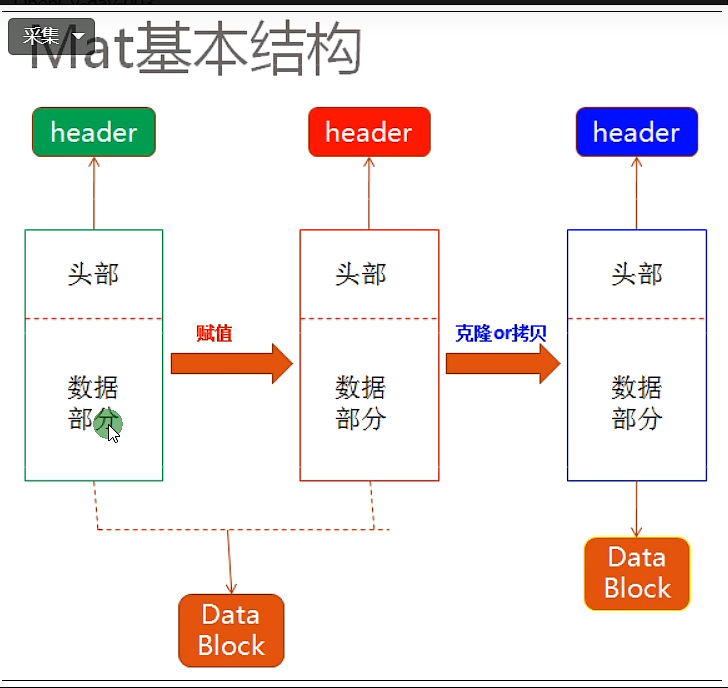
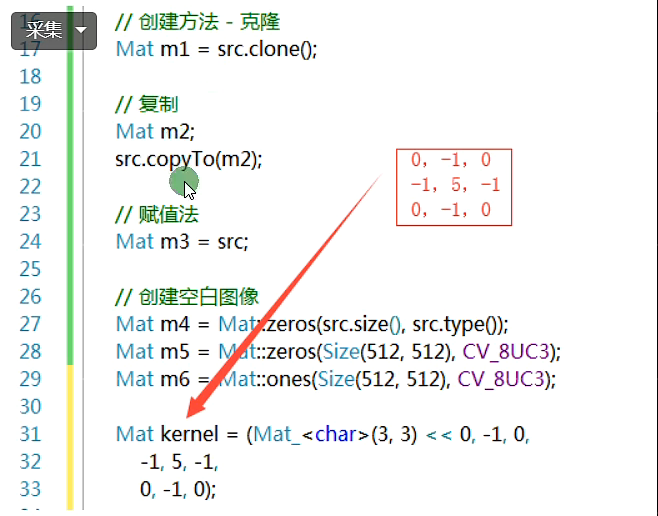


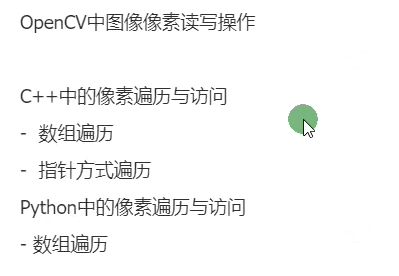
图像对象的创建与赋值

1.怎么操作 它 2.怎么访问每一个像素点 3.怎么创建一个空图或Mat

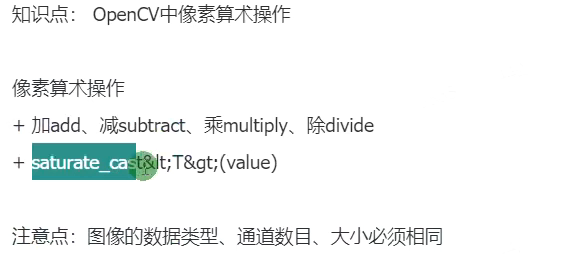




004 - 图像像素的读写操作



005 - 图像像素的算术操作

、

007 - 滚动条操作演示-参数传递度

void cvAddWeighted( const CvArr\* src1, double alpha,const CvArr\* src2, double beta,double gamma, CvArr\* dst );

参数1：src1，第一个原数组.

