matrix to group data points. The advantage of spectral clustering on image segmentation is generating good results and also reducing the computation layout. However the disadvantage of spectral clustering is when the image resolution is high, the spectral clustering method can lead to overlarge adjacency matrix and thus is impossible to compute.

3.4 Clustering Speed Improvement Methods

Hierarchical clustering algorithm is more preferred than k-means algorithm when the information of the data is not sufficient as it does not require to prespecify the number of clusters. In a same manner, spectral clustering algorithm is well-known that it produces high-quality clustering on small dataset. However, these two algorithms are impractical when the dataset is large since they suffer from the computational complexity. Computational complexity for hierarchical algorithm and spectral algorithm are $O(n^2)$ and $O(n^3)$ respectively. Therefore, the report describes two alternative ways of applying agglomerative hierarchical clustering and spectral clustering in large dataset. The first method is as follows.

1. Use k-means clustering P times to get $(P * K)$ centroids.

2. Apply hierarchical clustering (or spectral clustering) on the centroids to get cluster of all $(P * K)$ centroids.

3. Apply k-means clustering with these centroids as the initial centroids to get the final clusters.

This algorithm is a hybrid method of k-means clustering and another method (either Hierarchical clustering or Spectral clustering) to make use of the best features of the original methods and avoid their limitations. As K-means clustering is sensitive to the centroid initialization, the clusters it provides is not robust. By running K-means mulitple times, we can have more centroids from different seeds. Then either Hierarchical clustering or Spectral clustering is applied to get the cluster centers of these centroids and use them as initial centroids for the final K-means clustering, the final clusters will be more robust, especially for the ones combined with Hierarchical clustering since Hierarchical clustering is insensitive to initialization. Moreover, the running time is faster with this algorithm because the two slow algorithms are only implemented on a small set of centroids compared to the large dataset of pixels.

We also propose the another way to speed up the algorithm on the large dataset which is originated from **?**. The method of modified agglomerative hierarchical clustering (Spectral clustering) algorithm is called Multi-Stage method.

1. Split the dataset $X = \{X_1, X_2, \ldots, X_n\}$ that contains n objects into P subsets.
2. For the first stage, apply Hierarchical clustering algorithm (or spectral clustering) on each of P subsets.
3. Get K-cluster centroids of all P subsets.
4. For the second stage, apply agglomerative hierarchical clustering (or spectral clustering) algorithm on all the centroids to get the some K clusters.
5. The cluster labels of the second stage clustering are mapped back to the original dataset X.

As it splits the data into subset, the algorithm runs faster than when applying it to the entire data sets. Once we obtain with the centroids of each subset, we run agglomerative hierarchical clustering again on the centroids of whole subset and to get the clusters on the centroids. The clusters of the second stage are the final clusters and will be mapped back to the corresponding clusters of each subsets in the original dataset X. Unlike the first method, which is a hybrid version of two different clustering algorithms, the second method only uses one clustering method but two times and on small subsets of datapoints. Hence, the second method is expected to run much slower than the first one, especially the one using the spectral clustering which usually takes long time on its own.

4 Results

4.1 K-Means Clustering

In this section, we presents the result of the clustering based on k-means method. The summary of clustering results can be found in Table 1) in the appendix. It shows that the objective errors J, the size (the biggest cluster size with red color and the smallest with green color) of each cluster, the centroids and the number of iterations of the algorithm by K equals to 2, 4, 8 and 16. According to this table, the objective error J, which is the total within cluster sum of squares, decrease while the number of iterations increases when K increases. This implies that there is a trade-off between the objective error and the converging time of the algorithm.

Moreover, we compare the objective values and running time between two initialization methods of centroids in Table 2. From the table, it is observable that in most cases, the k-means++ method has lower objective error and lower running time than the random initialization, which means that k-means++ method performs better than random method. Therefore, for method 1 of the clustering speed improvement methods, we will apply the k-means++ initialization to speed up the algorithm and to get more concentrated clusters, especially for a large value of K.

Last, the total within cluster sum of squares with different values of K is presented in Figure 2 of the appendix. According to the elbow rule, we can take the best K as 4. Also, you can find the regenerated images (Figure 3) for after clustering with different values of K there. That is, for each regenerated images, the pixels in the same cluster have the same the RGB values (the centroid vector of each cluster). As expectation, the higher K is, the closer the recreated image is to the original image.

4.2 Agglomerative Hierarchical Clustering

Table 3 and Table 4 in the appendix presents the results of agglomerative hierarchical clustering Method 1 for complete and single linkage, respectively. Both tables show that as K increases, the of objective error decreases. This is an expected result as the final clustering is k-means which optimizes the total within cluster sum of squares. Moreover, we can observe that the objective error between complete and single linkage don't differ much in the final k-means clusters which implies that it does not matter what linkage measure is used in agglomerative hierarchical clustering for Method 1. The reason for this is that method 1 is the hybrid version of the k-means clustering and agglomerative hierarchical clustering, where the former algorithm clusters on the whole dataset and the latter algorithm clusters on the cluster centroids created by the former algorithm. Thus, the latter algorithm has less influence on the final clustering result than the former algorithm as it clusters on a small sets of concentrated data while the former is applied twice at the beginning and at the end.

In order to determine the value of K, we also plot Figure 4. From this figure, we can conclude that for both linkages, the best K is 4 according to the elbow rule. This result is similar to k-means clustering as k-means is used twice in the this method. Furthermore, the regenerated images of this method for two linkages are available in Figure 5 and 6. We can see that for K = 4, the recreated images of two linkages are quite close to the original picture

For Method 2, we don't report the within cluster error since it is not the objective function of agglomerative hierarchical clustering. Instead, the clustering results of the agglomerative hierarchical clustering with complete and single linkage can be seen via the regenerated pictures in Figure 7 and 8. In Figure 7, we can see that the higher K is, the closer the image represents the original version. While in Figure 8, even for high values of K, the images are no where near the original version and they cannot be distinguished between small K and high K (except for different colours). Hence, in Method 2, the complete linkage performs much better than the single linkage. As a result, we can conclude that complete linkage is a better measure for agglomerative hierarchical clustering than single linkage when the speed-improvement method only considers the agglomerative hierarchical clustering and no other clustering methods.

