assignment05

April 8, 2019

1 K-means clustering on color image

2 Name : Jinhyuk-Park

3 Student-ID : 20141775

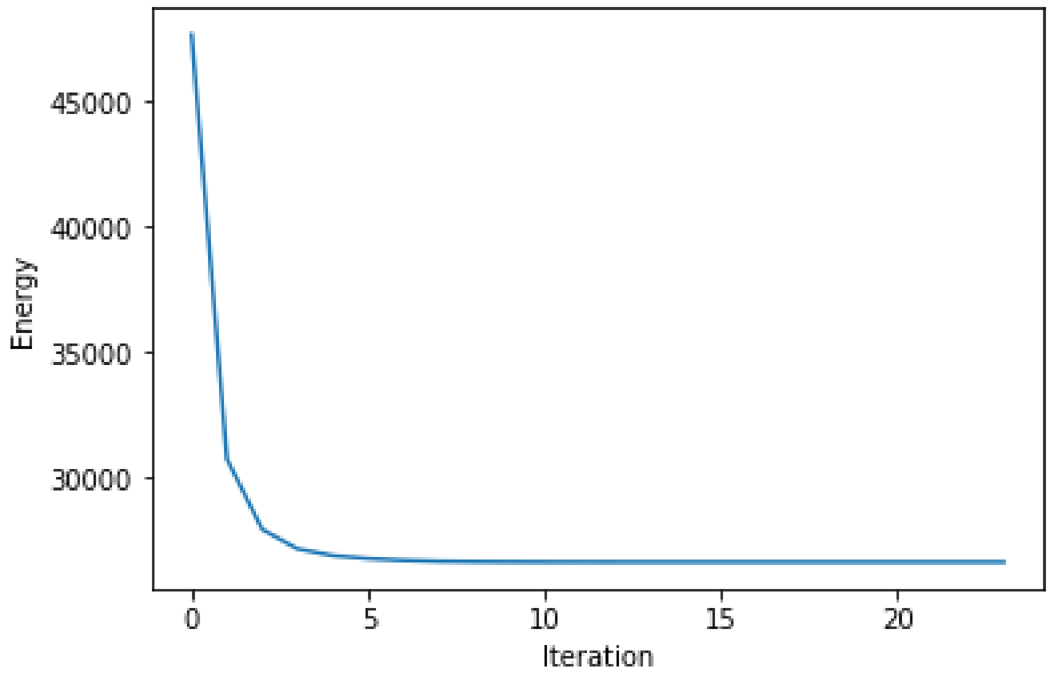
In [63]:
```
import PIL.Image as pilimg
import matplotlib.pyplot as plt
import numpy as np
import random


file_data                = pilimg.open("twosome.jpg")
data                        = np.array(file_data)
modifiedData                        = np.array(file_data)
size_row        = len(data[0]) # height of the image
size_col          = len(data)    # width of the image
size_rgb        = 3            # Red, Green, Blue

#
# normalize the values of the input data to be [0, 1]
#
def normalize(data):

    data_normalized = (data - min(data)) / (max(data) - min(data))

    return(data_normalized)

#
# example of distance function between two vectors x and y
#
def distance(x, y):

    d = (x - y) ** 2
    s = np.sum(d)
    #r = np.sqrt(s)
    return(s)

#
# calcuate the values of the input data in l2-norm
#
def norm(x):
    r = np.sqrt(x.T * x)

    return(r)
```

