Project 3
Color Image Compression Using Unsupervised Learning (Clustering)

Zhihua Zhang
zzhang78@vols.utk.edud
ECE 571

Abstract

Image data always needs to be compressed due to the storage limit, and several clustering algorithms are commonly used to compress the image. In this project, we applied four algorithms, K-Means, Winner-take-all, SOM Map and Mean Shift, to cluster the image, and compressed the image data by converting true color image to indexed color image. From the results, we found that, for K-Means, Winner-take-all, SOM Map, the optimal number of clusters is the same and is 16, the Winner-take-all has a slight better cluster result. For Mean Shift, the optimal number of clusters is provided automatically, and the cluster result is better than the other three algorithms.

1. Objective

The purpose of this project to compress the image data with clustering algorithm, since in some situations, only a limited number of colors is allowed to be simultaneously displayed. Thus, we need to convert the true color images to indexed color images, which can be done by applying several clustering algorithms. In our project, the data is a 120x120 full-color images with a 3-dimensional feature space (RGB), and we used the clustering algorithms (K-Means, Winner-take-all, Kohonen feature maps and Mean shift) to compress the data and compared the compressed results with external evaluation metrics.

2. Data wrangling and cleaning

The dataset used here is *flowersm.ppm*, represted by a 120x120x3 matrix, we converted the matrix to 2-dimension matrix with each row as the pixels and each column as the RGB colors. The new matrix is a 14400x3 matrix, which is ready for clustering.

3. Implementation

3.1 K-Means

K-Means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. First, we initialize k random centroids, then assign each point to its nearest centroids. Based on the assigned points, we update the centroids, then repeat the above steps several times until it converges. The algorithm will converge when then it got the optimum centroids solution. The pseudocode of K-Means algorithm is shown below.

Initialize $m_i, i = 1, \dots, k$, for example, to k random x^t

$$b_i^t \leftarrow \begin{cases} 1 \; if \; \|x^t - m_i\| = \min_j \|x^t - m_j\| \\ 0 \; otherwise \end{cases}$$

Repeat

For all $m_i, i = 1, \dots, k$

$$m_i = \frac{\sum_t b_i^t x^t}{\sum_t b_i^t}$$

Until m_i converge

3.2 Winner-take-all

The winner-take-all algorithm is similar to K-Means, which begins with an arbitrary set of n cluster centers, the steps for winner-take-all algorithm are shown below

- Randomly initialize n cluster centers
- Find the nearest cluster center ω_a for each sample x, which is called the winner
- Update the winner by $\omega_a^{new} = \omega_a^{old} + \varepsilon(\boldsymbol{x} - \omega_a^{old})$, where ε is a known as learning rate, and should be small

- Repeat the above step, until the cluster centers converge.

3.3 SOM Map

SOM Map is an extension of the winner-take-all algorithm. Instead of just update the winner, the SOM map also updates the neighbors of the winner. The steps for SOM map are shown below.

- Initialize the topological relationships, represented by a grid network
- For each sample x, find the nearest cluster center ω_a in the grid
- Update the winning cluster center and its neighbors in the sense of this topological distance.
- Repeat the above step, until the cluster centers converge.

3.4 Mean Shift

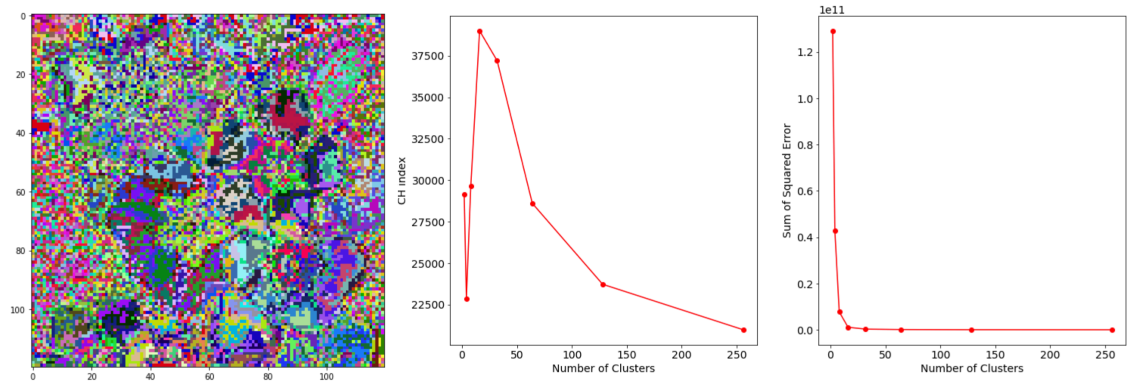
Compared with the above three algorithms, the Mean Shift does not require to specific the number of clusters, it will give us the final clusters based on the data. The steps of the Mean Shift algorithm are shown below.

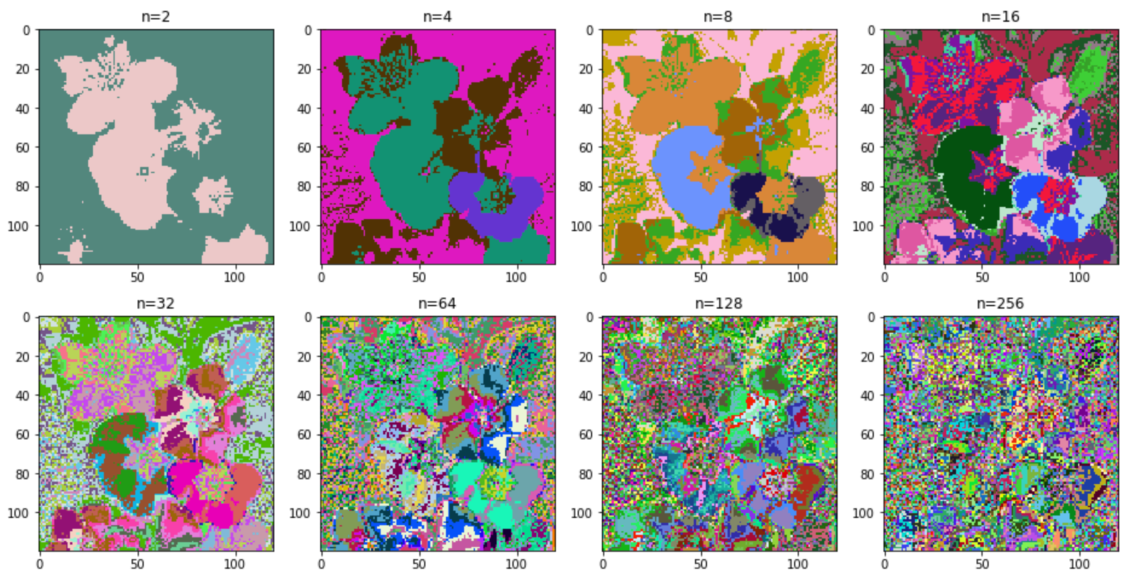
- Initialization: Choose a window/kernel $K(x)$ of size h,, e.g., a flat kernel or a Gaussian kernel
- Mean calculation: Within each window centered at x, compute the mean of data, where $\Omega(x)$ is the set of points enclosed within window h
- Mean shift: Shift the window to the mean, i.e., $x = m(x)$, where the difference $\mathrm{m}(x) - x$ is referred to as the mean shift
- Repeat the above steps, until $\mathrm{m}(x)$ converge

4. Results

4.1 K-Means

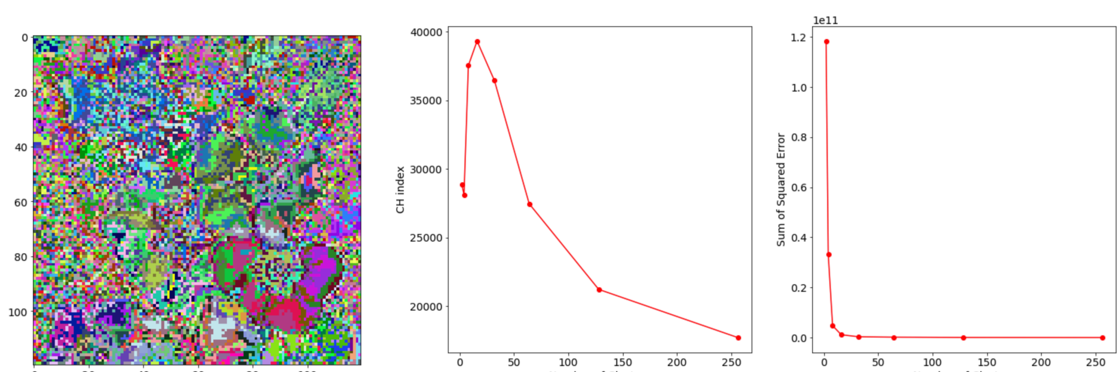
Below is the indexed color image with K-Means (n=256). Though we specified the number of clusters, we have some clusters with no sample data assigned, thus we finally got 248 clusters, and from the image we found the clustering result is not good since we it is totally different from the original image. Then to find the best number of clusters, we applied two external evaluation metrics (SSE and CH index), and the plots are shown below. The SSE is the sum of squared error, and the CH index is defined as ratio between the within-cluster dispersion and the between-cluster dispersion, which means the big the better. Thus, we can find that the best clustering results is with number of clusters equals 16. When number of clusters is 16, the CH index is the highest, and the SSE plot has the elbow. Then we plot the images with different number of clusters number. From the plot, we can also see that the number of clusters=16 is the best.

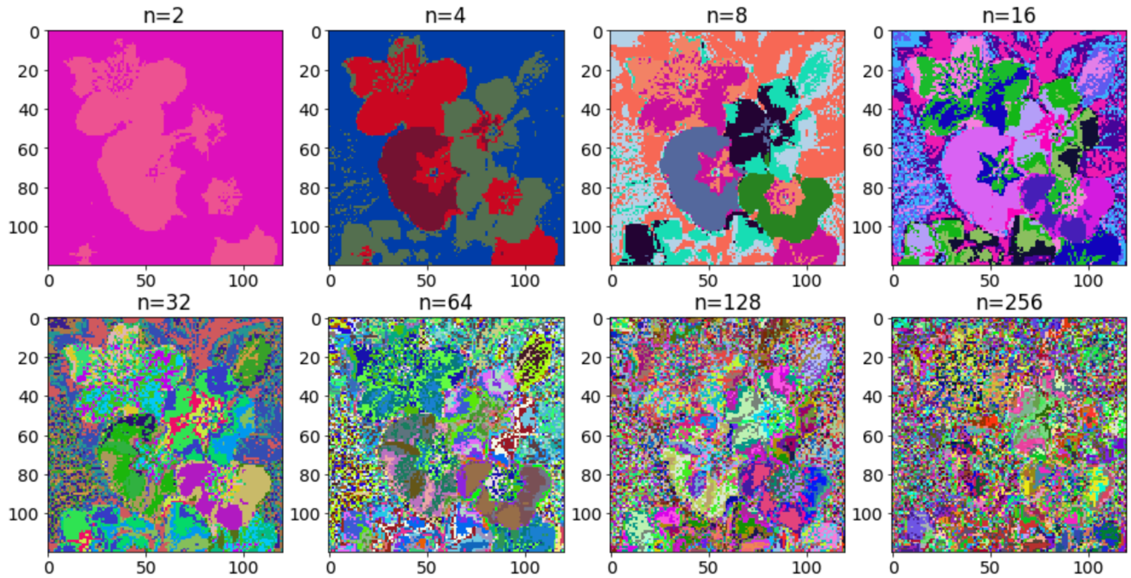




4.2 Winner-take-all

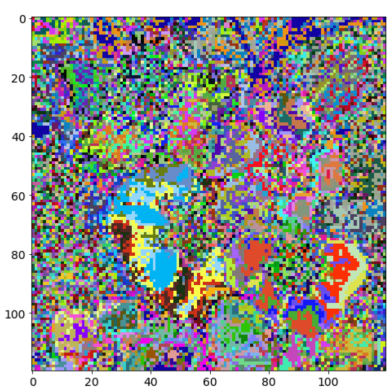
Then, we did the clustering with winner-take-all algorithm, below is the indexed color image with Winner-take-all (n=256). The clustering result is not good since we it is totally different from the original image. Similar to the results of K-Means, the best number of clusters for Winner-take-all is also 16, based on the results of SSE and CH index (shown below). In addition, the CH index with Winner-take-all algorithm is a little bit higher than that with K-Means.

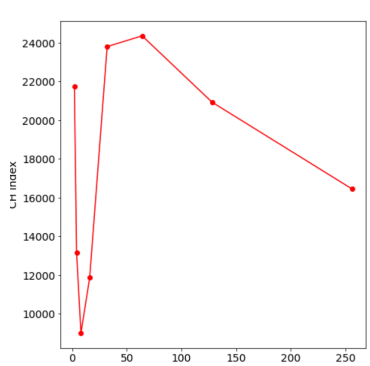


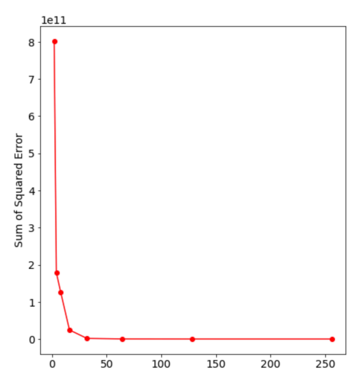


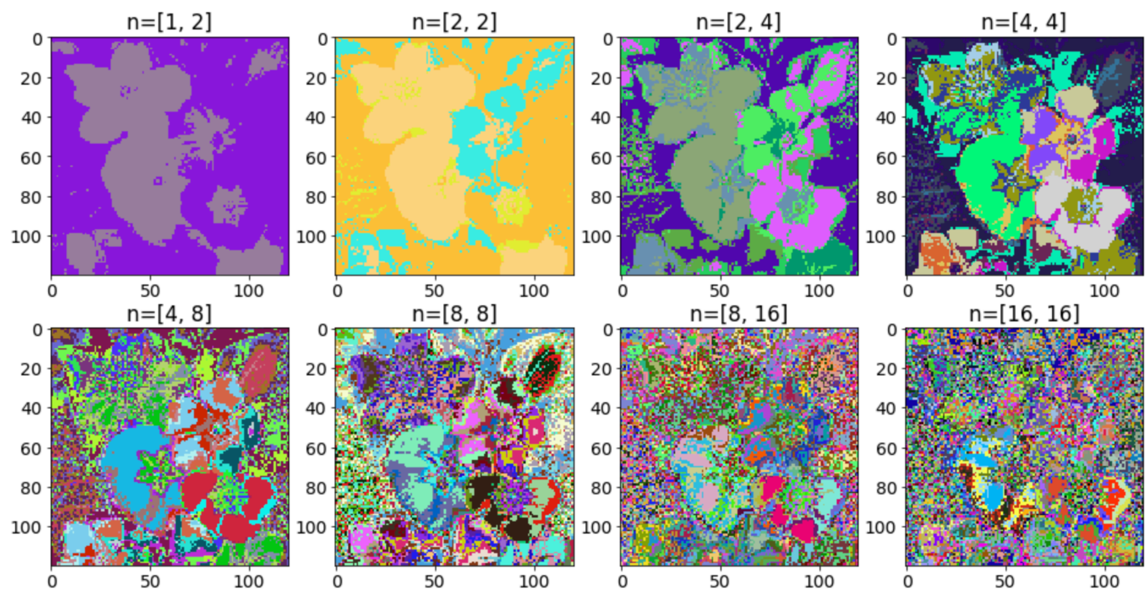
4.3 SOM Map

The same image data was clustered with SOM Map, and below is the indexed color image (n = 16x16). Since for SOM map we need to specify the grid, to get 256 clusters, we initialize the 16x16 grid, and it seems that the clustering result is not good. From the CH index plot, the best number of clusters is 64, but from the SSE plot, the best number of clusters is 16, which may indicate that more performance metrics are needed to get the accurate result. But from the indexed color images with different grid, we found that the best number of clusters is 16, which represented by a grid network matrix of 4x4.



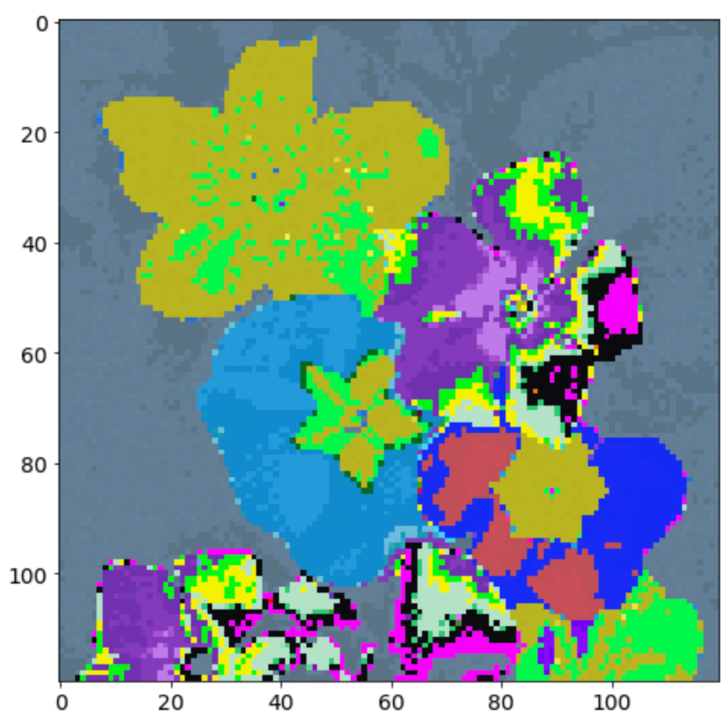






4.4 Mean Shift

Finally, we applied the mean shift algorithm to compress the image, which we do not need to specify the number of clusters. The mean shift algorithm automatically gives us the optimal cluster number. We set bandwidth=5, and use Gaussian kernel, the clustering results are shown below and is quite good.



5. Conclusion

In this project, we compressed the image data with different clustering algorithms, K-Means, Winner-take-all, SOM and Mean Shift. We found that K-Means, Winner-take-all and SOM have the similar results, and the best number of clusters is identical and is 16. Then we applied the mean shift algorithm, which can automatically give the optical number of clusters, and it seems that the mean shift is the best and gives the best indexed color image, compared with other three clustering algorithms.