4.3 Spectral Clustering

The results of spectral clustering of Method 1 can be seen in Table 5. In general, the higher the K is, the lower the with cluster error is. This corresponds with the result in k-means clustering and Hierarchical Clustering (Method 1) as the major algorithm for Method 1 is k-means which optimize on this objective error. The value of K can be decided by Figure 9, which also agrees with the above result that K = 4 is a good choice of number of clusters. This result is as expectation as k-means clustering is the main algorithm of this method. The regenerated images of this method are available in Figure 10 where we can see that for K = 4, the recreated images of two linkages are clear and close enough to the original picture.

The results of spectral clustering in Method 2 cannot be reported since it takes too long to run. Therefore, it can be deduced that spectral cannot be speed improved by Method 2.

5 Conclusion

In this report, grouping the pixels of cell image into clusters using three clustering methods which are k-means clustering, agglomarerative hierarchical clustering and spectral clustering are shown. In order to analyze the image data, we take the each pixels of an RGB image and this leads to the huge size of the dataset. As it is described in the previous part of the report, k-means algorithm works well without any improvement on our data because it runs faster than other methods. By using k-means++ method, we select initial centroids of $K = 2, 4, 8, 16$ and compare WCSS of each case. We decide to take the best K as 4 from visually checking the elbow graph. Then, we implement agglomerative hierarchical clustering and spectral clustering on our data. As these two algorithms suffer from computational complexity, we need alternative ways to apply these algorithms. By engaging two alternative ways in these two algorithms, it is possible to cluster the image data except the second alternative method on spectral clustering algorithm. As hierarchical clustering highly depend on which dissimilarity method to use, complete linkage method and single linkage method are used in agglomerative hierarchical clustering. Unlike k-means algorithm, it is not necessary to choose the number of initial centroids beforehand. Re-gererated pictures based on different linkage methods, applying the alternative ways are shown. Under the alternative method 2 using single linkage method however, did not performed well and it can be seen from the regenerated pictures. This is because alternative method 2 basically perform agglomerative hierarchical clustering twice and since single linkage method takes the mininal intercluster dissimilarity, the centroids do not represent the data well. Lastly, spectral clustering is applied and although it is possible to regenerate pictures based on alternative method 1, it is not possible to regenerate pictures based on alternative method 2 as the sorting the index is not efficient enough. Therefore, further study is required in this particular case.

.1 K-Means

K	Objective Error	Label	Size	Centroid			Iterations
2	844,611,707.20	0	217918	[216.57875572,	198.2937713,	201.96570116]	5
		1	142082	[129.44291491,	85.33751568,	131.42483761]	
4	304,068,436.97	0	57123	[90.61653898,	54.71115513,	107.36999702]	6
		1	96391	[203.95729464,	166.0816792,	184.58455558]	
		2	84219	[155.2601351,	105.7119437,	147.47172858]	
		3	122267	[226.38986642,	223.3486548,	215.46374144]	
8	150,632,490.90	0	23473	[170.53757718,	86.7514158,	122.3403449]	52
		1	97052	[228.24691651,	229.0526332,	217.82408886]	
		2	20630	[65.95935865,	37.70093005,	89.00329394]	
		3	52184	[203.82021309,	161.6722175,	180.96838111]	
		4	33411	[128.86315317,	100.6210511,	151.27725707]	
		5	33677	[103.84348858,	63.24556511,	116.84387422]	
		6	55942	[216.39517038,	192.1210446,	200.84436097]	
		7	43631	[172.92147245,	129.2199276,	164.61751627]	
16	79,464,617.65	0	32856	[[220.83241533,	207.588357,	209.59800371]	154
		1	19638	[165.53375452,	135.3294951,	171.25989708]	
		2	14744	[94.99593,	73.82254782,	134.78666395]	
		3	12757	[134.8478346,	64.88143159,	107.45626126]	
		4	35548	[214.98853707,	185.6922148,	196.96459978]	
		5	84301	[228.98647703,	231.2268182,	218.56789362]	
		6	18377	[196.92654016,	141.9887989,	168.47077375]	
		7	17303	[186.7667283,	165.9336646,	189.43406911]	
		8	13969	[189.29901118,	110.0635569,	142.64710519]	
		9	12505	[168.74087416,	84.22654499,	120.65802113]	
		10	10136	[65.3156287,	50.74694521,	115.34903429]	
		11	9255	[57.30167296,	29.81014571,	75.09217485]	
		12	25493	[215.1653565,	165.218497,	179.91392186]	
		13	18917	[122.88591668,	94.73540918,	147.17218228]	
		14	21129	[147.61900932,	112.5705162,	157.37048777]	
		15	13072	[99.55285736,	47.37176344,	93.60418263]	

Table 1: Summary of clustering results by K=2,4,8,16 under k-means method

K	Objective Error			Runing time	
	k-means++	Random	Difference (k-means++ - Random)	k-means++	Random
2	844,611,707.20	844,609,209.08	2,498.12	2.38 sec.	4.15 sec.
4	304,068,436.97	304,071,752.09	-3,315.12	10.66 sec.	13.46 sec.
8	150,632,490.90	150,635,689.57	-3,198.67	43.33 sec.	48.14 sec.
16	79,464,617.65	79,469,903.58	-5,285.93	153.38 sec	252.33 sec.

Table 2: The Comparing of Objective Error and Running Time between two initialization methods

