

OpenCV 函数简介

郑华

2015 年 11 月 12 日

1 读取图片

1.imread `Mat img=imread("D:\\ 美丽姐 \\Ant\\image-0000001.bmp");`

2.cvLoadImage 从文件中读取图像 `const char *pstrImageName = "D:\\ 美丽姐 \\Ant\\image-0000001.bmp";`
`IplImage *pImage = cvLoadImage(pstrImageName, CV_LOAD_IMAGE_UNCHANGED);`

2 显示窗口

1.imshow `cvNamedWindow(" 游戏原画");`
`// 在窗口中显示游戏原画 imshow(" 游戏原画",img);`

2.cvShowImage `cvNamedWindow(pstrWindowsTitle, CV_WINDOW_AUTOSIZE);`
`//在指定窗口中显示图像 cvShowImage(pstrWindowsTitle, pImage);`

3 基本数据结构

1.cvPoint 系列 表示图像中的点, `cvPoint`: `cvPoint2D32f`: 二维空间中的点 `cvPoint3D32f`: 三维空间中的点



图 1: 坐标图

2.cvSize 系列 表示图像的尺寸, CvSize2D32f 如果想用浮点型;
一个宽度, 一个高度。

3.cvRect 矩形框 x,y,width,height



图 2: 坐标图

4.cvScalar 包含 4 个 double 型数据, 可以用来表示 B 通道,G 通道,R 通道,Alpha 透明度.

构造函数 3 个:

cvScalar,cvRealScalar,cvScalarAll

```
CvScalar s = cvScalar(double B_Channle, double G_Channle, double R_Channle, double Alpha);
```

```
// Example:
```

```
CvScalar s = cvScalar(20.0);
```

```
s.val[0]=20.0;
```

5.cvRectangle cvRectangle(
myImg,

cvPoint(5,20), 利用构造函数构造一个实例, 而免去声明的过程

cvPoint(20,30),

cvScalar(255,255,255)

); 在 (5,10) 和 (20,30) 之间画一个白色矩形

6.cvArr

7.cvMat 矩阵维度和通道 1. 创建矩阵并初始化

```
cvMat mat;
```

```
float data[18] = {30,60,40,60,50,40, 67,88,55,33,22,97, 59,69,32,46,25,45}; 三行六列
```

```
cvInitMatHeader(&mat, 3, 6,CV_32FC1,data);
```

```
cvInitMatHeader(矩阵变量, 行数, 列数, 数据类型 (位数、类型、通道数), 数据);
```

黑白图像

	0	1	2	3	4	5
0	30	60	40	60	50	40
1	67	88	55	33	22	97
2	59	69	32	46	25	45

图 3: 单通道存储结果

```
cvInitMatHeader(&mat, 3, 6/2,CV_32FC2,data);
```

灰度图像

	0	1	2
0	30, 60	40, 60	50, 40
1	67, 88	55, 33	22, 97
2	59, 69	32, 46	25, 45

↓ 拆分

	0	1	2
0	30	40	50
1	67	55	22
2	59	32	25

	0	1	2
0	60	60	40
1	88	33	97
2	69	46	45

图 4: 双通道存储结果

```
cvInitMatHeader(&mat, 3, 6/3,CV_32FC3,data);
```

彩色图像 2. 相关函数

	0	1
0	30, 60, 40	60, 50, 40
1	67, 88, 55	33, 22, 97
2	59, 69, 32	46, 25, 45

	0	1
0	30	60
1	67	33
2	59	46

	0	1
0	60	50
1	88	22
2	69	25

	0	1
0	40	40
1	55	97
2	32	45

图 5: 三通道存储结果

格式: `double cvGetReal2D(const CvArr* arr, int idx_row, int idx_column);`

作用: 返回单通道数组的指定元素

3. 通道

通道的个数就是说一个元素上存储几个数。如 2

多通道可以分解成单通道。

4. 维度: 就是几维空间的维度, 比如 3 维空间的话就得 3 个坐标表示一个点 (x,y,z)

8.cvMat 矩阵数据访问 `mat.data.ptr + y*mat.step;`

`mat.step` 就是一行数据的字节数;

9. IplImage 结构 Width, Height 图像长宽

Depth: usually use IPL_DEPTH_*U/S/F (1、8、16、32) 每个像素占用多少个字节数表示, 如 8u 就表示可以有 $2^8 = 256$ 种颜色值

nChannels :can be valued 1,2,3,4

origin: only can be valued IPL_ORIGIN_TL or IPL_ORIGIN_BL

dataOrder: can be valued by IPL_DATA_ORDER_PIXEL or IPL_DATA_ORDER_PLANE

widthStep : 临行同列之间的字节数

4 直方图

1. 一维直方图 像素值多少在多少范围, 然后对应的峰值也会随之增加, 如图

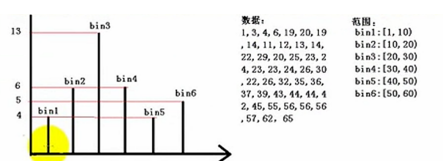


图 6: 一维单通道直方图模型