参数2：alpha，第一个数组元素权重

参数3：src2第二个原数组

参数4：beta，第二个数组元素权重

参数5：gamma，图1与图2作和后添加的数值。不要太大，不然图片一片白。总和等于255以上就是纯白色了。

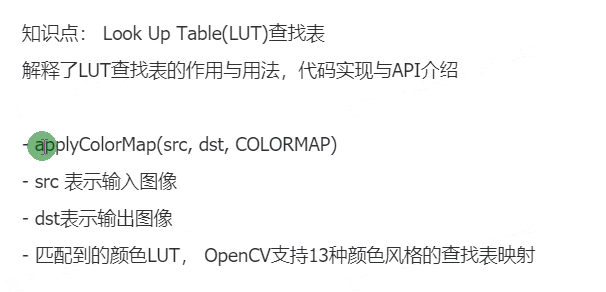
参数6：dst，输出图片

008 - 键盘响应操作

//waitKey()--这个函数是在一个给定的时间内(单位ms)等待用户按键触发;如果用户没有按下 键,则接续等待(循环)

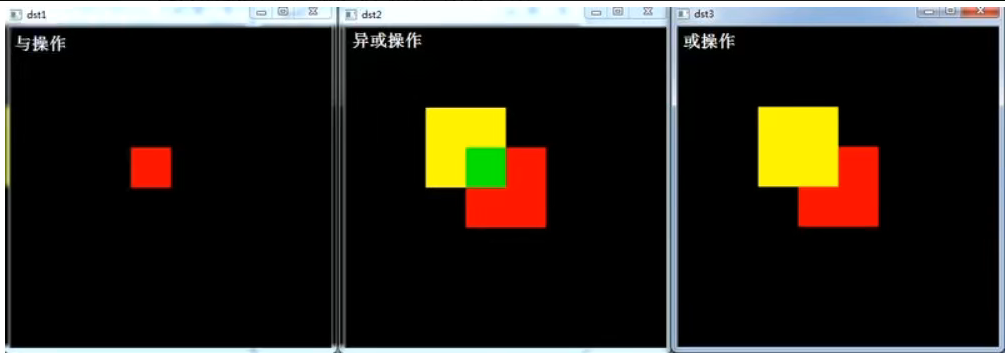
通过按键输入调整图片的模式

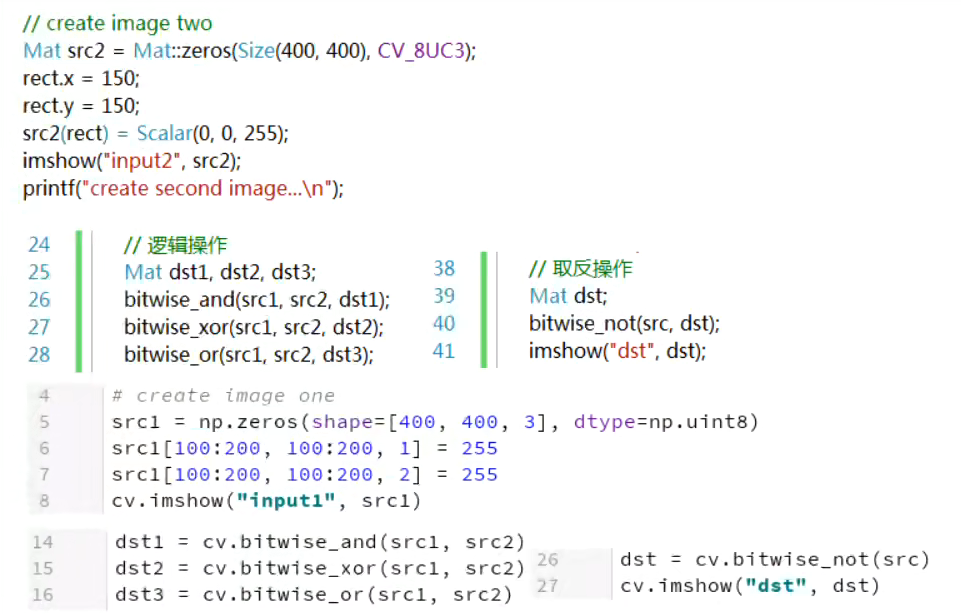
009 - OpenCV自带颜色表操作



010图像像素的逻辑操作

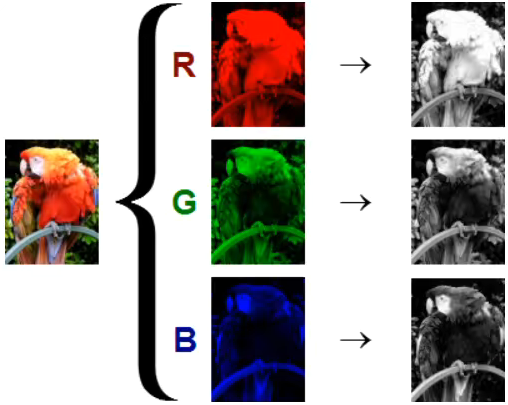


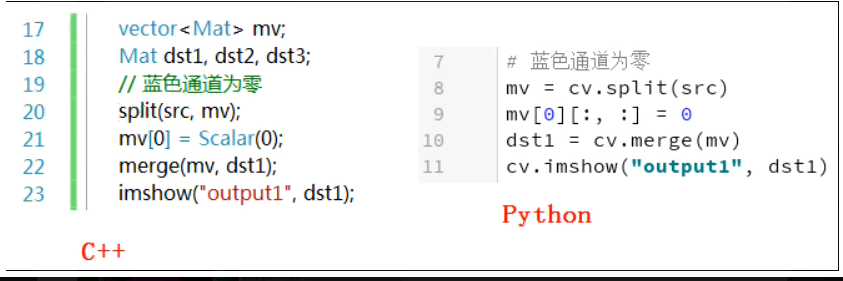




Rect(左上角横坐标，左上角纵坐标，宽度，高度)

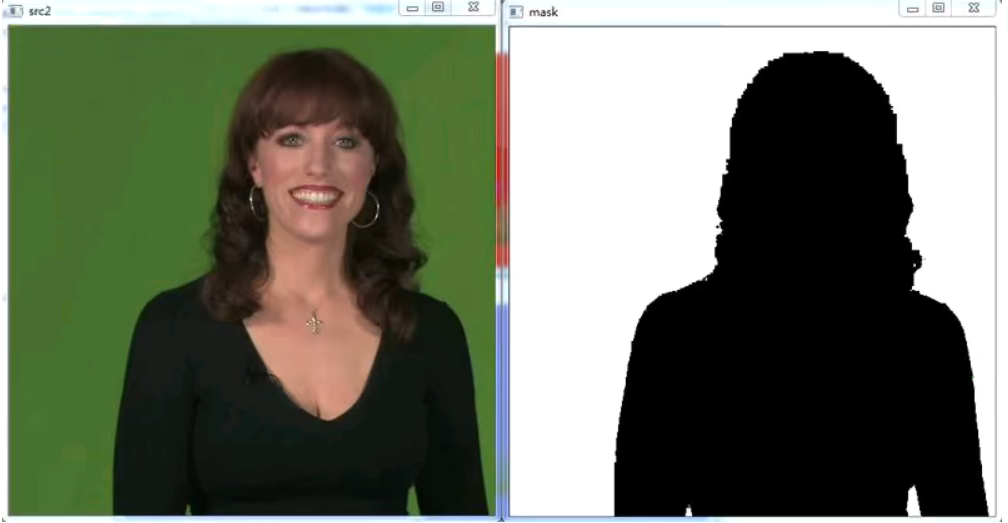
011 - 通道分离与合并

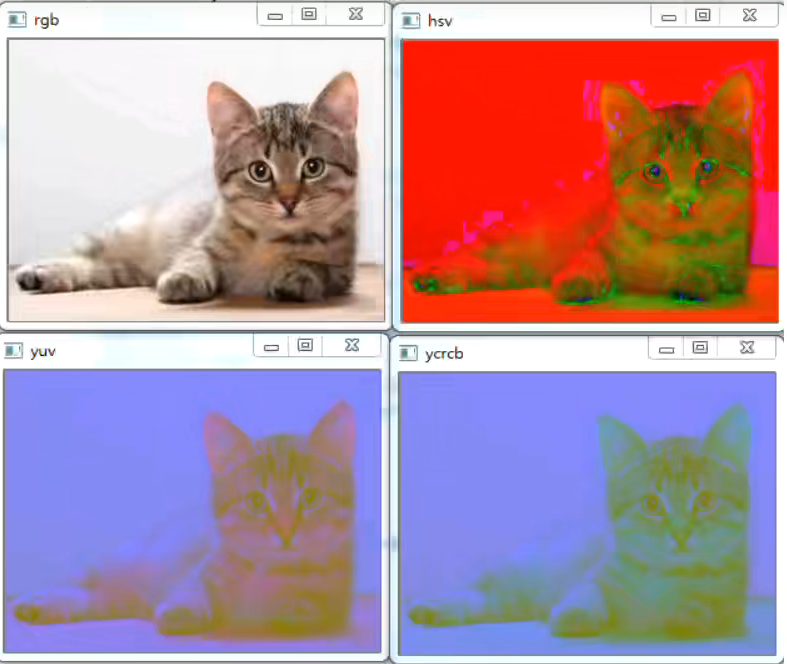


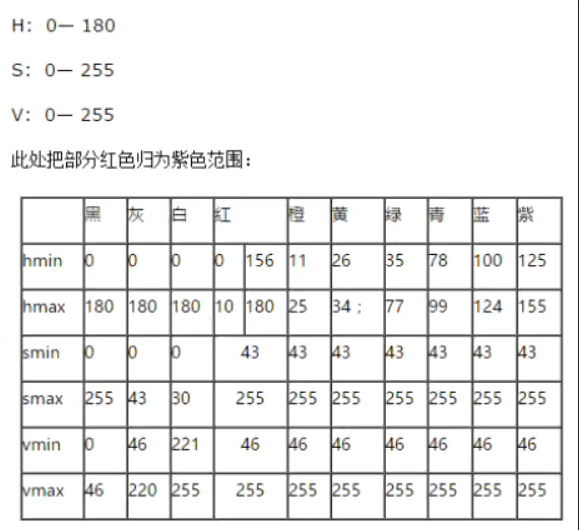


12 - 图像色彩空间转换

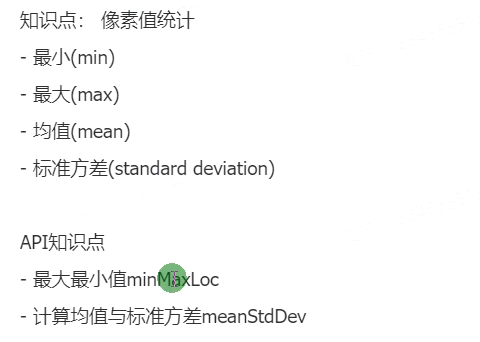


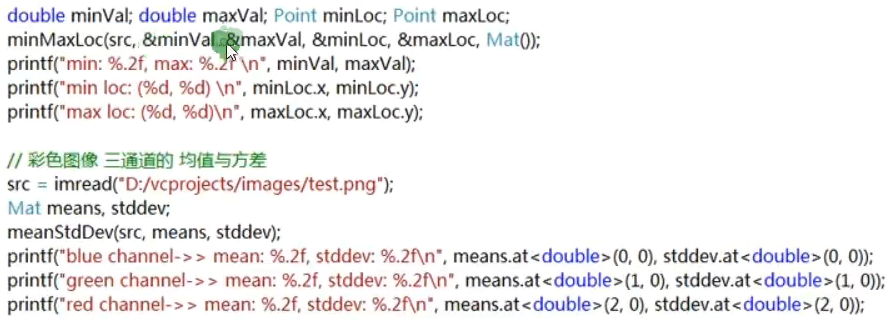






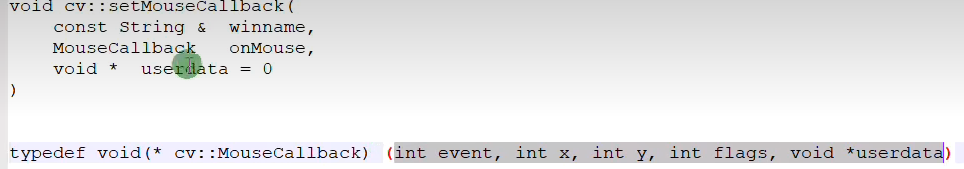
013 - 图像像素值统计



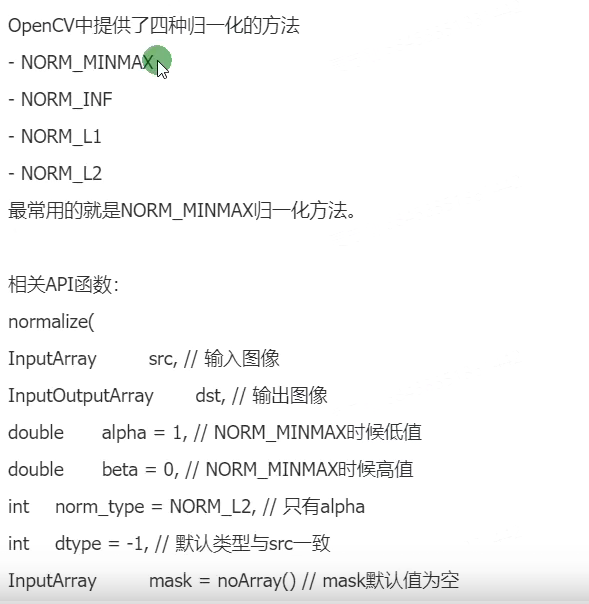


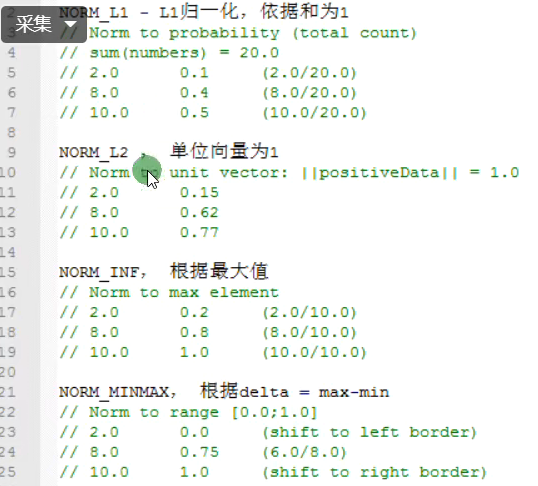
014 绘制图形

017 - 鼠标操作与响应



018 - 图像像素类型转换与归一化





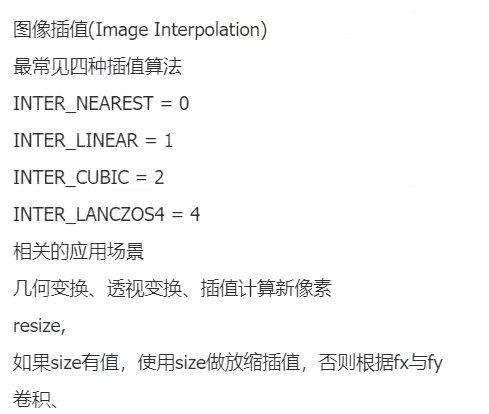
归一化是指把矩阵里面的数据同时控制到一个范围里面有利于提高计算精度

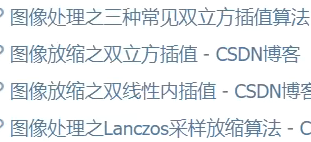
