

# 图像聚类

```
In [1]: import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
```

```
In [2]: from sklearn.cluster import KMeans
```

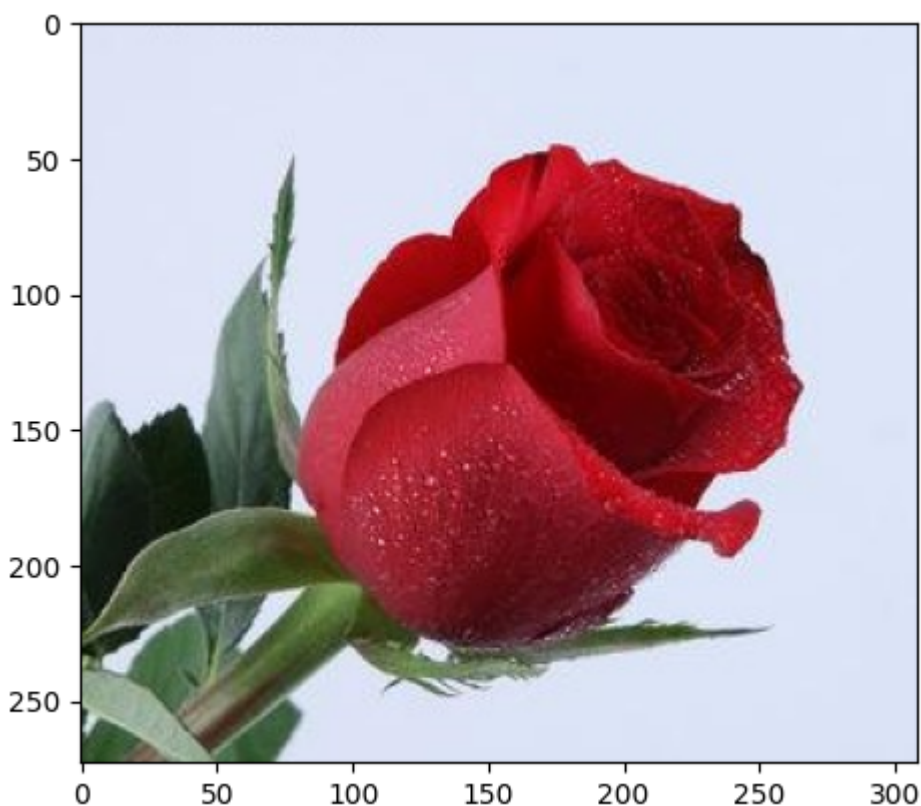
## 方法一：使用matplotlib库

```
In [3]: from matplotlib import image as mpimg
```

```
In [4]: # 图像路径：工作路径+文件名
img_path = os.path.join(os.getcwd(), 'red_rose.jpg')
```

```
In [5]: # 将图像读取为numpy数组
img = mpimg.imread(img_path)
```

```
In [6]: plt.imshow(img)
plt.show()
```



```
In [7]: # 检查图像大小
img.shape
```

```
Out[7]: (273, 309, 3)
```

```
In [8]: # 注意原图是 uint8 类型的数组
img.dtype
```

```
Out[8]: dtype('uint8')
```