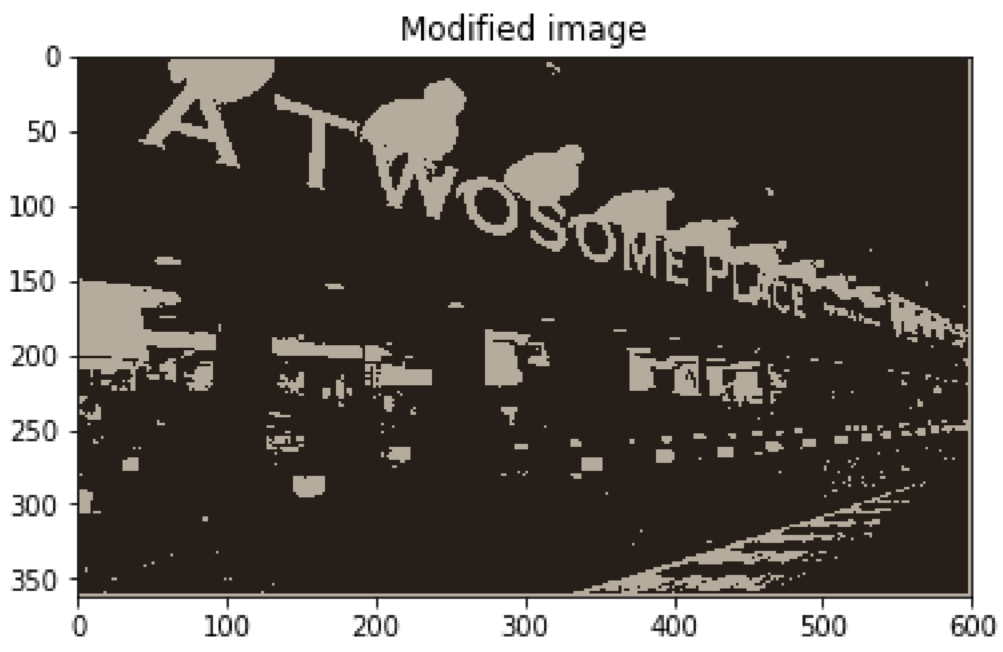
3.1 k = 2 clustering

```
In [64]: k = 2
         Energy = []

         list_centroid = np.zeros((k, size_rgb), dtype=float)
         list_count    = np.zeros(k)
         list_label    = np.empty((size_col, size_row), dtype=int)

         for i in range(size_col):
             for j in range(size_row):
                 label       = random.randint(0, k - 1)
                 list_label[i][j]       = label
                 list_centroid[label, :]+= data[i][j]
                 list_count[label]      += 1

         for i in range(0, k):
             list_centroid[i, :] /= list_count[i]

         while True:
             checkUpdate = 0
             sumEnergy   = 0
             for i in range(size_col):
                 for j in range(size_row):
                     label = list_label[i][j]
                     min = distance(list_centroid[label, :], data[i][j])
                     sumEnergy += min

                     for m in range(k):
                         if m == label:
                             continue

                         checkDistance = distance(list_centroid[m, :], data[i][j])
                         if(min > checkDistance):
                             list_label[i][j] = m
                             min = checkDistance
                             checkUpdate += 1
```

```
    Energy.append(np.sqrt(sumEnergy))
    if(checkUpdate == 0):
        break


    list_centroid = np.zeros((k, size_rgb), dtype=float)
    list_count = np.zeros(k)

    count = 0
    for i in range(size_col):
        for j in range(size_row):
            label = list_label[i][j]
            list_centroid[label, :] += data[i][j]
            list_count[label]       += 1
            count                   += 1
    for i in range(0, k):
        list_centroid[i, :] /= list_count[i]


#
# plot image
#
f1 = plt.figure(1)
plt.imshow(data)
plt.title('Input image')
plt.show()

x = np.arange(0, len(Energy), 1)
plt.plot(x, Energy)
plt.xlabel('Iteration')
plt.ylabel('Energy')
plt.show()

count = 0
for i in range(size_col):
    for j in range(size_row):
        label       = list_label[i][j]
        modifiedData[i][j] = list_centroid[label, :]
        count     += 1

plt.imshow(modifiedData)
plt.title('Modified image')
plt.show()
```

Input image
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600
A TWOSOME PLACE


Energy
45000
40000
35000
30000
0
5
10
15
20
Iteration

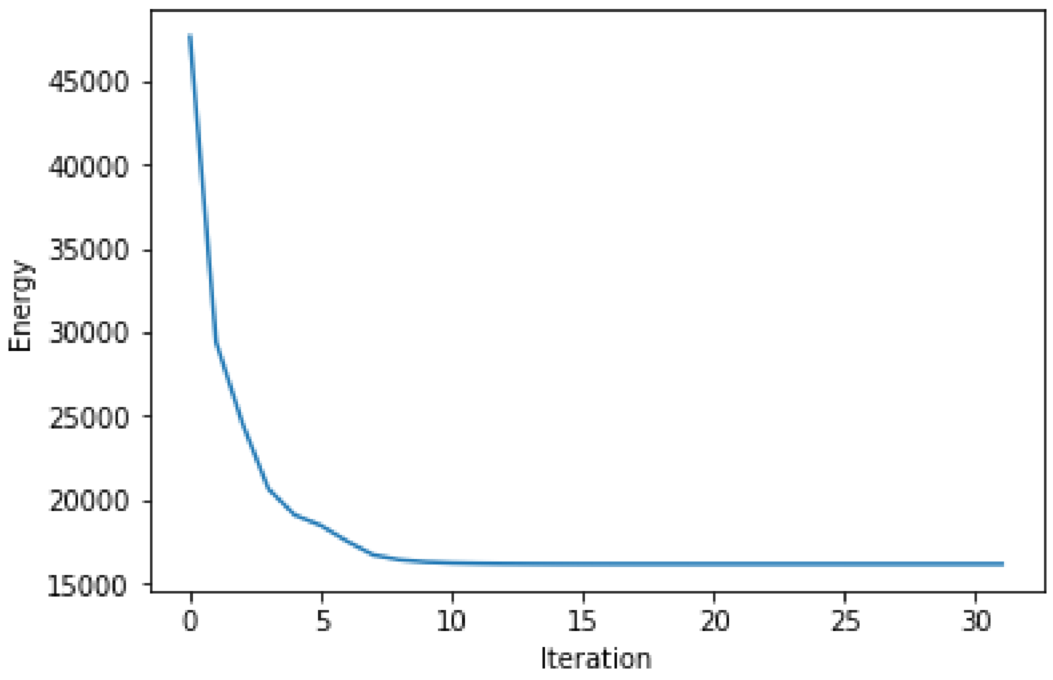


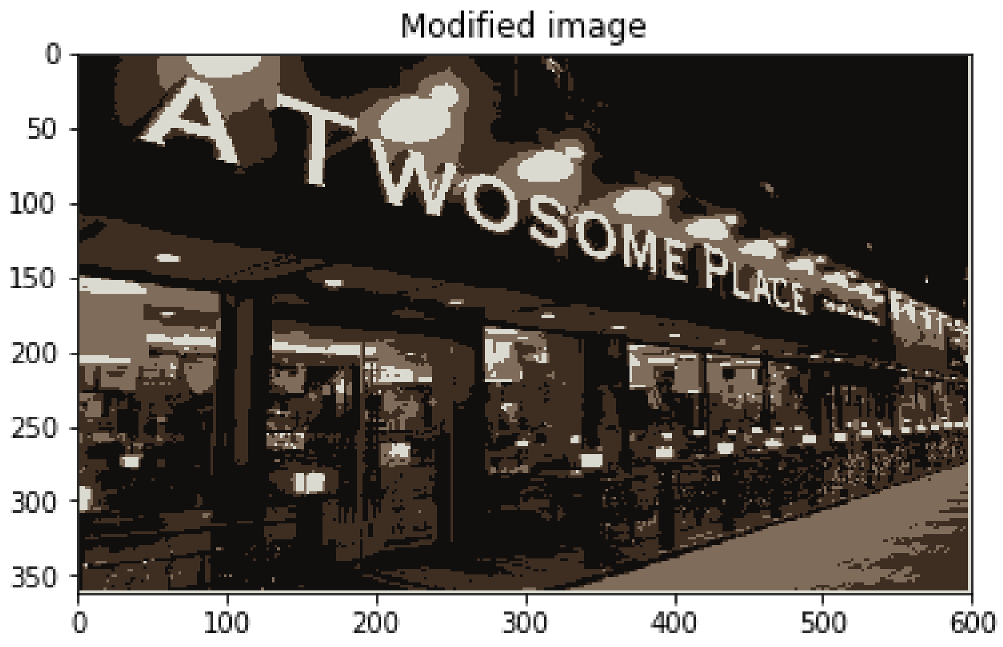
3.2 k = 4

```
In [65]: k = 4
         Energy = []

         list_centroid = np.zeros((k, size_rgb), dtype=float)
         list_count    = np.zeros(k)
         list_label    = np.empty((size_col, size_row), dtype=int)

         for i in range(size_col):
             for j in range(size_row):
                 label       = random.randint(0, k - 1)
                 list_label[i][j]       = label
                 list_centroid[label, :]+= data[i][j]
                 list_count[label]      += 1

         for i in range(0, k):
             list_centroid[i, :] /= list_count[i]

         while True:
             checkUpdate = 0
             sumEnergy   = 0
             for i in range(size_col):
                 for j in range(size_row):
                     label = list_label[i][j]
                     min = distance(list_centroid[label, :], data[i][j])
```

```
            sumEnergy += min

            for m in range(k):
                if m == label:
                    continue

                checkDistance = distance(list_centroid[m, :], data[i][j])
                if(min > checkDistance):
                    list_label[i][j] = m
                    min = checkDistance
                    checkUpdate += 1

    Energy.append(np.sqrt(sumEnergy))
    if(checkUpdate == 0):
        break


    list_centroid = np.zeros((k, size_rgb), dtype=float)
    list_count = np.zeros(k)

    count = 0
    for i in range(size_col):
        for j in range(size_row):
            label = list_label[i][j]
            list_centroid[label, :] += data[i][j]
            list_count[label]       += 1
            count                   += 1
    for i in range(0, k):
        list_centroid[i, :] /= list_count[i]


#
# plot image
#
f1 = plt.figure(1)

x = np.arange(0, len(Energy), 1)
plt.plot(x, Energy)
plt.xlabel('Iteration')
plt.ylabel('Energy')
plt.show()

count = 0
for i in range(size_col):
    for j in range(size_row):
        label      = list_label[i][j]
        modifiedData[i][j] = list_centroid[label, :]
        count     += 1
```

```
plt.imshow(modifiedData)
plt.title('Modified image')
plt.show()
```

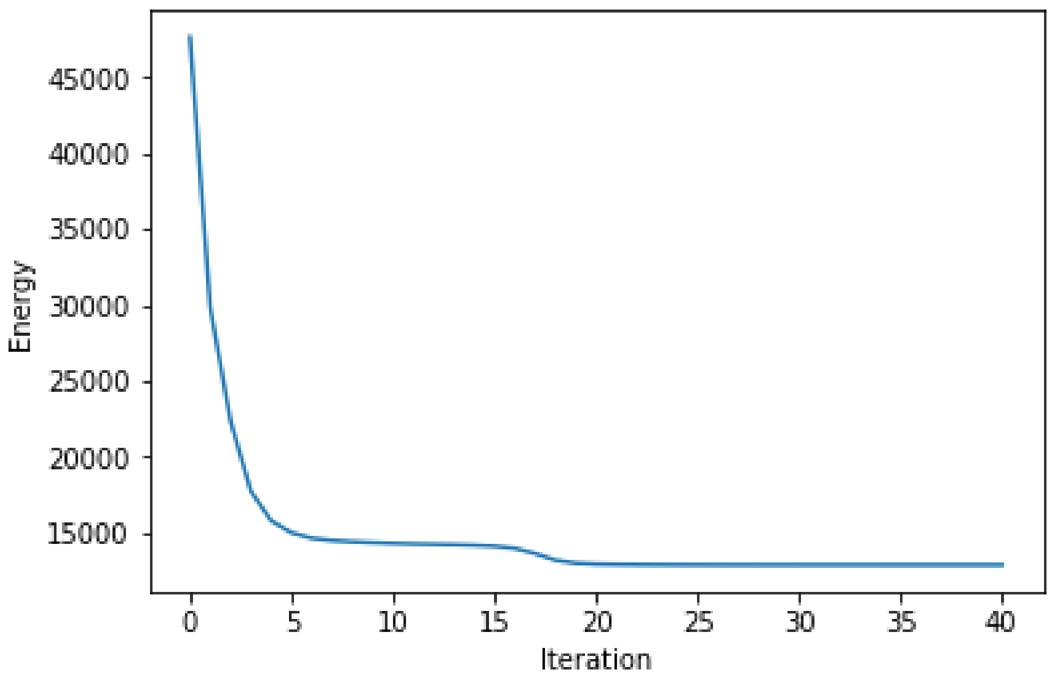


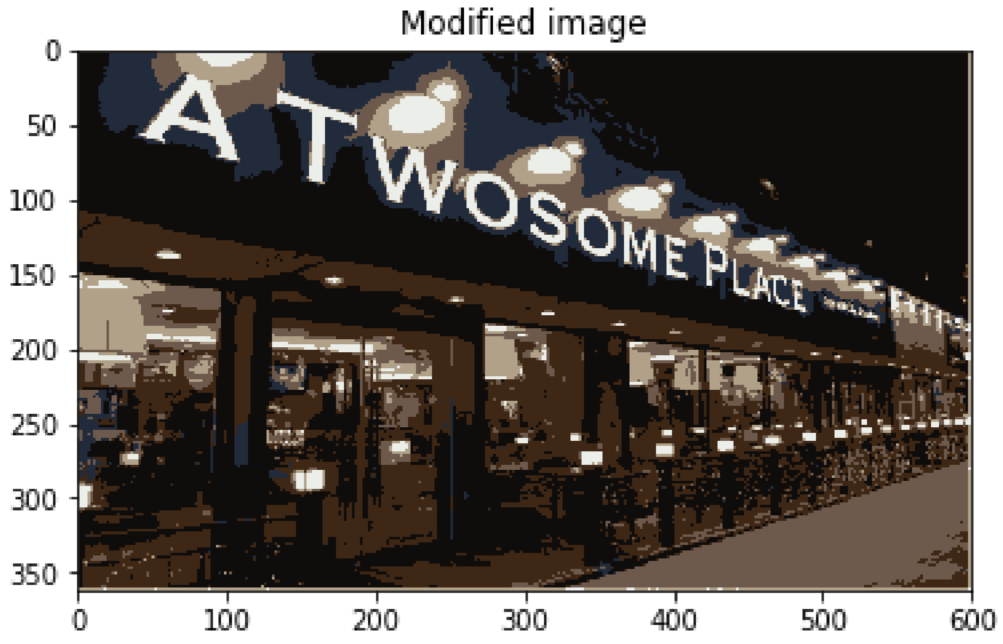


3.3 k = 6

```
In [66]: k = 6
         Energy = []

         list_centroid = np.zeros((k, size_rgb), dtype=float)
         list_count    = np.zeros(k)
         list_label    = np.empty((size_col, size_row), dtype=int)

         for i in range(size_col):
             for j in range(size_row):
                 label       = random.randint(0, k - 1)
                 list_label[i][j]       = label
                 list_centroid[label, :]+= data[i][j]
                 list_count[label]      += 1

         for i in range(0, k):
             list_centroid[i, :] /= list_count[i]

         while True:
             checkUpdate = 0
             sumEnergy   = 0
             for i in range(size_col):
                 for j in range(size_row):
                     label = list_label[i][j]
                     min = distance(list_centroid[label, :], data[i][j])
                     sumEnergy += min

                     for m in range(k):
                         if m == label:
                             continue

                         checkDistance = distance(list_centroid[m, :], data[i][j])
                         if(min > checkDistance):
                             list_label[i][j] = m
                             min = checkDistance
                             checkUpdate += 1

             Energy.append(np.sqrt(sumEnergy))
             if(checkUpdate == 0):
                 break


             list_centroid = np.zeros((k, size_rgb), dtype=float)
             list_count = np.zeros(k)
```

```
    count = 0
    for i in range(size_col):
        for j in range(size_row):
            label = list_label[i][j]
            list_centroid[label, :] += data[i][j]
            list_count[label]       += 1
            count                   += 1
    for i in range(0, k):
        list_centroid[i, :] /= list_count[i]


#
# plot image
#
f1 = plt.figure(1)

x = np.arange(0, len(Energy), 1)
plt.plot(x, Energy)
plt.xlabel('Iteration')
plt.ylabel('Energy')
plt.show()

count = 0
for i in range(size_col):
    for j in range(size_row):
        label       = list_label[i][j]
        modifiedData[i][j] = list_centroid[label, :]
        count     += 1

plt.imshow(modifiedData)
plt.title('Modified image')
plt.show()
```


Energy
45000
40000
35000
30000
25000
20000
15000
0
5
10
15
20
25
30
35
40
Iteration

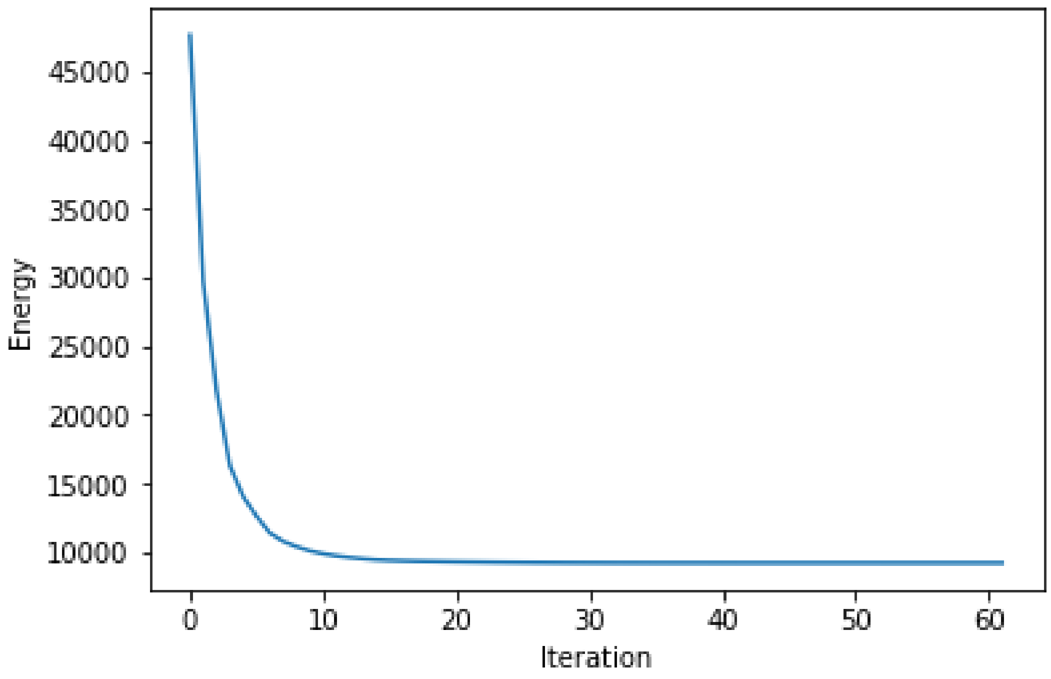

Modified image
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600
A TWOSOME PLACE

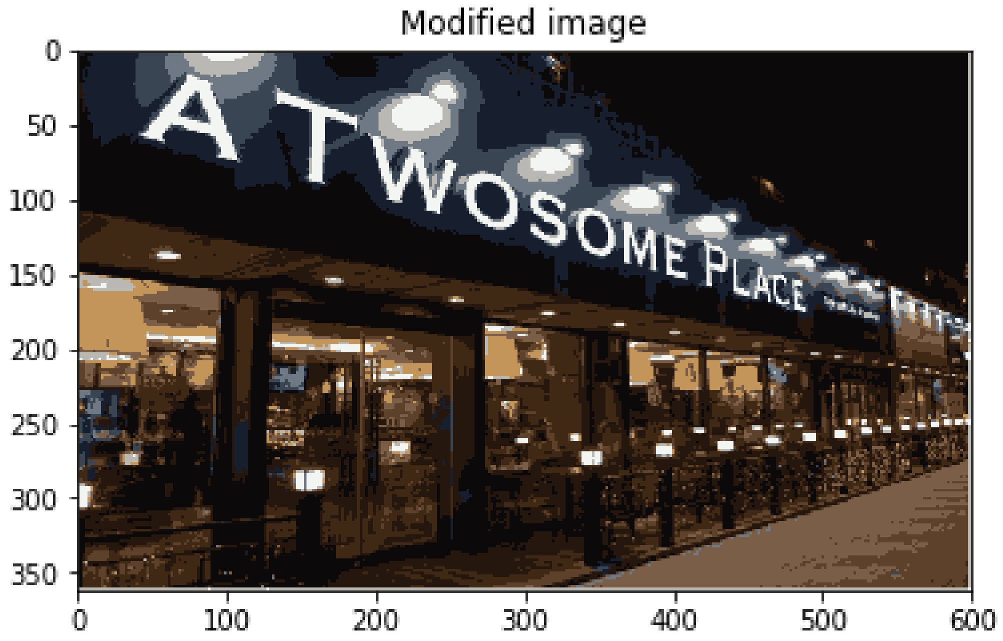
4 k = 15

```
In [67]: k = 15
         Energy = []

         list_centroid = np.zeros((k, size_rgb), dtype=float)
         list_count    = np.zeros(k)
         list_label    = np.empty((size_col, size_row), dtype=int)

         for i in range(size_col):
             for j in range(size_row):
                 label       = random.randint(0, k - 1)
                 list_label[i][j]       = label
                 list_centroid[label, :]+= data[i][j]
                 list_count[label]      += 1

         for i in range(0, k):
             list_centroid[i, :] /= list_count[i]

         while True:
             checkUpdate = 0
             sumEnergy   = 0
             for i in range(size_col):
                 for j in range(size_row):
                     label = list_label[i][j]
                     min = distance(list_centroid[label, :], data[i][j])
                     sumEnergy += min

                     for m in range(k):
                         if m == label:
                             continue

                         checkDistance = distance(list_centroid[m, :], data[i][j])
                         if(min > checkDistance):
                             list_label[i][j] = m
                             min = checkDistance
                             checkUpdate += 1

             Energy.append(np.sqrt(sumEnergy))
             if(checkUpdate == 0):
                 break


             list_centroid = np.zeros((k, size_rgb), dtype=float)
             list_count = np.zeros(k)

             count = 0
             for i in range(size_col):
```

```
            for j in range(size_row):
                label = list_label[i][j]
                list_centroid[label, :] += data[i][j]
                list_count[label]       += 1
                count                   += 1
        for i in range(0, k):
            list_centroid[i, :] /= list_count[i]


    #
    # plot image
    #
    f1 = plt.figure(1)

    x = np.arange(0, len(Energy), 1)
    plt.plot(x, Energy)
    plt.xlabel('Iteration')
    plt.ylabel('Energy')
    plt.show()

    count = 0
    for i in range(size_col):
        for j in range(size_row):
            label      = list_label[i][j]
            modifiedData[i][j] = list_centroid[label, :]
            count     += 1

    plt.imshow(modifiedData)
    plt.title('Modified image')
    plt.show()
```

```
C:\Users\ParkJinHyuk\Anaconda3\lib\site-packages\ipykernel_launcher.py:53: RuntimeWarning: inva
```


Energy
45000
40000
35000
30000
25000
20000
15000
10000
0
10
20
30
40
50
60
Iteration


Modified image
A TWOSOME PLACE
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600