山东大学 计算机科学与技术 学院

机器学习（双语） 课程实验报告

学号：	姓名：	班级：
实验题目：Experiment 6: K-Means		
实验学时：4	实验日期： 2022/11/30	

实验目的：

1. 实现实验指导书中 K-Means 的相关内容；
2. 学习使用 MATLAB、Python 等工具进行实验。

硬件环境：

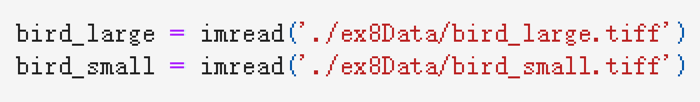
Inter（R） Core（TM） i7-8750H
RAM：16.0 GB

软件环境：

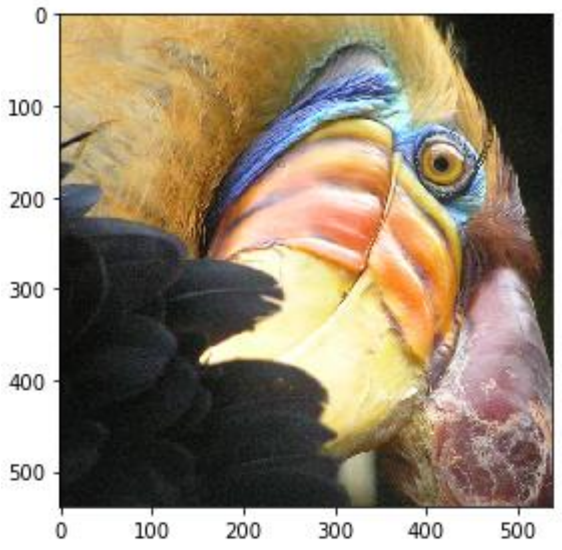
Visual Studio Code
版本: 1.67.2 (user setup)
OS: Windows_NT x64 10.0.19044
Python 3.9.7
    numpy 1.20.3
    matplotlib 3.4.3

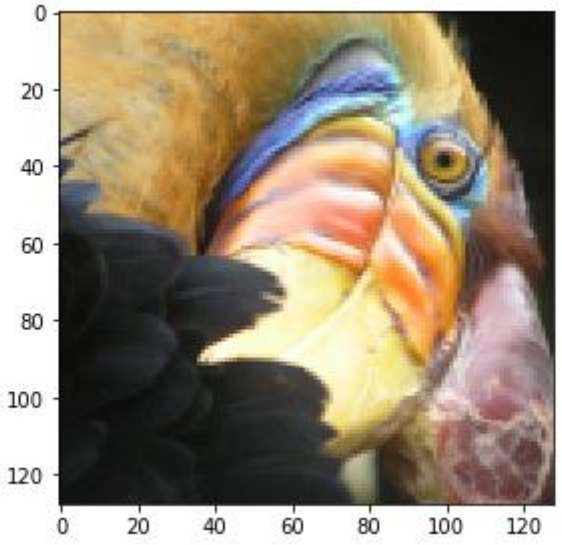
实验步骤与内容：

1. 首先使用 skimage.io.imread 方法读取数据集，并使用 imshow 方法显示图片：

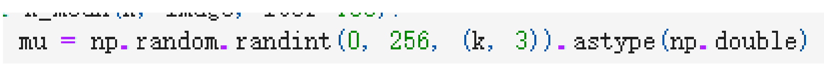


```
bird_large = imread('./ex8Data/bird_large.tiff')
bird_small = imread('./ex8Data/bird_small.tiff')
```