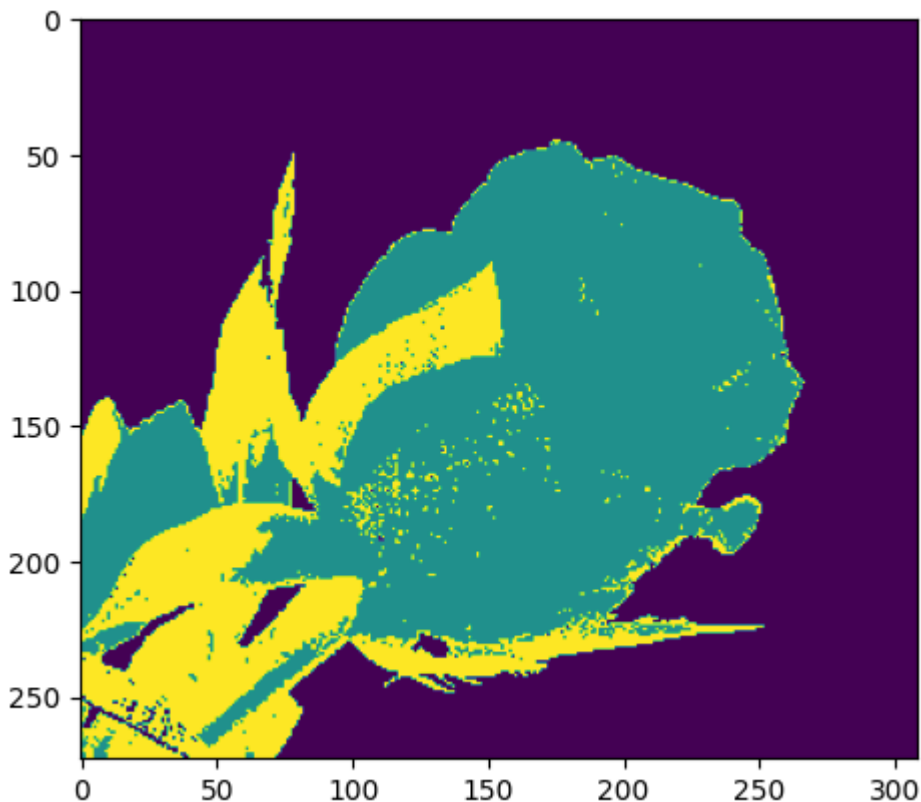
```
In [9]: # 将图像的像素点展平
imgData = img.reshape((-1,3))
```

```
In [10]: imgData.shape
```

```
Out[10]: (84357, 3)
```

```
In [11]: # 聚类
km = KMeans(n_clusters=3)
labels = km.fit_predict(imgData)
```

```
In [12]: # 可视化
img_class = labels.reshape(img.shape[:2]) #img.shape[:2]->(273, 309)
plt.imshow(img_class)
plt.show()
```



## 方法二：使用PIL库

```
In [13]: from PIL import Image
#PIL库是一个具有强大图像处理能力的第三方库，不仅包含了丰富的像素、色彩操作功能，还
```

```
In [14]: # 图像路径：工作路径+文件名
img_path = os.path.join(os.getcwd(), 'R-C.jpg')
```

```
In [15]: # 将图像读取为 Image 对象
```

```
img = Image.open(img_path)
```

In [16]: `img`

Out[16]:



```
In [17]: # 将 Image 对象转为 numpy 数组
img_np = np.array(img)
```

```
In [18]: # 检查图像大小
img_np.shape
```

Out[18]: (1024, 1024, 3)

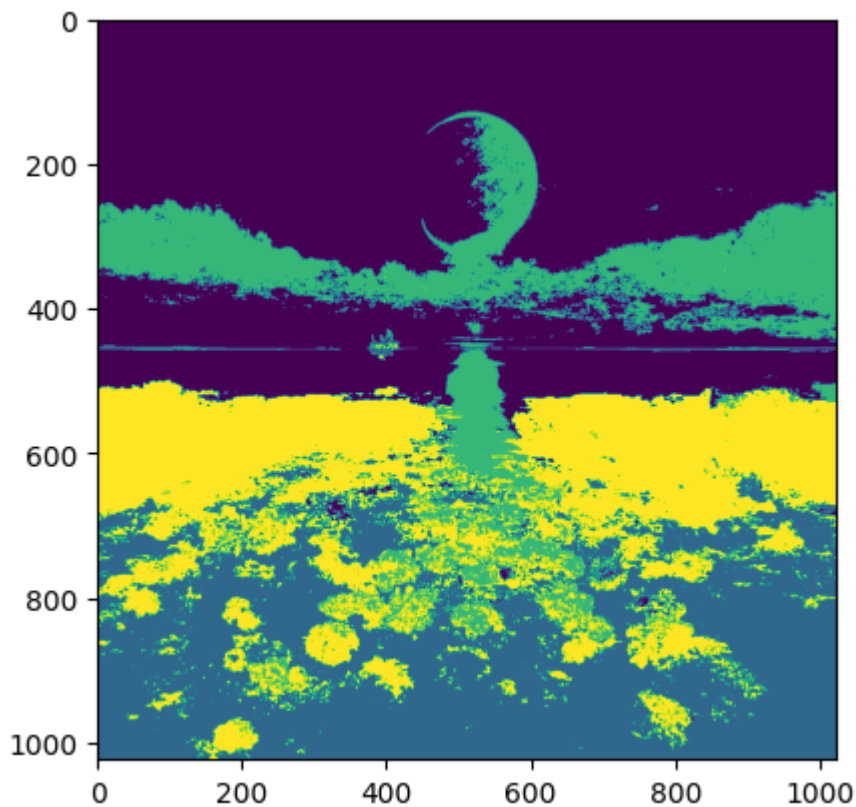
```
In [19]: # uint8 类型的数组
img_np.dtype
```