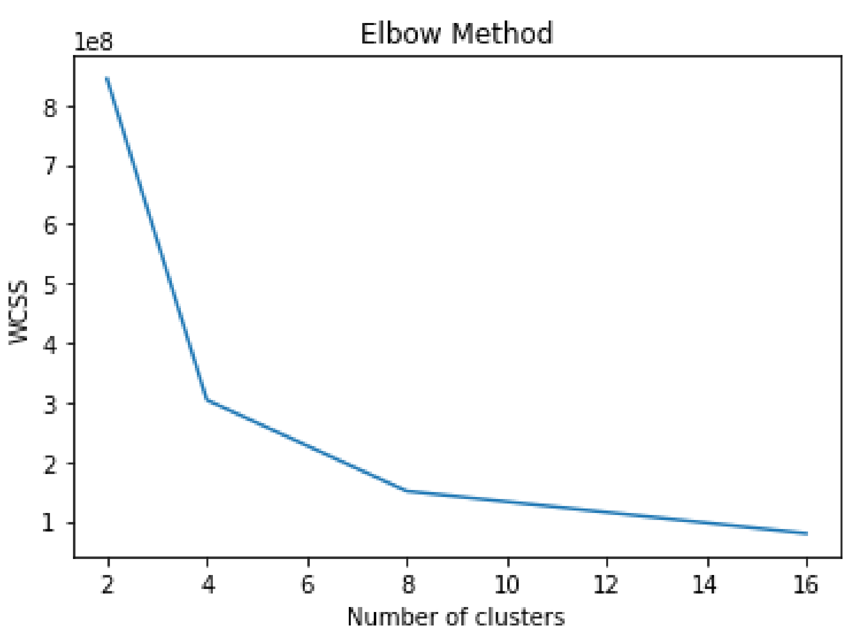
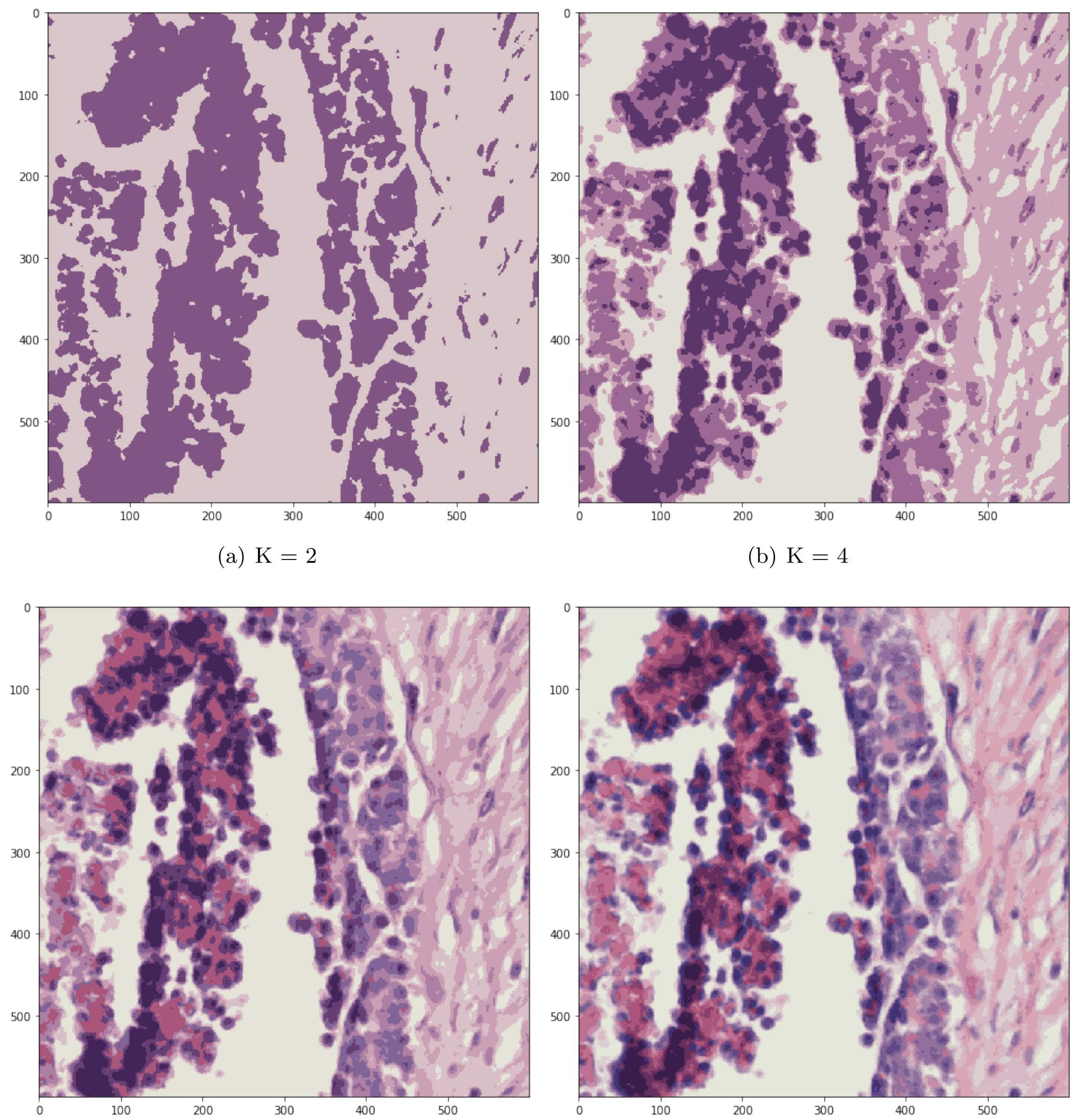


Figure 2: Distance Score of Elbow Graph for k-means Clustering (k-means++ initialization)



(a) K = 2 (b) K = 4

(c) K = 8 (d) K = 16

Figure 3: The clustered images which RGB based on the each centroid vector under different K by k-means clustering

.2 Agglomerative Hierarchical Clustering

K	Objective Error	Label	Size	Centroid			Linkage
2	844,611,414.44	0	142,082	[129.44910222,	85.34265749,	131.4278851]	complete
		1	217,918	[216.5807218 ,	198.29819436,	201.96856984]	
4	304,074,002.70	0	84,173	[155.77801391,	106.19744497,	147.79412533]	
		1	96,250	[204.20343209,	166.47184942,	184.81345826]	
		2	57,732	[91.02382848,	55.03218661,	107.66754343]	
		3	121,845	[226.43214644,	223.47862272,	215.52306972]	
8	150,624,302.10	0	33,371	[128.33746965,	100.15313371,	150.98276534]	
		1	43,542	[172.41442124,	128.96253417,	164.54055544]	
		2	23,436	[170.73155965,	86.88108638,	122.43075808]	
		3	56,449	[216.34266627,	191.89525261,	200.69658521]	
		4	52,049	[203.58228894,	161.31716131,	180.76839536]	
		5	20,486	[65.70401093,	37.65770469,	89.01727335]	
		6	33,505	[103.8803842 ,	62.92724615,	116.40445651]	
		7	97,162	[228.23680501,	229.02837091,	217.81408152]	
16	79,650,656.17	0	19,988	[177.83604837,	145.87622427,	176.59439336]	
		1	35,172	[219.96274086,	205.71889388,	208.71440346]	
		2	16,265	[192.09010449,	172.83103872,	193.19114935]	
		3	10,865	[66.66936746,	51.17705552,	115.47417365]	
		4	25,543	[211.44712743,	158.205933,	175.52610363]	
		5	10,002	[58.7803902 ,	30.34037019,	75.70435218]	
		6	14,189	[104.14098314,	49.25290923,	94.91099513]	
		7	17,473	[121.6112065 ,	97.55786401,	150.43028846]	
		8	19,283	[151.48395958,	123.69873024,	165.72422907]	
		9	15,829	[190.06597925,	122.904542,	155.1239246]	
		10	10,571	[185.9063447 ,	93.05568182,	124.12632576]	
		11	15,664	[97.3639903 ,	74.25689303,	134.47447026]	
		12	14,011	[143.9361125 ,	69.06602898,	110.05724891]	
		13	34,727	[217.21772801,	182.56699102,	194.0156322]	
		14	14,519	[153.01648503,	100.08566699,	143.98730859]	
		15	85,899	[228.91842746,	230.99167627,	218.48284613]	

Table 3: Summary of the clustering results by K=2,4,8,16 under agglomerative hierarchical clustering with complete linkage in method 1

K	Objective Error	Label	Size	Centroid			Linkage
2	844,610,215.42	0	217,864	[216.59288775,	198.3208908,	131.4278851]	single
		1	142,136	[129.47766795,	85.36906321,	201.96856984]	
4	304,091,660.82	0	122,706	[226.3407894 ,	223.19857879,	147.79412533]	
		1	84,174	[154.87937161,	105.34288363,	184.81345826]	
		2	96,476	[203.72517414,	165.71608342,	107.66754343]	
		3	56,644	[90.33497275,	54.48628901,	215.52306972]	
8	150,624,302.10	0	33,505	[103.8803842 ,	62.92724615,	150.98276534]	
		1	33,371	[128.33746965,	100.15313371,	164.54055544]	
		2	43,542	[172.41442124,	128.96253417,	122.43075808]	
		3	20,486	[65.70401093,	37.65770469,	200.69658521]	
		4	52,049	[203.58228894,	161.31716131,	180.76839536]	
		5	23,436	[170.73155965,	86.88108638,	89.01727335]	
		6	97,162	[228.23680501,	229.02837091,	116.40445651]	
		7	56,449	[216.34266627,	191.89525261,	217.81408152]	
16	79,459,955.78	0	10,204	[65.69080043,	51.0577055,	176.59439336]	
		1	25,440	[215.36280716,	165.53107922,	208.71440346]	
		2	13,861	[189.47290178,	110.54138702,	193.19114935]	
		3	14,855	[95.40641388,	74.13076509,	115.47417365]	
		4	17,414	[187.18183905,	166.27327517,	175.52610363]	
		5	21,214	[147.92885767,	112.87893074,	75.70435218]	
		6	18,442	[197.22699919,	142.37538628,	94.91099513]	
		7	9,468	[57.49430019,	29.99187249,	150.43028846]	
		8	13,136	[99.99634397,	47.55160332,	165.72422907]	
		9	35,400	[215.08390561,	185.99923696,	155.1239246]	
		10	12,819	[135.49992202,	65.21116656,	124.12632576]	
		11	18,937	[123.29292395,	94.99466258,	134.47447026]	
		12	19,606	[165.84188754,	135.74745927,	110.05724891]	
		13	84,107	[228.99250113,	231.24775985,	194.0156322]	
		14	12,521	[169.3696868 ,	84.71636306,	143.98730859]	
		15	32,576	[220.90973565,	207.79856314,	218.48284613]	

Table 4: Summary of the clustering results by K=2,4,8,16 under agglomerative hierarchical clustering with single linkage in method 1

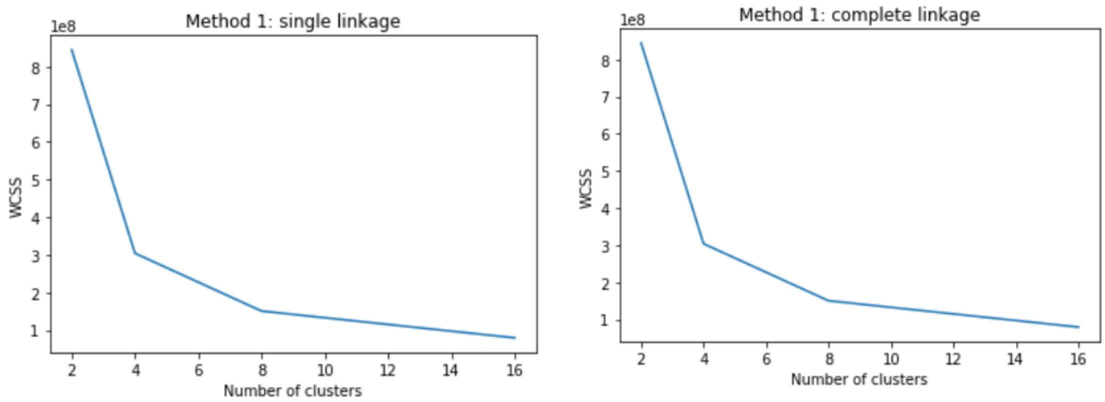
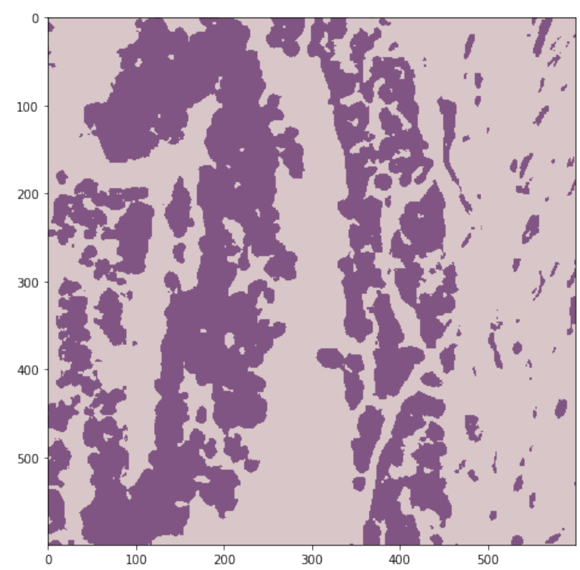
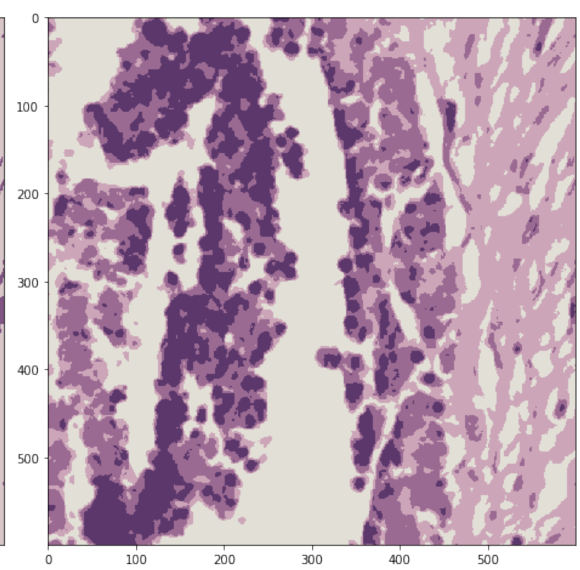


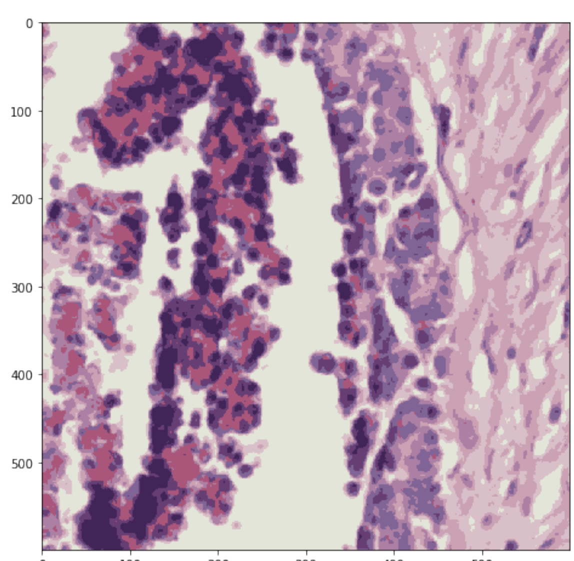
Figure 4: Distance Score of Elbow Graph for agglomerative hierarchical clustering with Two Linkage in Method 1



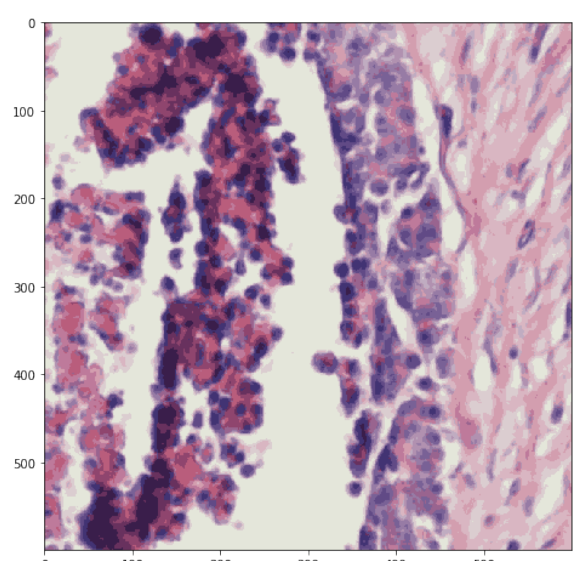
(a) K = 2



(b) K = 4

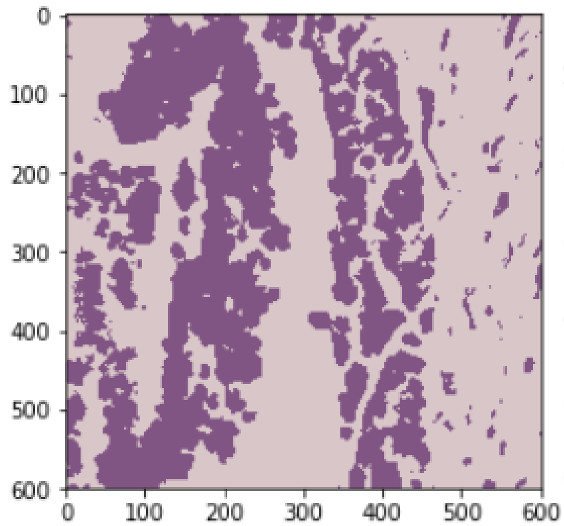


(c) K = 8

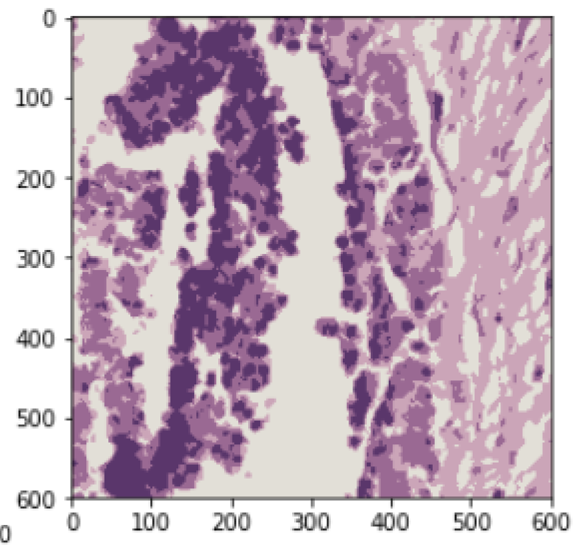


(d) K = 16

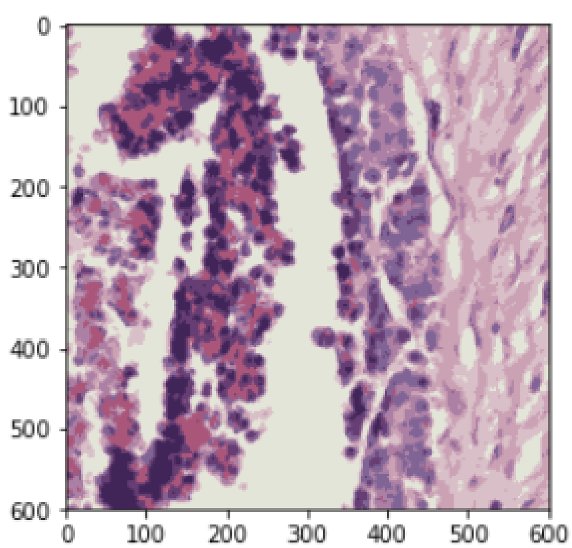
Figure 5: The clustered images which RGB based on the centroid vectors under different K by agglomerative hierarchical clustering with complete linkage in method 1



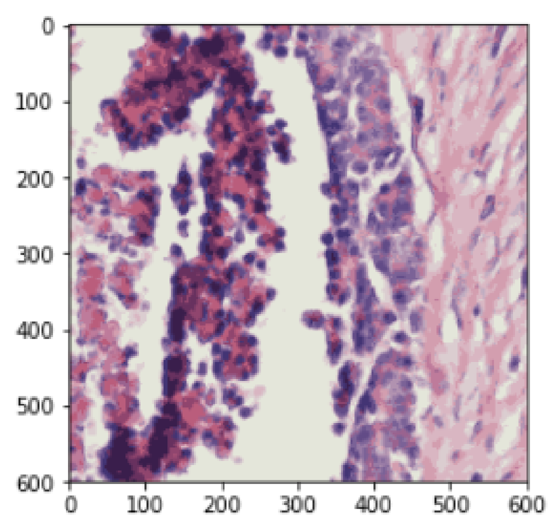
(a) K = 2



(b) K = 4

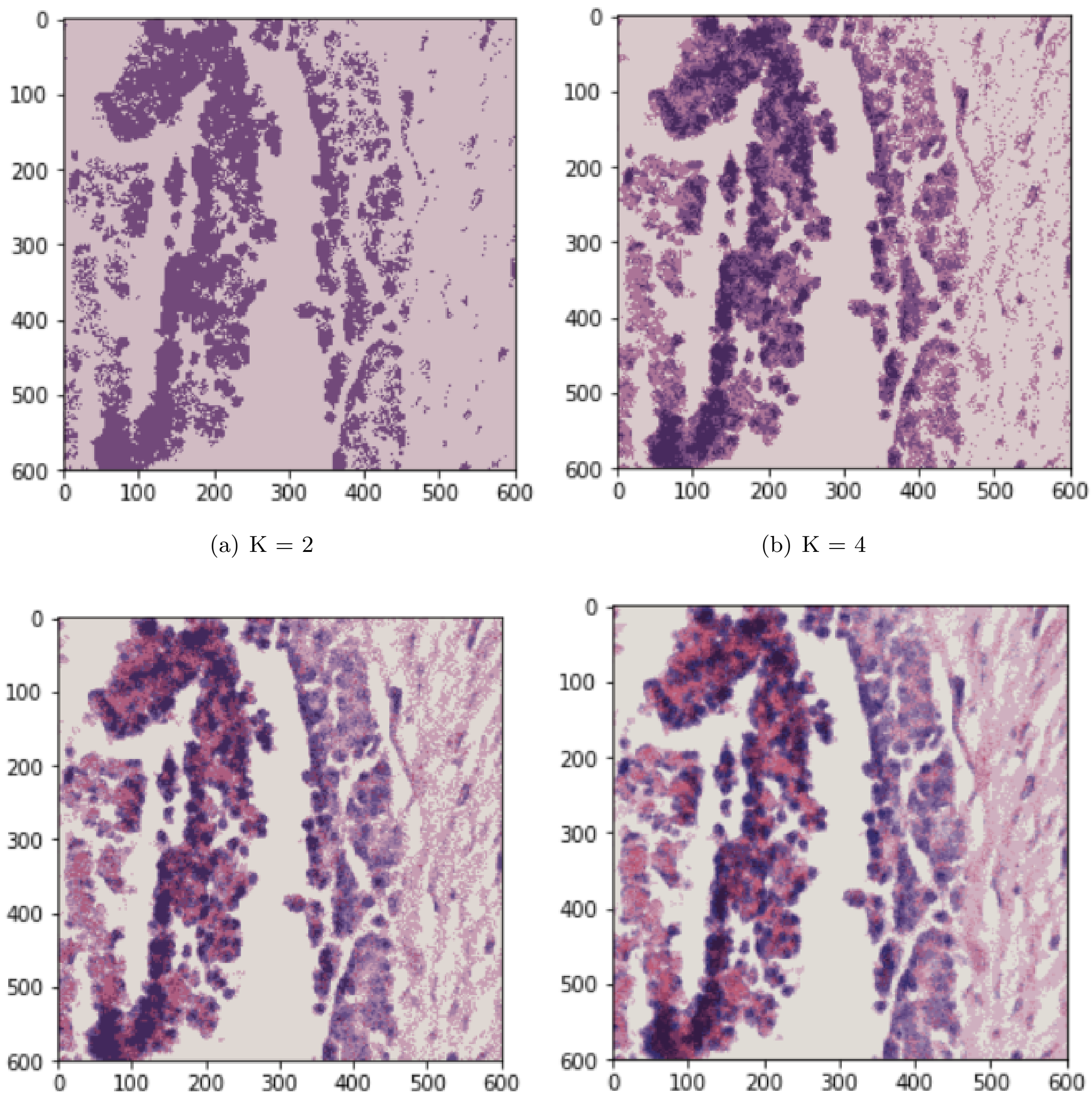


(c) K = 8



(d) K = 16

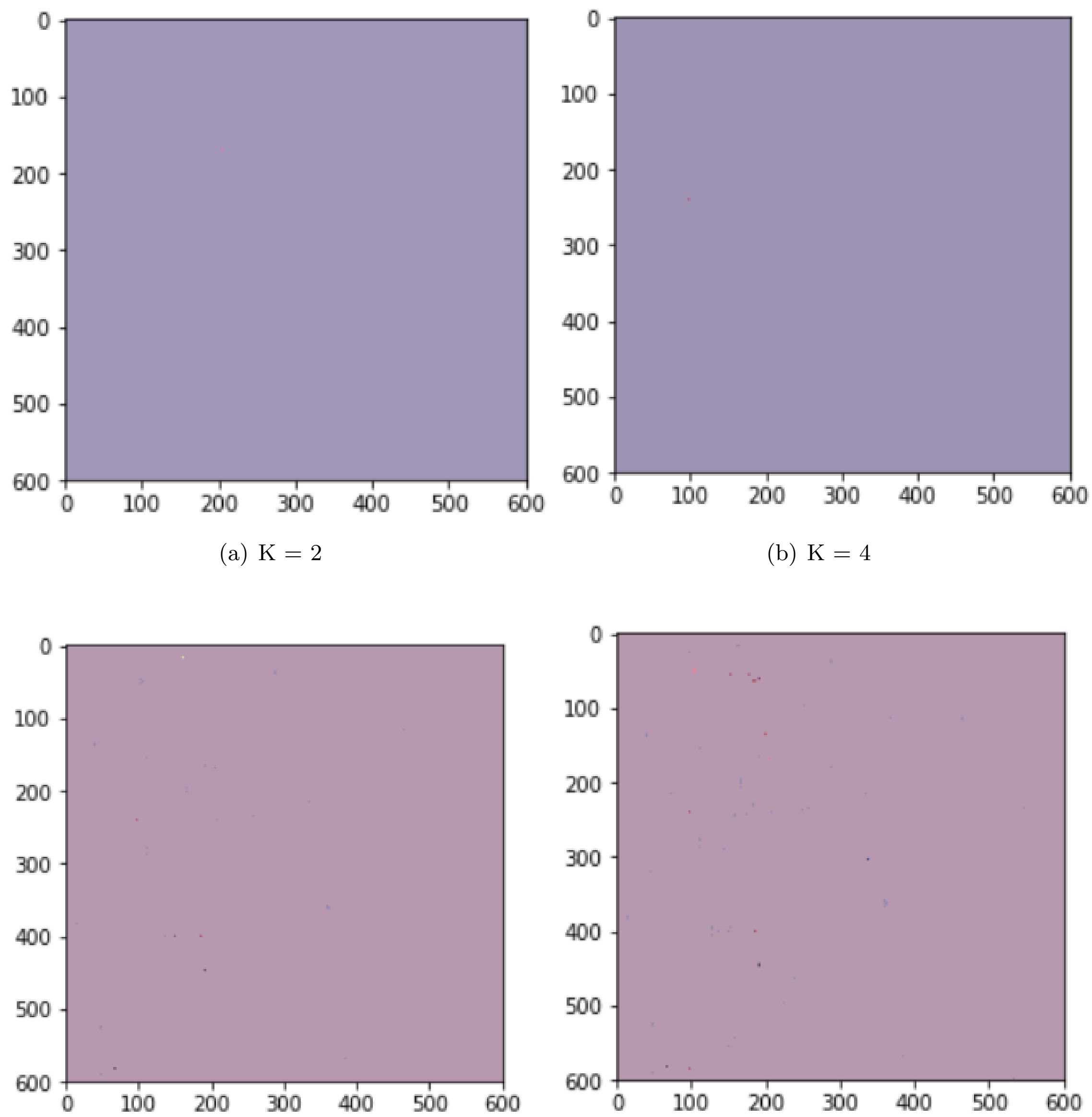
Figure 6: The clustered images which RGB based on the centroid vectors under different K by agglomerative hierarchical clustering with single linkage in method 1



(a) K = 2 (b) K = 4

(c) K = 8 (d) K = 16

Figure 7: The clustered images which RGB based on the centroid vectors under different K by agglomerative hierarchical clustering with complete linkage in method 2



(a) K = 2

(b) K = 4

(c) K = 8

(d) K = 16

Figure 8: The clustered images which RGB based on the centroid vectors under different K by agglomerative hierarchical clustering with single linkage in method 2

.3 Spectral Clustering

K	Objective Error	Label	Size	Centroid		
2	844,611,414.44	0	142,082	[129.44910222,	85.34265749,	131.42788512]
		1	217,918	[216.5807218 ,	198.29819436,	201.96856984]
4	304,095,351.48	0	96,450	[203.69326838,	165.6653884,	184.34156605]
		1	122,789	[226.33496844,	223.1810328,	215.38706397]
		2	84,171	[154.84586846,	105.29917833,	147.18722428]
		3	56,590	[90.30704549,	54.47514604,	107.15804567]
8	150,623,434.98	0	52,013	[203.51802573,	161.22784518,	180.71910631]
		1	56,520	[216.33164042,	191.82737569,	200.65742346]
		2	33,431	[103.80785439,	62.81714935,	116.2798948]
		3	43,533	[172.29708747,	128.87449467,	164.49384417]
		4	33,382	[128.14555216,	100.01874513,	150.91089317]
		5	97,238	[228.23201505,	229.01546561,	217.80667983]
		6	23,472	[170.69278596,	86.88395367,	122.45290009]
		7	20,411	[65.62755952,	37.61124718,	88.96350544]
16	79,947,698.40	0	12,296	[120.40169064,	58.16369991,	103.10745347]
		1	34,683	[220.08889466,	206.07819037,	208.89806399]
		2	15,622	[193.18351676,	129.40433837,	159.78410545]
		3	18,144	[174.55108905,	146.83837883,	178.00325338]
		4	22,234	[155.09509478,	120.86834067,	162.47152098]
		5	25,871	[211.65332303,	159.58218702,	176.56731066]
		6	34,030	[217.32987814,	183.25127,	194.5225958]
		7	85,601	[228.93243259,	231.03169339,	218.49486763]
		8	21,206	[131.83029731,	100.47574328,	150.32458707]
		9	13,865	[184.35937387,	99.1984419,	132.1944745]
		10	16,184	[102.61035102,	80.65362471,	139.41942673]
		11	12,127	[72.50490236,	56.29142292,	120.86924281]
		12	10,448	[90.31332694,	42.11994247,	87.87881112]
		13	12,690	[154.31917743,	75.43704696,	114.96438702]
		14	8,753	[52.11402104,	30.26097896,	78.73307411]
		15	16,246	[192.60944814,	173.18422025,	193.28812515]

