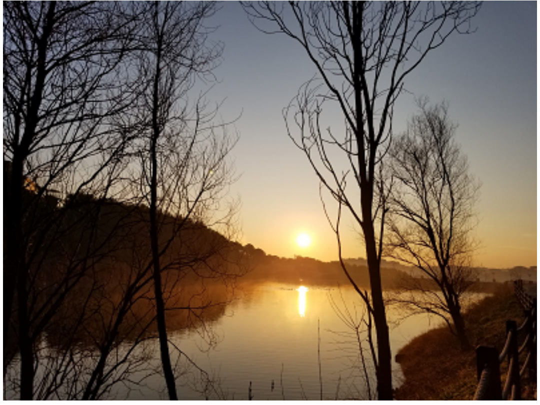
BBL558E – HOMEWORK2
MUHAMMET BOLAT (bolat20@itu.edu.tr) 708191019

GitHub Link: https://github.com/muhammetbolat/ITU_ComputerVision_HW2

ORIGINAL IMAGE

You can see the original image below. We'll play this image using Kmeans and mean-shift clustering algorithm and then we'll compare two algoritgm.

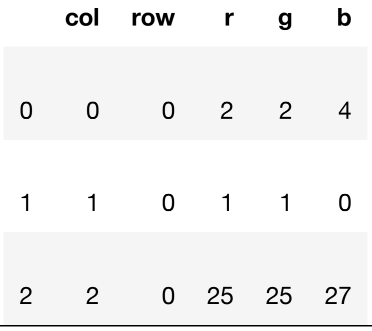
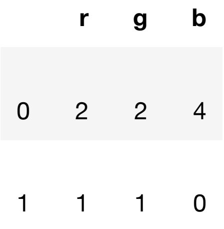
Sunny Lake Original Image



K-MEANS CLUSTERING

K-means clustering is one of the simplest unsupervised algorithm. It means that we don't have any labeled data. So, the first thing is that we need to decide how many cluster we should select for the best performance. There are lots of method to choose how many number of cluster is the best. The best famous are Elbow and Silhouette.

Normally, we have RGB pixels as feature vectors. I'm gonna play with them. That's I will introduce 4 different feature vectors.

<table>
<tr><td>RGB colours with their location on the images(5 features)</td><td>
<table>
<tr><th></th><th>col</th><th>row</th><th>r</th><th>g</th><th>b</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>2</td><td>2</td><td>4</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>2</td><td>2</td><td>0</td><td>25</td><td>25</td><td>27</td></tr>
</table></td></tr>
<tr><td>RGB colours without their location on the image(3 features)</td><td>
<table>
<tr><th></th><th>r</th><th>g</th><th>b</th></tr>
<tr><td>0</td><td>2</td><td>2</td><td>4</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table></td></tr>
</table>

	2 25 25 27
Grayscale values with their location on the image(3 features)	col row grayscale 0 0 0 2.666667 1 1 0 0.666667 2 2 0 25.666667
Grayscale values without their location on the image(1 feature	0 2.666667 1 0.666667 2 25.666667 3 61.333333 4 75.333333 dtype: float64