Out[19]: dtype('uint8')

```
In [20]: # 将图像的像素点展平
imgData = img_np.reshape((-1,3))
```

```
In [21]: # 聚类
km = KMeans(n_clusters=4)
labels = km.fit_predict(imgData)
```

```
In [22]: # 可视化
img_class = labels.reshape(img_np.shape[:2]) #img.shape[:2]->(1024,1024)
plt.imshow(img_class)
plt.show()
```

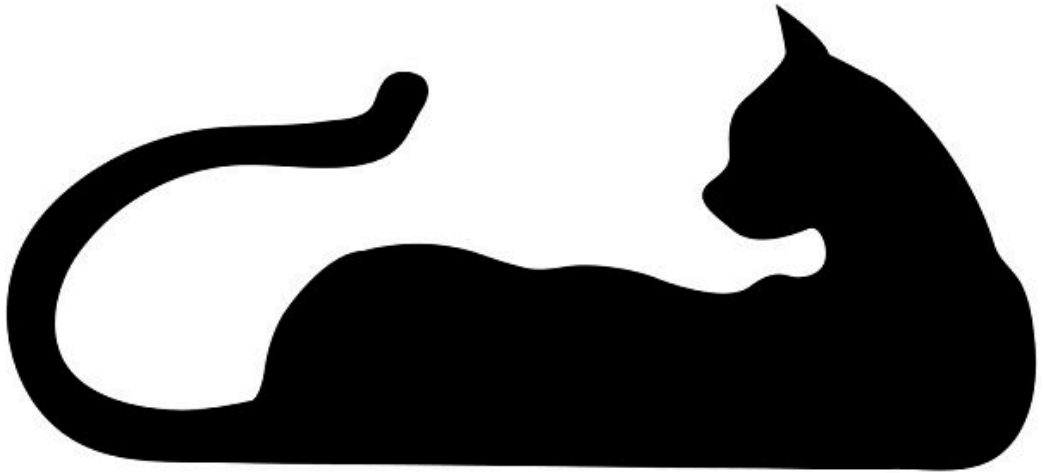


## 补充：仅对单通道聚类

```
In [23]: # 图像路径：工作路径+文件名
img_path = os.path.join(os.getcwd(), 'black_cat.jpg')
```

```
In [24]: # 将图像读取为 Image 对象
img = Image.open(img_path)
img
```

Out[24]:



```
In [26]: img_np = np.array(img)/255 # c w h 归一化
w,h,c = img_np.shape
```

```
In [27]: img_np[:, :, 0]
```

```
Out[27]: array([[1., 1., 1., ..., 1., 1., 1.],
                [1., 1., 1., ..., 1., 1., 1.],
                [1., 1., 1., ..., 1., 1., 1.],
                ...,
                [1., 1., 1., ..., 1., 1., 1.],
                [1., 1., 1., ..., 1., 1., 1.],
                [1., 1., 1., ..., 1., 1., 1.]])
```

```
In [28]: # 调用自己实现的KMeans
img_np_1, _ = Kmeans(img_np[:, :, 0], 2) # 仅对第一个通道进行聚类
```

相比之前的聚类模式速度更快  
原先的聚类问题需要 5min, 这种只需要 50s

```
In [29]: from skimage.transform import resize
img_np_1 = resize(img_np_1, (w,h))
img_np = (img_np*255).astype(np.uint8)
```

```
img_c = Image.fromarray(img_np)  
img_c
```

Out[29]:

