

OpenCV-python 学习笔记 OpenCV几何变换

1. 扩展缩放

`cv2.resize()` 只改变图像的尺寸大小。

缩放时: `cv2.INTER_AREA`

拓展时: `cv2.INTER_CUBIC` (较慢) 和 `cv2.INTER_LINEAR`

默认所有改变图像尺寸大小的操作使用的是插值法都是 `cv2.INTER_LINEAR`

例如:

```
1 import cv2
2
3 img = cv2.imread('1.jpg')
4 # 下面的None本应该是输出图像的尺寸，但是因为后面我们设置了缩放因子，所以，这里为None
5 res1 = cv2.resize(img, None, fx=2, fy=2,
6                   interpolation=cv2.INTER_CUBIC)
7 # or
8 # 这里直接设置输出图像的尺寸，所以不用设置缩放因子
9 height, width = img.shape[:2]
10 res2 = cv2.resize(img, (2 * width, 2 * height),
11                   interpolation=cv2.INTER_CUBIC)
12
13 cv2.imshow('img', img)
14 cv2.imshow('res', res1)
15 cv2.waitKey(0)
```

2. 平移

使用 `cv2.warpAffine()`

图像的平移分为两步，第一步时定义好图像的平移矩阵，分别指定 x 方向和 y 方向上的平移量 t_x 和 t_y ，平移矩阵形如：

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

平移矩阵可以利用 `np.float32()` 来定义，然后将平移矩阵传入

`cv2.warpAffine()` 的第二个参数，例如：

```

1 img = cv2.imread('messi5.jpg',0)
2 rows,cols = img.shape
3 # 平移矩阵M: [[1,0,x],[0,1,y]]
4 M = np.float32([[1,0,100],[0,1,50]])
5 dst = cv2.warpAffine(img,M,(cols,rows))

```



其中，函数 `cv2.warpAffine()` 的第三个参数的是输出图像的大小，它的格式应该是图像的（宽，高）。应该记住的是图像的宽对应的是列数，高对应的是行数。

3. 旋转

使用 `cv2.getRotationMatrix2D` 构建旋转矩阵，再使用 `cv2.warpAffine()` 进行变换。

`getRotationMatrix2D`:

- center-表示旋转的中心点
- angle-表示旋转的角度degrees
- scale-图像缩放因子

`warpAffine`:

- src - 输入的图像
- M - 2 X 3 的变换矩阵.
- dsize - 输出的图像的size大小
- dst - 输出的图像
- flags - 输出图像的插值方法
- borderMode - 图像边界的处理方式
- borderValue - 当图像边界处理方式为~BORDER_CONSTANT ~时的填充值

例如:

```

1 img = cv2.imread('messi5.jpg',0)
2 rows,cols = img.shape
3 #90度旋转
4 M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
5 dst = cv2.warpAffine(img,M,(cols,rows))

```



4. 仿射变换

在仿射变换中，原图中所有平行线在结果图像中同样平行。

为创建这个矩阵，需要从原图像中找到三个点以及他们在输出图像中的位置，然后 `cv2.getAffineTransform()` 会创建一个2X3的矩阵。最后这个矩阵会被传给函数 `cv2.warpAffine()`。

例如：

```
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 img=cv2.imread('draw.png')
6 rows,cols,ch = img.shape
7
8 pts1 = np.float32([50,50],[200,50],[50,200])
9 pts2 = np.float32([10,100],[200,50],[100,250])
10 #行，列，通道数
11 M = cv2.getAffineTransform(pts1,pts2)
12
13 dst = cv2.warpAffine(img,M,(cols,rows))
14
15 plt.subplot(121,plt.imshow(img),plt.title('Input'))
16 plt.subplot(121,plt.imshow(dst),plt.title('output'))
17 plt.show()
```

5. 透视变换

对于视角变换，我们需要一个3x3变换矩阵。在变换前后直线还是直线。

需要在原图上找到4个点，以及他们在输出图上对应的位置，这四个点中任意三个都不能共线，可以有函数 `cv2.getPerspectiveTransform()` 构建，然后这个矩阵传给函数 `cv2.warpPerspective()`

例如：

```
1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 img=cv2.imread('sudokusmall.png')
6 rows,cols,ch=img.shape
7
8 pts1 = np.float32([[56,65],[368,52],[28,387],[389,390]])
9 pts2 = np.float32([[0,0],[300,0],[0,300],[300,300]])
10
11 M=cv2.getPerspectiveTransform(pts1,pts2)
12
13 dst=cv2.warpPerspective(img,M,(300,300))
14
15 plt.subplot(121,plt.imshow(img),plt.title('Input'))
16 plt.subplot(121,plt.imshow(img),plt.title('Output'))
17 plt.show()
```