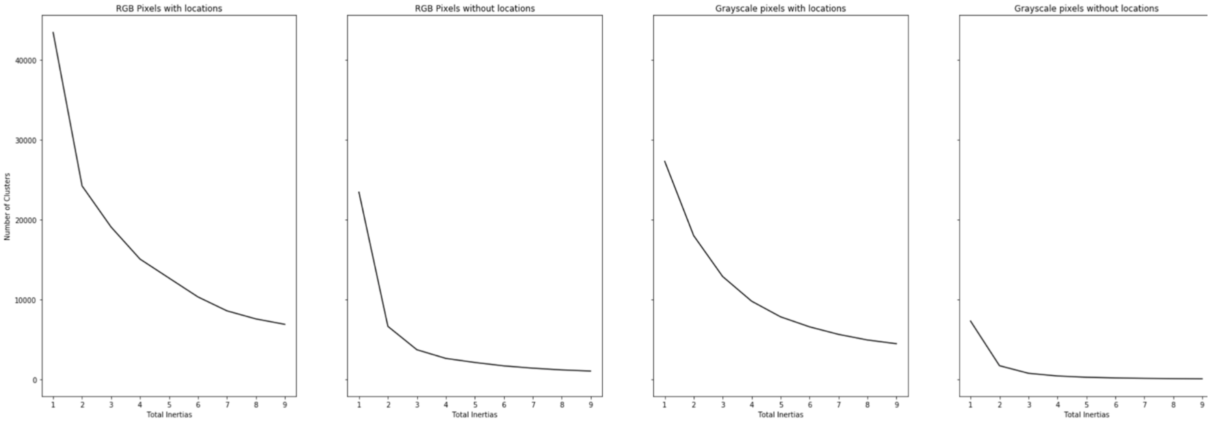
Normalize Features

Pixels are to set between 0-255 and locations are 0-300 and 0-400. These values are quite different for us. Image we have 5 dimension space. We must use euclidean distance of two feaute vector. That means locations have higher effect on the result. Because of this problem, we should set all the values between 0-1.

Cluster Number Selection

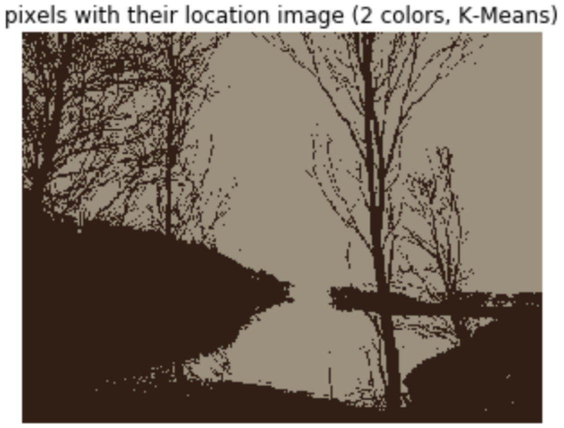
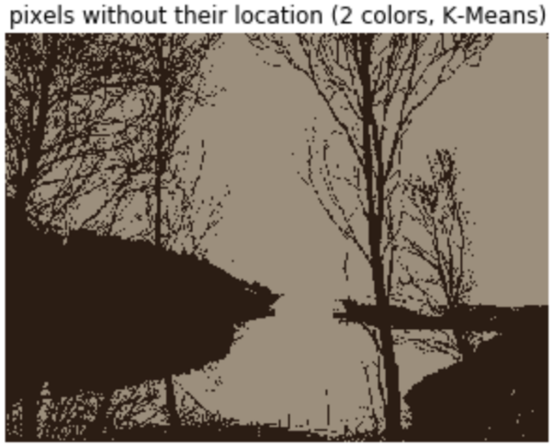
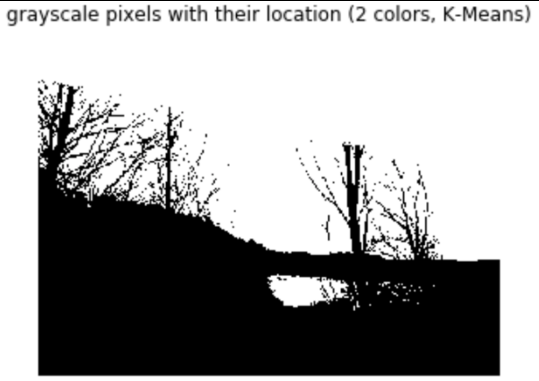
There are some method to choose cluster number of the unlabaled data. Remember mnist dataset. I'll remind you dataset. There are handwrite numbers from 0 to 10 in this dataset. We know how many clusters at the beginning of the fit the model and we do not need to find the cluster number. However, If we don't know like sunny lake image, it is critical to decide optimum number. For this, we will use a method called elbow.

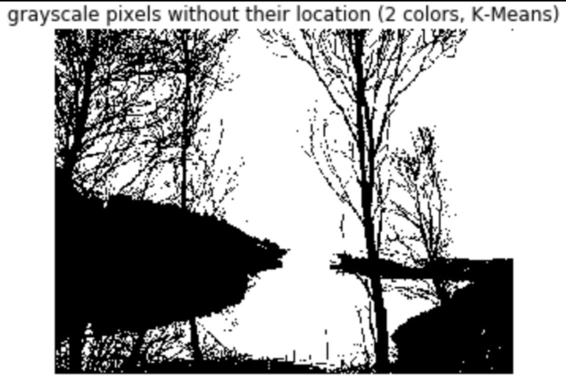
One method to validate the number of clusters is the elbow method. The idea of the elbow method is to run k-means clustering on the dataset for a range of values of k (say, k from 1 to 10), and for each value of k calculate the sum of inertieas. Then, plot a line chart of total inertias for each value of k. If the line chart looks like an arm, then the "elbow" on the arm is the value of k that is the best.



