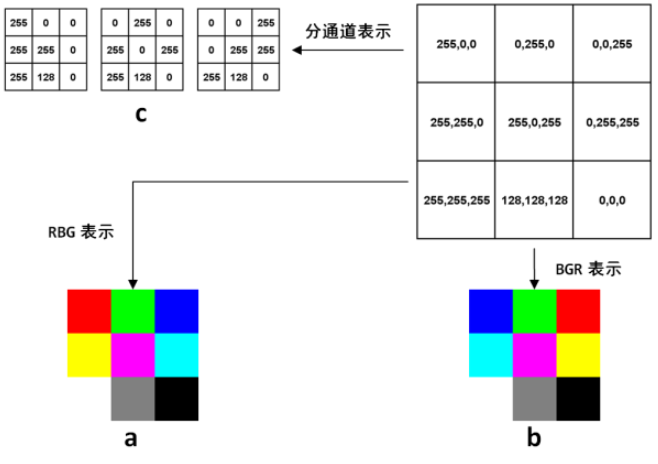
RGB就是指Red,Green和Blue,一副图像由这三个channel(通道)构成



其中c分别为每个像素中RGB的颜色值；右上角的矩阵则表示每个像素的像素值

Gray就是只有灰度值一个channel

HSV即Hue(色调),Saturation(饱和度)和Value(亮度)三个channel

RGB是为了让机器更好的显示图像,对于人类来说并不直观,HSV更为贴近我们的认知,所以通常我们在针对某种颜色做提取时会转换到HSV颜色空间里面来处理.

HSV = cv2.cvtColor(Img, cv2.COLOR\_BGR2HSV)

H, S, V = cv2.split(HSV)

openCV的基本使用:

加载图像:cv2.LoadImage或imread

显示图像:cv2.ShowImage或imshow

存储图像:cv2.SaveImage或imwrite

获取属性:

img.shape #(长x、宽y、通道z)

img.size #图像大小

img.dtype #图像数据类型

在图像上显示文字:

cv.PutText(image,"Hello World !", (x,y),font, cv.RGB(255, 255, 255))

给图像添加边框:

cv2.copyMakeBorder(img, 50, 50, 0, 0, cv2.BORDER\_CONSTANT, value=(0, 0, 0))

图像剪裁:

Img[20:150, -180:-50]

缩放变换：

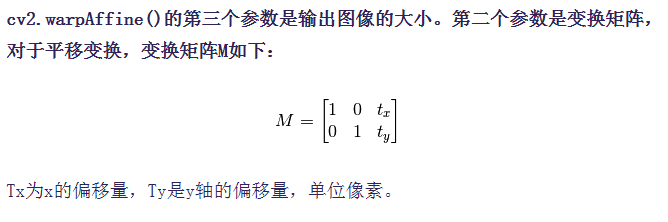
缩放就是改变图像的大小，使用cv2.resize()函数。图像的大小可以手动指定，也可以使用缩放比例。cv2.resize()支持多种插值算法，默认使用的是cv2.INTER\_LINEAR（不管放大和缩小）。缩小最适合使用：cv2.INTER\_AREA，放大最适合使用：cv2.INTER\_CUBIC (慢) 或 cv2.INTER\_LINEAR。

平移变换

Rows,cols,channel = img.shape

M = numpy.float([[1, 0, 100], [0, 1, 50]])

cv2.warpAffine(img, M, (cols, rows))



旋转变化：

在opencv中提供了cv2.getRotationMatrix2D函数获得变换矩阵。第一参数指定旋转圆点；第二个参数指定旋转角度；第二个参数指定缩放比例。

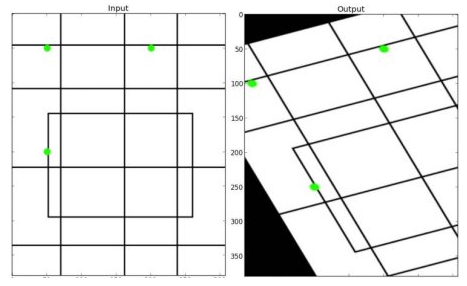
rows,cols,channel = img.shape

M = cv2.getRotationMatrix2D((cols/2, rows/3), 90, 0.4)

Dst = cv2.warpAffine(img, M, (cols, rows))

仿射变换:

仿射变换是一种二维坐标到二维坐标之间的线性变换，并保持二维图形的“平直性”。转换前平行的线，在转换后依然平行。如下图:



rows,cols,channel = img.shape

#仿射前的3个坐标点坐标

pts1 = numpy.float32([[50, 50], [200, 50], [50, 200]])

#仿射后的3个坐标点坐标

pts2 = numpy.float32([[10, 100], [200, 50], [100, 250]])

#从仿射点1到放射点2的变换

M = cv2.getAffineTrasfrom(pts1, pts2)

#扭曲仿射

Dst = cv2.warpAffine(img, M,((cols, rows))

透视变换：

透视变换需要3×3的变换矩阵，直线在变换后还是保持直线。为了构造变换矩阵，你需要输入图像的4个点和对应的要输出图像的4个点；要求这4个点其中3个点不共线。使用cv2.getPerspectiveTransform函数构造透视变换矩阵。代码：

rows,cols,ch = img.shape

#透视变换前的4个点的坐标

pts1 = np.float32([[56,65],[368,52],[28,387],[389,390]])

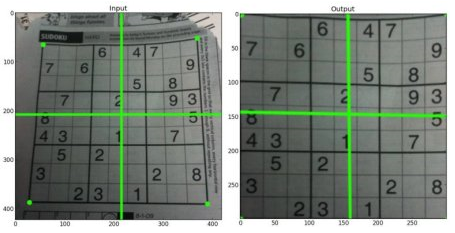
#透视变换后的4个点的坐标

pts2 = np.float32([[0,0],[300,0],[0,300],[300,300]])

#透视变化

M = cv2.getPerspectiveTransform(pts1,pts2)

dst = cv2.warpPerspective(img,M,(300,300))



灰度处理：

由彩色变为灰色，灰度图像与黑白图像不同，在计算机图像领域中黑白图像只有黑白两种颜色，灰度图像在黑色与白色之间还有许多级的颜色深度。

在RGB模型中，如果R=G=B时，则彩色表示一种灰度颜色，其中R=G=B的值叫灰度值。

grey = cv.CreateImage(size, depth, channels)

cv.CvtColor(image, grey, cv.CV\_BGR2GRAY)

光滑去噪：

使图片上剧烈变化的地方过度平滑，即让图像上不同颜色之间的分界线模糊

smooth = cv.CreateImage(size, depth, channels)

cv.Smooth(grey, smooth, cv.CV\_MEDIAN)

均衡化操作：

统计一个图像的像素值，绘制成一个直方图，然后均衡化（均衡化即让直方图分布更接近正态分布）

equalize = cv.CreateImage(size, depth, channels)

cv.Smooth(grey, equalize)

拉普拉斯操作：

把图像中的轮廓突出出来

Laplace = cv.CreateImage(size, depth, channels) cv.CvtColor(image, Laplace, cv.CV\_BGR2GRAY)

dst\_16s2 = cv.CreateImage(size, cv.IPL\_DEPTH\_16S, 1)

cv.Laplace(Laplace, dst\_16s2)

cv.Convert(dst\_16s2, Laplace)