L1：除以和，归一化后和为1；L2：除以模，归一化后模为1；INF：除以最大值，归一化后最大值为1；MINMAX:除以最大最小差值，归一化后最大最小差值为1

.NORM\_L1、NORM\_INF、NORM\_L2模式下归一化结果与beta无关，只与alpha有关，详见第4部分的公式说明；

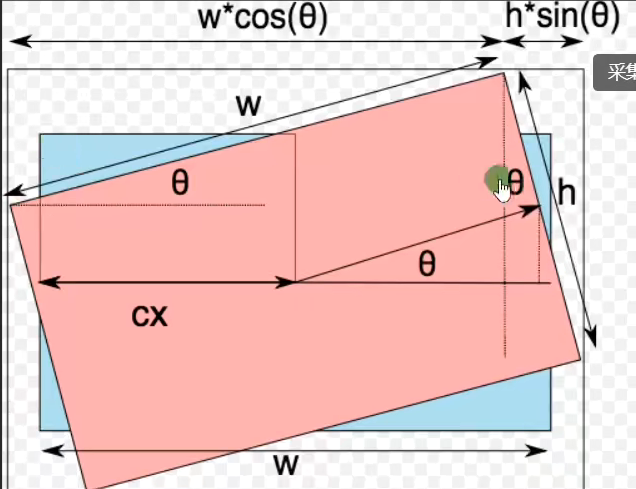
2.NORM\_MINMAX中alpha、beta都起作用，同时需要注意的是alpha和beta的取值顺序与归一化结果无关。即alpha=255,beta=0和alpha=0,beta=255最后的归一化结果是相同的。

