機械二 陳志臻 專題第一次作業報告

Code:

```
# import packages
import cv2 as cv2
import numpy as np

#read the image I.shape -> 1024 X 1024 X 3
I = cv2.imread('bird.jpg')

#show the original image
# cv2.imshow("bird",I) #the first param is the name of the windows,and the second one is t
he input matrix
# cv2.waitKey() #the window will display infinitely until any keypress
pos_x = np.zeros((1024,1024))
for i in pos_x:
    for j in range(1024):
        i[j] = j/5
pos_y = np.transpose(pos_x)
pos_x = np.expand_dims(pos_x,axis=2)
pos_y = np.expand_dims(pos_y,axis=2)

# print(pos_y)
I1 = np.concatenate((I,pos_x,pos_y),axis=2)
print(I1.shape,pos_x.shape,pos_y.shape)
#since the image is a 3-D matrix,we have to first reshape it into a 2-D matrix
I2 = I1.reshape(-1,5)
# The data type of the image is uint8.We have to convert the unit8 values to float as it i
s a requirement of the k-means method of OpenCV.
I2 = np.float32(I2)
print(I2.shape)
#converge criterion: If the max iteration or the desired accuracy is reached the algorithm
 will terminate
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER,50,1.0)
#Opencv provides two center initialization method:
    #cv2.KMEANS_RANDOM_CENTERS:random centers
    #cv2.KMEANS_PP_CENTERS:kmeans++
flags = cv2.KMEANS_PP_CENTERS
#parameters:
    #data(I1):the data for clustering
    #K(32):number of the clusters
```

```
    #bestLabels(None):Input/output integer array that stores the cluster indices for every
 sample
    #criteria(criteria):The algorithm termination criteria, that is, the maximum number of
 iterations and/or the desired accuracy.
    #attempts(10):the number of times the algorithm is executed using different initial la
bellings.The algorithm returns the labels that yield the best compactness
    #flags(flags):center initialization method
#output:
    #retval(ret):the compactness measure(the detail is show below)
    #bestLabels(label):the matrix indicating the label of each sample(the shape is same as
 the input matrix)
    #centers(center):the value of the centers
ret,label,center = cv2.kmeans(I2,32,None,criteria,10,flags)
# convert the data type back to uint8
center = np.uint8(center)
# access the labels to regenerate the clustered image
res = center[label.flatten()]
# convert the output back to the original shape
I3 = res.reshape((I1.shape))
I3 = I3[:,:,0:3]
print(I3.shape)
#show the resultW
cv2.imshow("segmentation",I3)
cv2.waitKey()
#you can also use following code to save the result image
cv2.imwrite("result.jpg",I3)
```

結論:在加入位置參數後，可以看到顏色變成一塊一塊的分布，鄰近區域會有變成同一顏色的跡象但位置參數的設定值不能取太大，否則會變成純粹的色塊。

效果比較(左列為不含位置參數)