该系列文章是讲解Python OpenCV图像处理知识，前期主要讲解图像入门、OpenCV基础用法，中期讲解图像处理的各种算法，包括图像锐化算子、图像增强技术、图像分割等，后期结合深度学习研究图像识别、图像分类应用。希望文章对您有所帮助，如果有不足之处，还请海涵~

该系列在github所有源代码：https://github.com/eastmountyxz/ImageProcessing-Python
PS：请求帮忙点个Star，哈哈，第一次使用Github，以后会分享更多代码，一起加油。

同时推荐作者的C++图像系列知识：
[数字图像处理] 一.MFC详解显示BMP格式图片
[数字图像处理] 二.MFC单文档分割窗口显示图片
[数字图像处理] 三.MFC实现图像灰度、采样和量化功能详解
[数字图像处理] 四.MFC对话框绘制灰度直方图
[数字图像处理] 五.MFC图像点运算之灰度线性变化、灰度非线性变化、阈值化和均衡化处理详解
[数字图像处理] 六.MFC空间几何变换之图像平移、镜像、旋转、缩放详解
[数字图像处理] 七.MFC图像增强之图像普通平滑、高斯平滑、Laplacian、Sobel、Prewitt锐化详解

前文参考：
[Python图像处理] 一.图像处理基础知识及OpenCV入门函数
[Python图像处理] 二.OpenCV+Numpy库读取与修改像素
[Python图像处理] 三.获取图像属性、兴趣ROI区域及通道处理
[Python图像处理] 四.图像平滑之均值滤波、方框滤波、高斯滤波及中值滤波
[Python图像处理] 五.图像融合、加法运算及图像类型转换
[Python图像处理] 六.图像缩放、图像旋转、图像翻转与图像平移
[Python图像处理] 七.图像阈值化处理及算法对比
[Python图像处理] 八.图像腐蚀与图像膨胀
[Python图像处理] 九.形态学之图像开运算、闭运算、梯度运算
[Python图像处理] 十.形态学之图像顶帽运算和黑帽运算
[Python图像处理] 十一.灰度直方图概念及OpenCV绘制直方图
[Python图像处理] 十二.图像几何变换之图像仿射变换、图像透视变换和图像校正
[Python图像处理] 十三.基于灰度三维图的图像顶帽运算和黑帽运算
[Python图像处理] 十四.基于OpenCV和像素处理的图像灰度化处理
[Python图像处理] 十五.图像的灰度线性变换
[Python图像处理] 十六.图像的灰度非线性变换之对数变换、伽马变换
[Python图像处理] 十七.图像锐化与边缘检测之Roberts算子、Prewitt算子、Sobel算子和Laplacian算子
[Python图像处理] 十八.图像锐化与边缘检测之Scharr算子、Canny算子和LOG算子

前面的文章讲解了图像锐化和边缘提取技术，该篇文章将开始围绕图像分割进行讲解。百度百科将其定义为：

> 图像分割就是把图像分成若干个特定的、具有独特性质的区域并提出感兴趣目标的技术和过程。它是由图像处理到图像分析的关键步骤。现有的图像分割方法主要分以下几类：基于阈值的分割方法、基于区域的分割方法、基于边缘的分割方法以及基于特定理论的分割方法等。从数学角度来看，图像分割是将数字图像划分成互不相交的区域的过程。图像分割的过程也是一个标记过程，即把属于同一区域的像素赋予相同的编号。

本篇文章主要讲解基于理论的图像分割方法，通过K-Means聚类算法实现图像分割或颜色分层处理。基础性文章，希望对你有所帮助。

1.K-Means原理

2.K-Means聚类分割灰度图像

3.K-Means聚类对比分割彩色图像

注意：该部分知识均为自己查阅资料撰写，转载请署名CSDN+杨秀璋及原地址出处，谢谢！！

PS：文章参考自己以前系列图像处理文章及OpenCV库函数，同时参考如下文献：
杨秀璋等. 基于苗族服饰的图像锐化和边缘提取技术研究[J]. 现代计算机，2018(10).
《数字图像处理》（第3版），冈萨雷斯著，阮秋琦译，电子工业出版社，2013年.
《数字图像处理学》（第3版），阮秋琦，电子工业出版社，2008年，北京.
《OpenCV3编程入门》，毛星云，冷雪飞，电子工业出版社，2015.

一.K-Means聚类原理

第一部分知识主要参考自己的新书《Python网络数据爬取及分析从入门到精通（分析篇）》和之前的博客 [Python数据挖掘课程] 二.Kmeans聚类数据分析。

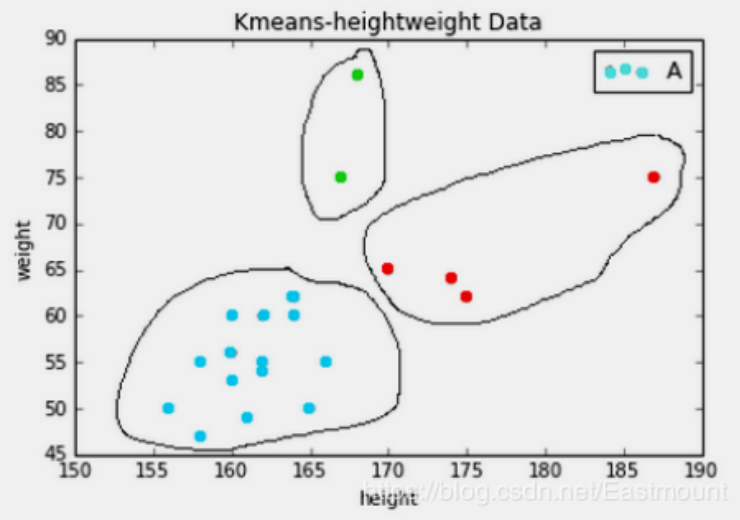
K-Means聚类是最常用的聚类算法，最初起源于信号处理，其目标是将数据点划分为K个类簇，找到每个簇的中心并使其度量最小化。该算法的最大优点是简单、便于理解，运算速度较快，缺点是只能应用于连续型数据，并且要在聚类前指定聚集的类簇数。

下面是K-Means聚类算法的分析流程，步骤如下：

- 第一步，确定K值，即将数据集聚集成K个类簇或小组。
- 第二步，从数据集中随机选择K个数据点作为质心（Centroid）或数据中心。
- 第三步，分别计算每个点到每个质心之间的距离，并将每个点划分到离最近质心的小组，跟定了那个质心。

- 第四步，当每个质心都聚集了一些点后，重新定义算法选出新的质心。
- 第五步，比较新的质心和老的质心，如果新质心和老质心之间的距离小于某一个阈值，则表示重新计算的质心位置变化不大，收敛稳定，则认为聚类已经达到了期望的结果，算法终止。
- 第六步，如果新的质心和老的质心变化很大，即距离大于阈值，则继续迭代执行第三步到第五步，直到算法终止。

下图是对身高和体重进行聚类的算法，将数据集的人群聚集成三类。



二.K-Means聚类分割灰度图像

在图像处理中，通过K-Means聚类算法可以实现图像分割、图像聚类、图像识别等操作，本小节主要用来进行图像颜色分割。假设存在一张100×100像素的灰度图像，它由10000个RGB灰度级组成，我们通过K-Means可以将这些像素点聚类成K个簇，然后使用每个簇内的质心点来替换簇内所有的像素点，这样就能实现在不改变分辨率的情况下量化压缩图像颜色，实现图像颜色层级分割。

在OpenCV中，Kmeans()函数原型如下所示：

retval, bestLabels, centers = kmeans(data, K, bestLabels, criteria, attempts, flags[, centers])

- data表示聚类数据，最好是np.flloat32类型的N维点集
- K表示聚类类簇数
- bestLabels表示输出的整数数组，用于存储每个样本的聚类标签索引

- criteria表示算法终止条件，即最大迭代次数或所需精度。在某些迭代中，一旦每个簇中心的移动小于criteria.epsilon，算法就会停止
- attempts表示重复试验kmeans算法的次数，算法返回产生最佳紧凑性的标签
- flags表示初始中心的选择，两种方法是cv2.KMEANS_PP_CENTERS ;和 cv2.KMEANS_RANDOM_CENTERS
- centers表示集群中心的输出矩阵，每个集群中心为一行数据

下面使用该方法对灰度图像颜色进行分割处理，需要注意，在进行K-Means聚类操作之前，需要将RGB像素点转换为一维的数组，再将各形式的颜色聚集在一起，形成最终的颜色分割。

```
# coding: utf-8
import cv2
import numpy as np
import matplotlib.pyplot as plt

#读取原始图像灰度颜色
img = cv2.imread('scenery.png', 0)
print img.shape

#获取图像高度、宽度
rows, cols = img.shape[:]

#图像二维像素转换为一维
data = img.reshape((rows * cols, 1))
data = np.float32(data)

#定义中心 (type,max_iter,epsilon)
criteria = (cv2.TERM_CRITERIA_EPS +
            cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)

#设置标签
flags = cv2.KMEANS_RANDOM_CENTERS

#K-Means聚类 聚集成4类
compactness, labels, centers = cv2.kmeans(data, 4, None, criteria, 10, f

#生成最终图像
dst = labels.reshape((img.shape[0], img.shape[1]))

#用来正常显示中文标签
plt.rcParams['font.sans-serif']=['SimHei']

#显示图像
```

```
titles = [u'原始图像', u'聚类图像']
images = [img, dst]
for i in xrange(2):
   plt.subplot(1,2,i+1), plt.imshow(images[i], 'gray'),
   plt.title(titles[i])
   plt.xticks([]),plt.yticks([])
plt.show()
```