019 -图像收缩与插值



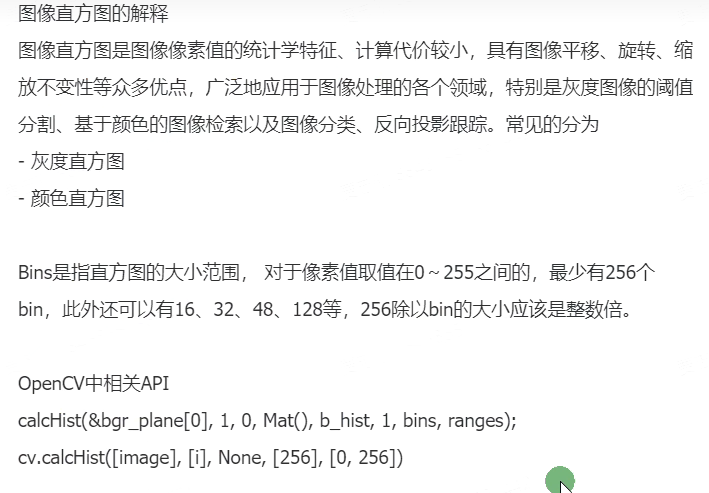


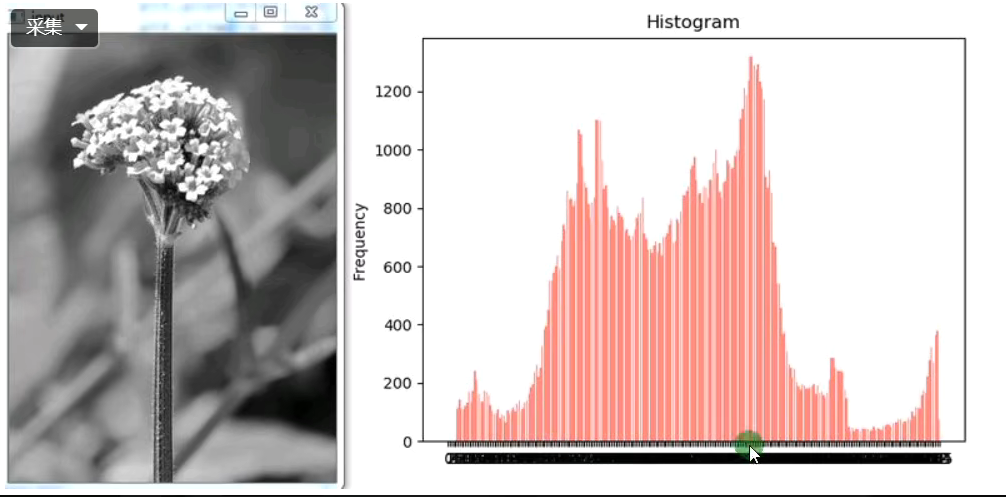
20 图像旋转

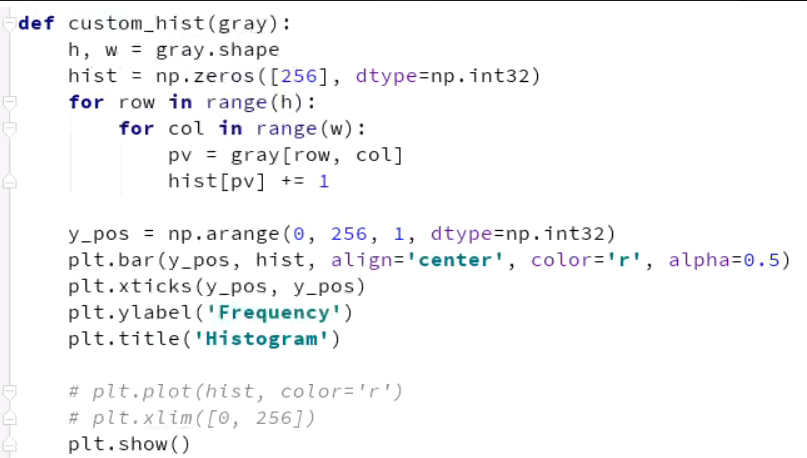


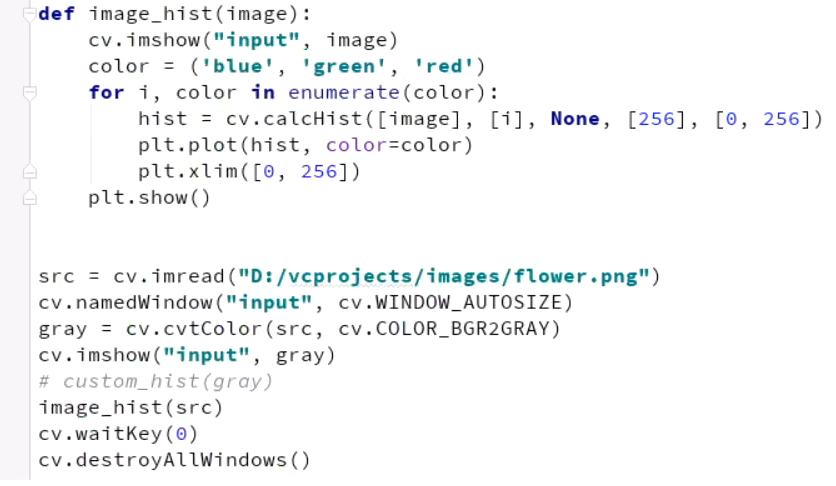
//（0，0）是仿射矩阵M的第一行第一个元素就是cos，（0，1）就是第一行第二个元素就是sin

024 图像的直方图

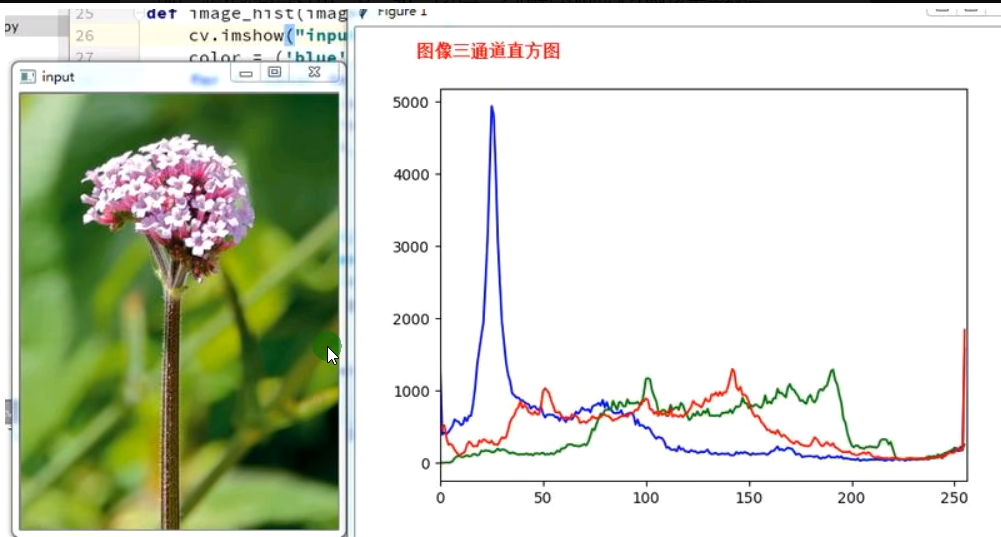


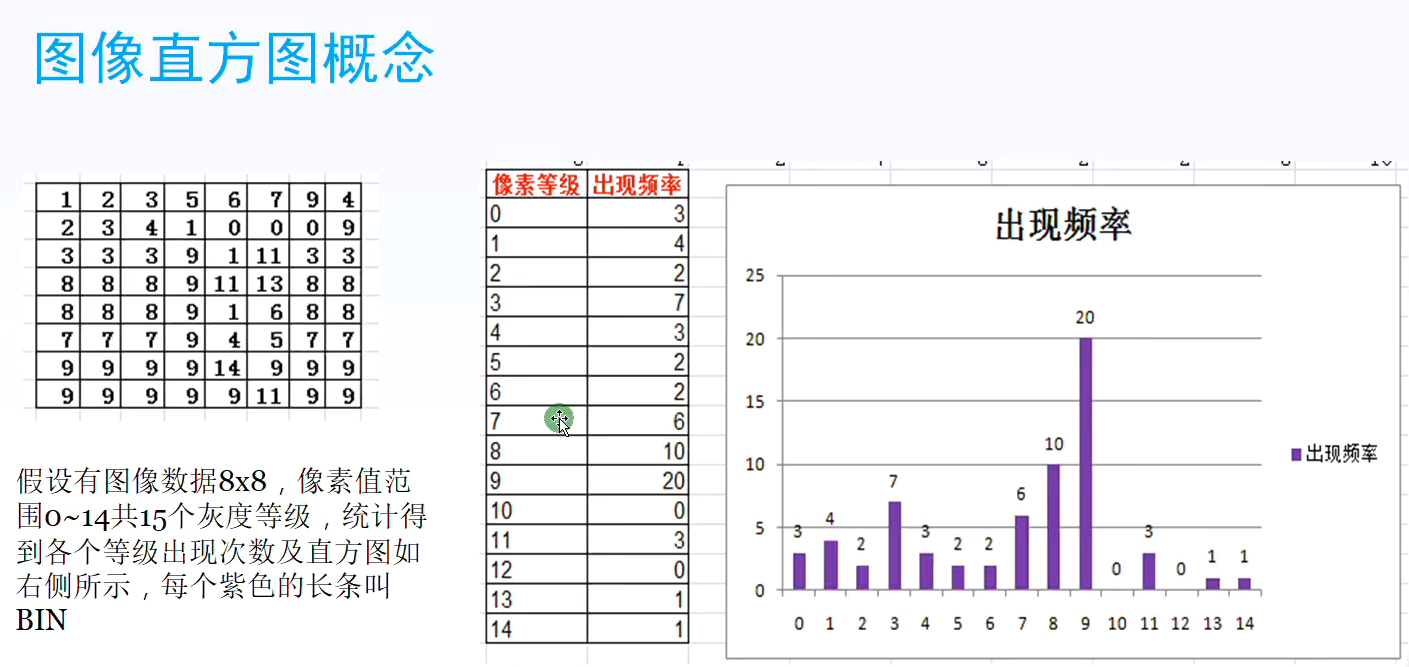






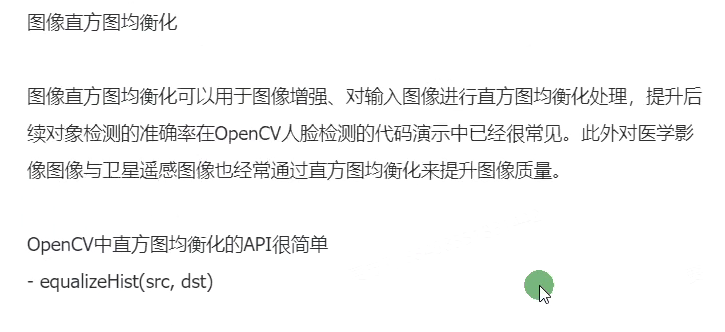


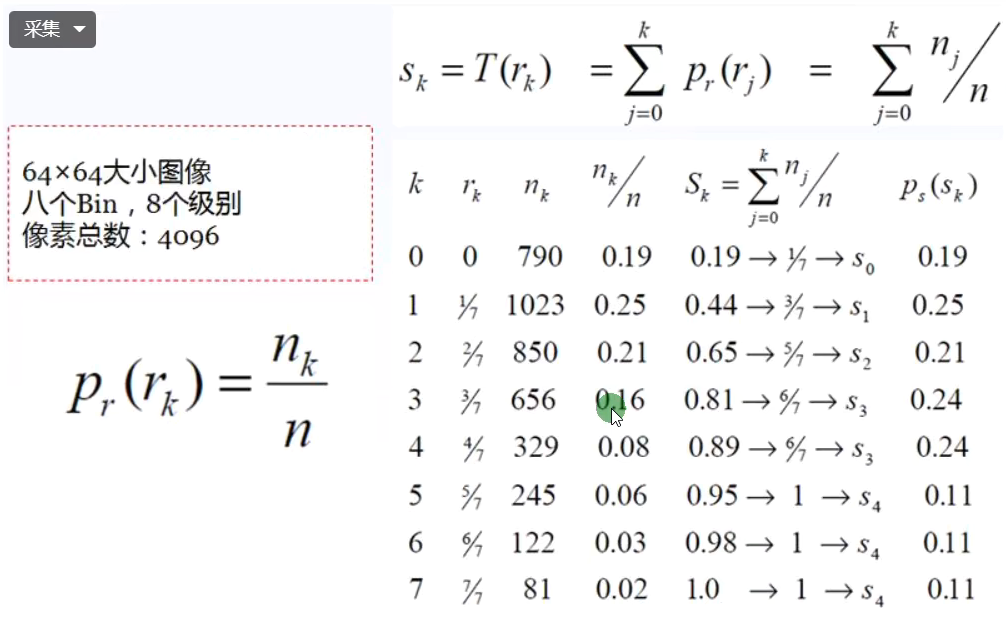




Bin是灰度等级

026 直方图均衡化

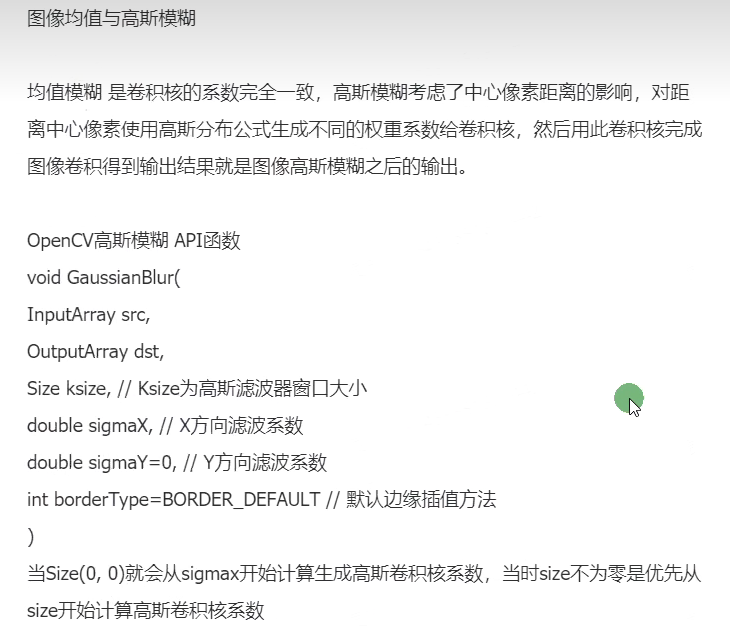


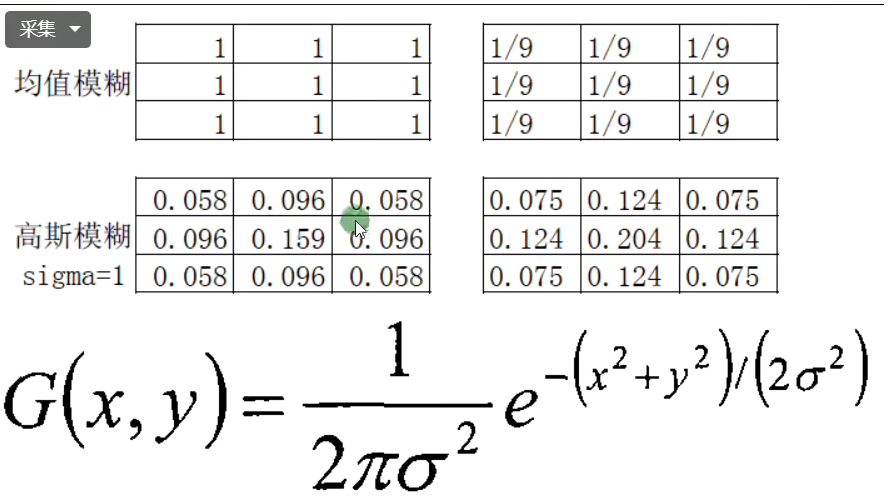


rk是步长均匀增加，nk是步长内像素总数

第二项是以步长为单位的四舍五入最小为步长，

028 高斯模糊





029 高斯双边模糊