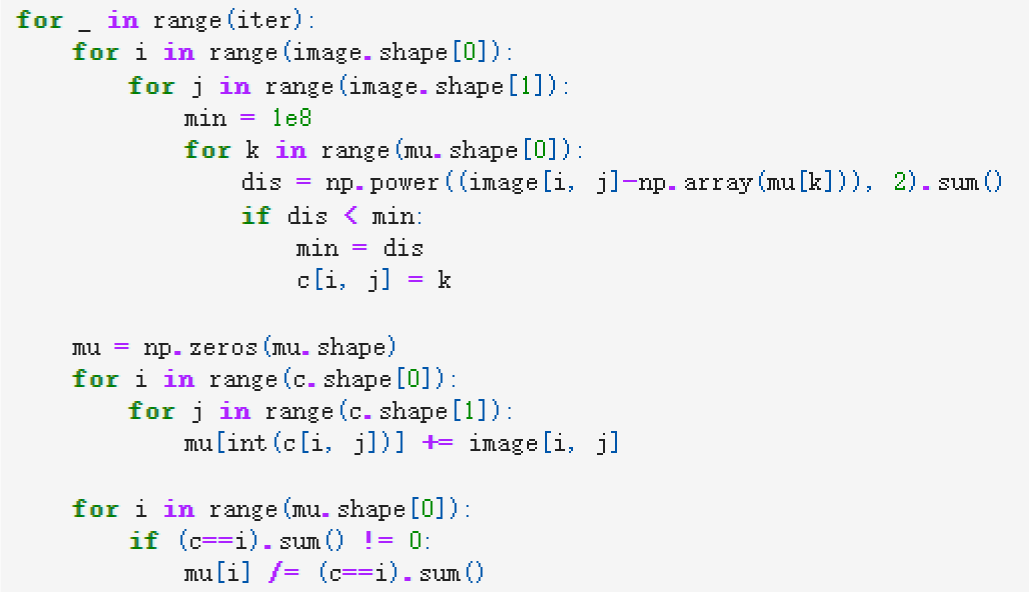




2. 对于第二幅像素较低的图片，通过实现并使用 K-Means 算法对其中的像素进行提取。对于要提取 k 个均值的程序，首先随机选取 k 个像素的 RGB 值：

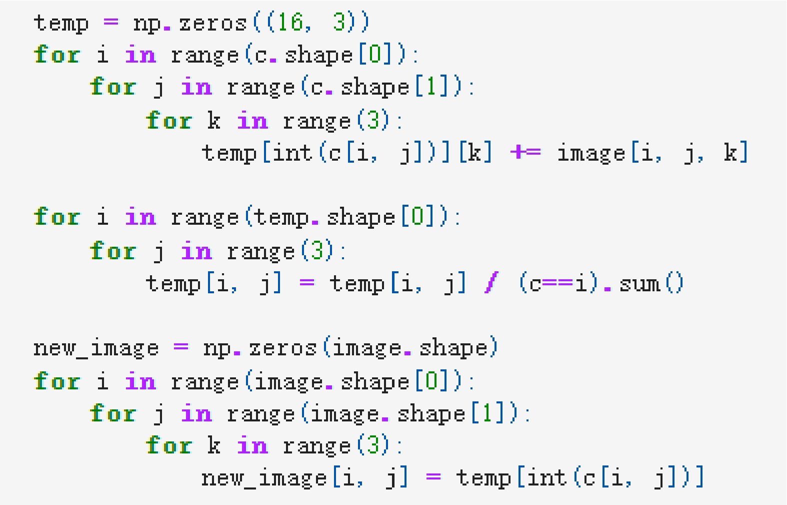


```
k_mean(k, image, iter=100):
mu = np.random.randint(0, 256, (k, 3)).astype(np.double)
```



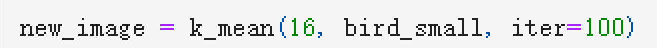
```
for _ in range(iter):
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            min = 1e8
            for k in range(mu.shape[0]):
                dis = np.power((image[i, j]-np.array(mu[k])), 2).sum()
                if dis < min:
                    min = dis
                    c[i, j] = k

    mu = np.zeros(mu.shape)
    for i in range(c.shape[0]):
        for j in range(c.shape[1]):
            mu[int(c[i, j])] += image[i, j]

    for i in range(mu.shape[0]):
        if (c==i).sum() != 0:
            mu[i] /= (c==i).sum()
```

上述代码为 K-Means 的主代码，首先遍历所有像素点，找到距离之前选择的最近的μ值，接着对于每组像素点求均值，并更新相应的μ值。
最后，在每组像素点的对应位置上填充对应的μ值即可：

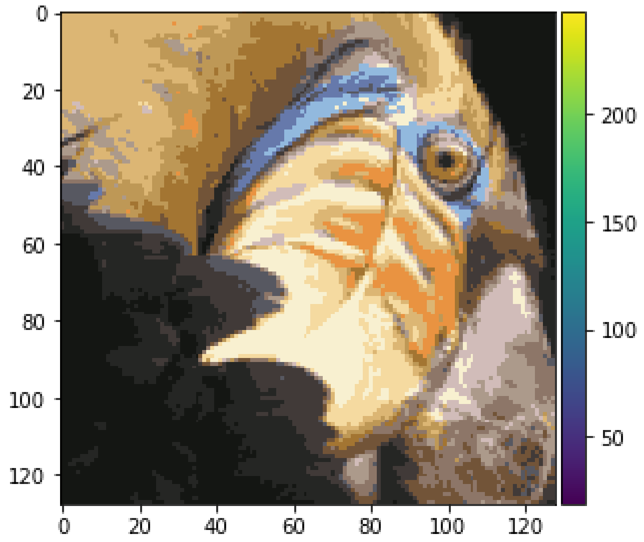


```
temp = np.zeros((16, 3))
for i in range(c.shape[0]):
    for j in range(c.shape[1]):
        for k in range(3):
            temp[int(c[i, j])][k] += image[i, j, k]

for i in range(temp.shape[0]):
    for j in range(3):
        temp[i, j] = temp[i, j] / (c==i).sum()

new_image = np.zeros(image.shape)
for i in range(image.shape[0]):
    for j in range(image.shape[1]):
        for k in range(3):
            new_image[i, j] = temp[int(c[i, j])]
```

3. 显示运行 K-means 后的图片：



```
new_image = k_mean(16, bird_small, iter=100)
```



可见，实现的 K-means 算法成功实现实验要求。

结论分析与体会：

1. 在实验前，需要充分理解使用 matlab、python 等工具，才能更好地进行实验，实现实验中的各个步骤。
2. 在实验中，需要理解掌握 K-Means 的实现原理，掌握其深层含义，结合实验指导书，才能更好地完成实验；
3. 通过使用 skimage 等第三方库，可以便捷地实现图片读取、显示等问题。
4. K-Means 原理较为简单，通过编写程序便能很快地实现并达到实验要求。

附录：程序源代码

```
# %%
import numpy as np
import matplotlib.pyplot as plt
from skimage.io import imread, imshow, imsave

# %%
bird_large = imread('./ex8Data/bird_large.tiff')
bird_small = imread('./ex8Data/bird_small.tiff')

# %%
imshow(bird_large)

# %%
imshow(bird_small)

# %%
bird_large = bird_large.astype(np.double)
```

```python
bird_small = bird_small.astype(np.double)

# %%
bird_small.shape

# %% [markdown]
# Go through each pixel in the small image and calculate its nearest mean.
# $$
# c^{(i)}:=arg\mathop{min}\limits_{j}||x^{(i)}-\mu_j||^2
# $$
# Update the values of the means based on the pixels assigned to them.
# $$
# \mu_j:=\frac{\sum^m_i1\{c^{(i)}=j\}x^{(i)}}{\sum^m_i1\{c^{(i)}=j\}}
# $$

# %%
def k_mean(k, image, iter=100):
    mu = np.random.randint(0, 256, (k, 3)).astype(np.double)
    c = np.zeros((image.shape[0], image.shape[1]))

    for _ in range(iter):
        for i in range(image.shape[0]):
            for j in range(image.shape[1]):
                min = 1e8
                for k in range(mu.shape[0]):
                    dis = np.power((image[i, j]-np.array(mu[k])), 2).sum()
                    if dis < min:
                        min = dis
                        c[i, j] = k

        mu = np.zeros(mu.shape)
        for i in range(c.shape[0]):
            for j in range(c.shape[1]):
                mu[int(c[i, j])] += image[i, j]

        for i in range(mu.shape[0]):
            if (c==i).sum() != 0:
                mu[i] /= (c==i).sum()

    temp = np.zeros((16, 3))
    for i in range(c.shape[0]):
        for j in range(c.shape[1]):
            for k in range(3):
                temp[int(c[i, j])][k] += image[i, j, k]

    for i in range(temp.shape[0]):
        for j in range(3):
            temp[i, j] = temp[i, j] / (c==i).sum()

    new_image = np.zeros(image.shape)
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            for k in range(3):
                new_image[i, j] = temp[int(c[i, j])]

    imshow(new_image.astype(int))

    return new_image

# %%
new_image = k_mean(16, bird_small, iter=100)

# %%
imsave('./new_image.tiff', new_image.astype(int))
```