输出结果如图所示，左边为灰度图像，右边为K-Means聚类后的图像，它将灰度级聚集成四个层级，相似的颜色或区域聚集在一起。

三.K-Means聚类对比分割彩色图像

下面代码是对彩色图像进行颜色分割处理，它将彩色图像聚集成2类、4类和64类。

```
# coding: utf-8
import cv2
import numpy as np
import matplotlib.pyplot as plt

#读取原始图像
img = cv2.imread('scenery.png')
print img.shape

#图像二维像素转换为一维
data = img.reshape((-1,3))
data = np.float32(data)

#定义中心 (type,max_iter,epsilon)
criteria = (cv2.TERM_CRITERIA_EPS +
            cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)

#设置标签
flags = cv2.KMEANS_RANDOM_CENTERS
```

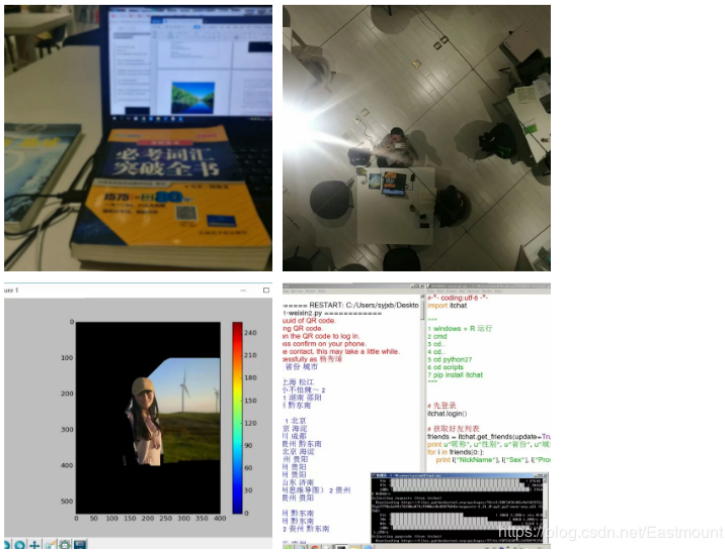
```
#K-Means聚类 聚集成2类
compactness, labels2, centers2 = cv2.kmeans(data, 2, None, criteria, 10,

#K-Means聚类 聚集成4类
compactness, labels4, centers4 = cv2.kmeans(data, 4, None, criteria, 10,

#K-Means聚类 聚集成8类
compactness, labels8, centers8 = cv2.kmeans(data, 8, None, criteria, 10,

#K-Means聚类 聚集成16类
compactness, labels16, centers16 = cv2.kmeans(data, 16, None, criteria, 1

#K-Means聚类 聚集成64类
compactness, labels64, centers64 = cv2.kmeans(data, 64, None, criteria, 1

#图像转换回uint8二维类型
centers2 = np.uint8(centers2)
res = centers2[labels2.flatten()]
dst2 = res.reshape((img.shape))

centers4 = np.uint8(centers4)
res = centers4[labels4.flatten()]
dst4 = res.reshape((img.shape))

centers8 = np.uint8(centers8)
res = centers8[labels8.flatten()]
dst8 = res.reshape((img.shape))

centers16 = np.uint8(centers16)
res = centers16[labels16.flatten()]
dst16 = res.reshape((img.shape))

centers64 = np.uint8(centers64)
res = centers64[labels64.flatten()]
dst64 = res.reshape((img.shape))

#图像转换为RGB显示
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
dst2 = cv2.cvtColor(dst2, cv2.COLOR_BGR2RGB)
dst4 = cv2.cvtColor(dst4, cv2.COLOR_BGR2RGB)
dst8 = cv2.cvtColor(dst8, cv2.COLOR_BGR2RGB)
dst16 = cv2.cvtColor(dst16, cv2.COLOR_BGR2RGB)
dst64 = cv2.cvtColor(dst64, cv2.COLOR_BGR2RGB)

#用来正常显示中文标签
plt.rcParams['font.sans-serif']=['SimHei']
```

```
#显示图像
titles = [u'原始图像', u'聚类图像 K=2', u'聚类图像 K=4',
          u'聚类图像 K=8', u'聚类图像 K=16',  u'聚类图像 K=64']
images = [img, dst2, dst4, dst8, dst16, dst64]
for i in xrange(6):
   plt.subplot(2,3,i+1), plt.imshow(images[i], 'gray'),
   plt.title(titles[i])
   plt.xticks([]),plt.yticks([])
plt.show()
```

输出结果如下图所示，当K=2颜色聚集成两种，当K=64颜色聚集成64种。

希望这篇基础性文章对您有所帮助，如果有错误 或不足之处，请海涵！一起加油，考博加油。

Eastmountain 书山有路勤为径，学海无涯苦作舟。下班从花溪回来，书店中继续奋战，几场考试即将来临，加油共勉。下午上课分享了itchat微信操作。



（By：Eastmount 2019-04-11 夜8点写于贵阳·钟书阁 https://blog.csdn.net/Eastmount/）