Table 5: Summary of the clustering results by K=2,4,8,16 under Spectral Clustering in Method 1

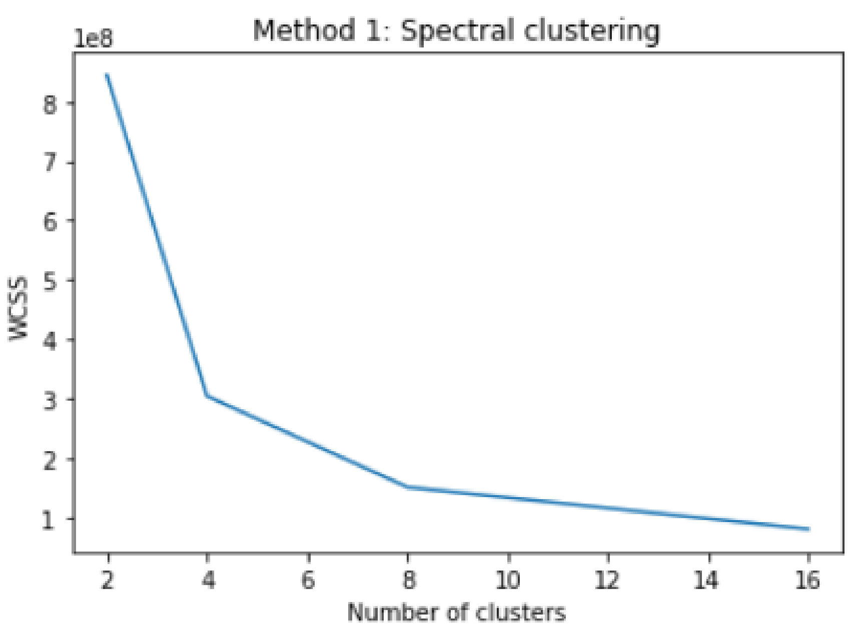
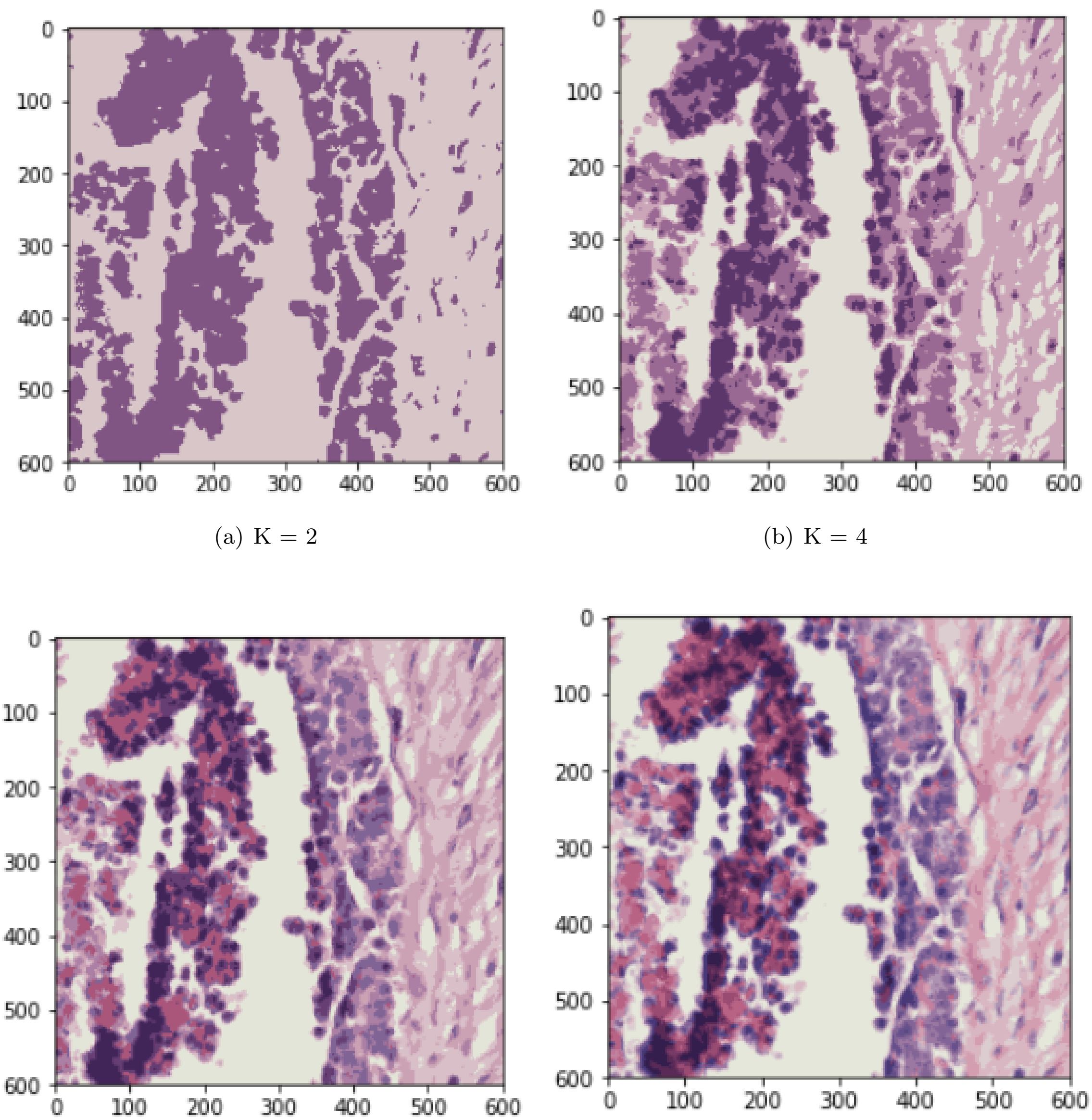


Figure 9: Distance Score of Elbow Graph for Spectral Clustering



(a) K = 2

(b) K = 4

(c) K = 8

(d) K = 16

Figure 10: The clustered images which RGB based on the centroid vectors under different K by spectral clustering in Method 1