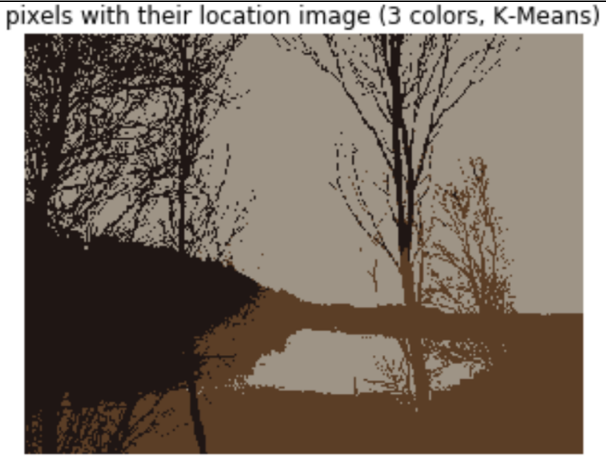
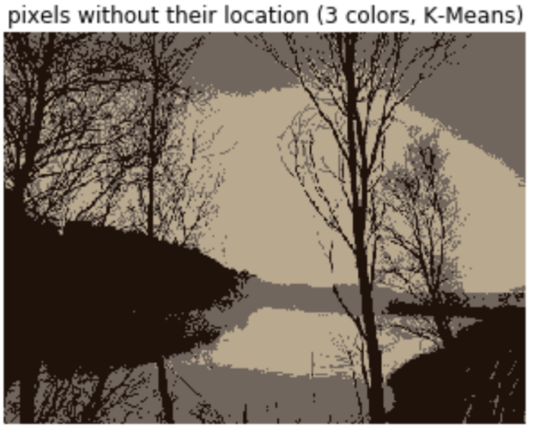
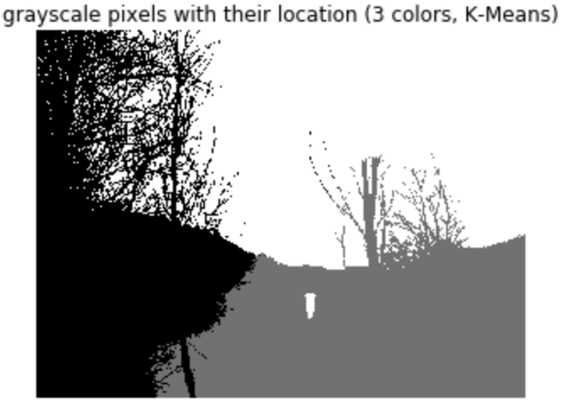
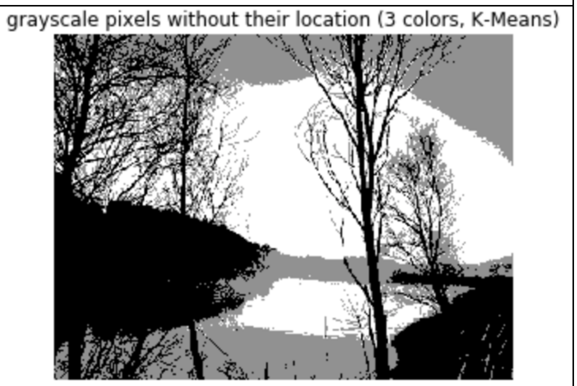
Let’s check elbow results.

Feature Vector	**Optimum Cluster**
RGB pixels with locations	2
RGB pixels without locations	2
Grayscale pixels with locations	2
Grayscale pixels without locations	2

	we add the location information on the picture, the lower parts of the sun appear to be more clearly separated. While calculating the euclidiean distance, 2 features come from location, 3 features come from colors. Therefore, the distance has been affected due to excessive color.
	Some regions under the sun could not be separated as we want in our clustering. Here, too, we feel the lack of location features. Kmeans algorithm only calculate the distance of points which has R, G and B features.
	In the grayscale image, 2 features consist of location and 1 feature consists of color. As the location features are high, the image seems to be divided into 2 parts in the middle. This is a situation we never want. In data such as pictures, only location information is very inefficient in clustering.

 grayscale pixels without their location (2 colors, K-Means)	Contrary to above locations, there is a cluster of only one feature which is gray-scale values. It can be seen, a more distinct clustering has been achieved by making a distinction by color.

Let's see how elbow method sets optimum value. I'm gonna set cluster value is 3.

 pixels with their location image (3 colors, K-Means)	 pixels without their location (3 colors, K-Means)
 grayscale pixels with their location (3 colors, K-Means)	 grayscale pixels without their location (3 colors, K-Means)

When the number of clusters is 3, the sky and the lake are clustered into 2 parts as you can seen. This appears to be an wrong clustering.

MEAN-SHIFT CLUSTERING

Mean-Shift algorithm is another known unsupervised clustering algorithm. The algorithm assşgns the feature vectors to the clusters iteratively by shifting points towards the highest density of feature points in region.
Mean-shift algorithm has so many fields. One of them are image processing and computer vision.

Unlike K-Means cluster algorithm, mean-shift does not require specifying the number of cluster in advance. The number of clusters is determined by algorithm with respect to data. As you remind, I mentioned that there are some algorithms to detect K-means cluster number like Silhouette or Elbow methods.

Kernel-Density Estimation

The first step to apply mean shift algorithm is respresenting our data as point in spatial domain. Mean shift builds on kernel density estimation. Kernel density estimation estimate PDF(probability density function) for a set of data.
It works by placing a kernel on each point in the data set. A kernel has a mathematical algorithm for weighting function generally user in convolution. There are many different kernels. But the most popular one is the Gaussian kernel. Gaussian kernel has a bandwith which is needed to start mean-shift algorithm. It is also called window.

There is a known python class which calculate gaussian kernel depending on data. Class is called as estimate_bandwidth.

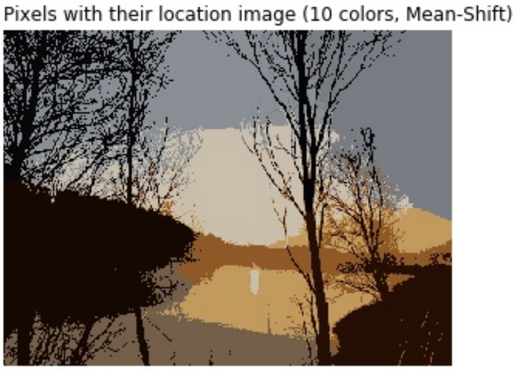
Bandwidth = estimate_bandwidth(data, quantile=.04, n_jobs=-1)

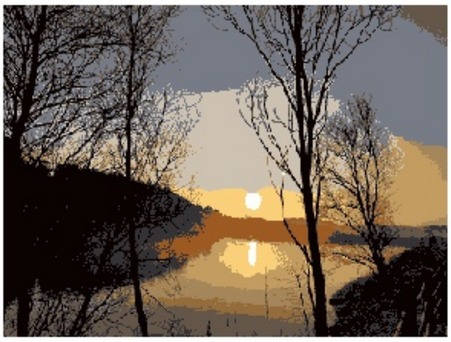
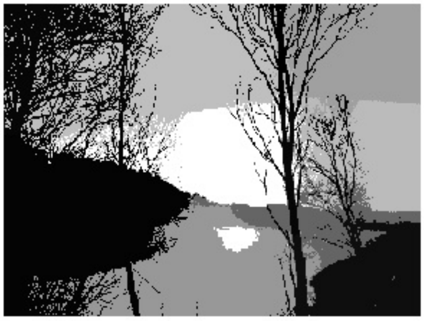
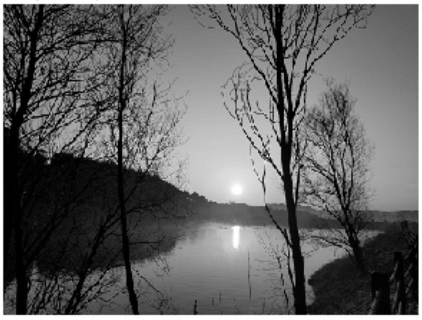
We’ll use this class to get bandwidth.

Feature Vector	**Window Size (Bandwidth)**
RGB pixels with locations	0.23298278428525732
RGB pixels without locations	0.07890197160942992
Grayscale pixels with locations	0.18561836089009423
Grayscale pixels without locations	0.0019921677559912826

Clustering

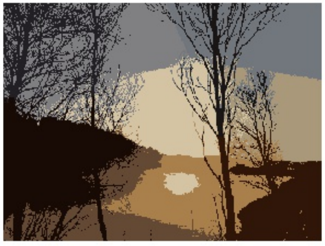
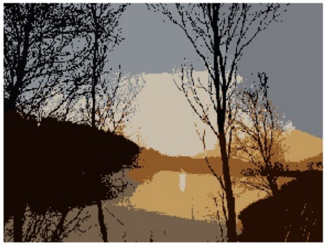
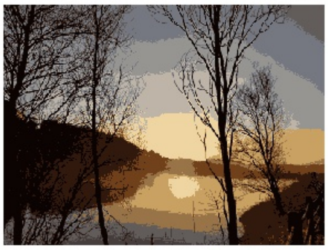
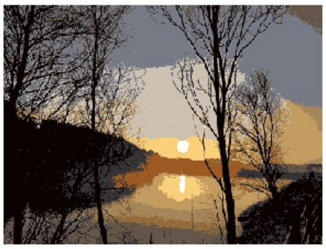
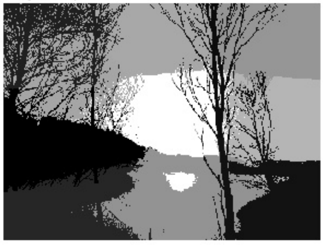
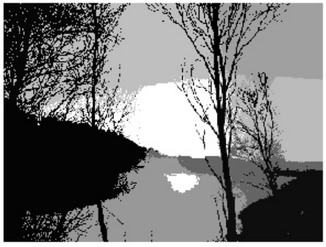
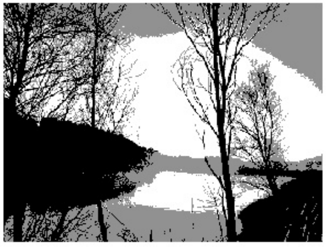
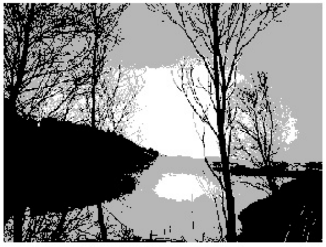
Accoring to the below result, we can calculate mean-shift algorithm.

 Pixels with their location image (10 colors, Mean-Shift)	Bandwidth: 0.23298278428525732 Mean-shift clustering algorithm determined the number of clusters is 10 according to the bandwidth. The location feature is 2 (height and width). 3 features were also came from the color space(Red, Green Blue).

Pixels without their location (11 colors, Mean-Shift) 	Bandwidth: 0.07890197160942992 Mean-shift clustering algorithm determined the number of clusters is 11 according to the bandwidth. We have only color features. Clustering is based on color space. As it can be seen between the above result and this result, the sun is not clustered above due to its location features.
Grayscale pixels with their location (11 colors, Mean-Shift) 	Bandwidth: 0.18561836089009423 2 features come from location, 1 feature comes from color space. Therefore, location information has become more effective. Trees and lake are clearly clustered. However, the sky should be a single cluster, it is divided into 2 clusters.
Grayscale pixels without their location (261 colors, Mean-S 	Bandwidth:0.0019921677559912826 Feature vector consists of one feature. Therefore, clustering was not done exactly as we wanted. A very inefficient clustering sample was obtained.

EVALUATE K-MEANS AND MEAN-SHIFT

So far so good. we have completed our clustering operations using k-means and mean-shift. Using different feature vectors, we examined how effective the location and color spaces are. However, since different number of clusters are used in both algorithms, it can be difficult to compare between them. For this, let's make a comparison using the same set number in both algorithms.

K-Means	Mean-Shift	Cluster
Pixels with their location image (10 colors, K-Means) 	Pixels with their location image (10 colors, Mean-Shift) 	10
Pixels without their location (11 colors, K-Means) 	Pixels without their location (11 colors, Mean-Shift) 	11
Grayscale pixels with their location (11 colors, K-Means) 	Grayscale pixels with their location (11 colors, Mean-Shift) 	11
Grayscale pixels without their location (3 colors, K-Means) 	Grayscale pixels without their location (3 colors, Mean-Shif 	3

References

1. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
2. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html
3. https://www.geeksforgeeks.org/ml-mean-shift-clustering/