OpenCV 自2.x版本以来，处理图像的主要数据结构从CvMat变成了Mat，下面是OpenCV官方文档对Mat的描述：

[cpp] view plain copy

class CV\_EXPORTS Mat

{

public:

// ... a lot of methods ...

...

/\*! includes several bit-fields:

- the magic signature

- continuity flag

- depth

- number of channels

\*/

int flags;

//! the array dimensionality, >= 2

int dims;

//! the number of rows and columns or (-1, -1) when the array has more than 2 dimensions

int rows, cols;

//! pointer to the data

uchar\* data;

//! pointer to the reference counter;

// when array points to user-allocated data, the pointer is NULL

int\* refcount;

// other members

...

};

1：访问图像像素

之前的一篇博客讲了访问图像像素的三种方法为 at、ptr、迭代器。

现在我们尝试使用data数据成员访问图像像素

原理：图像矩阵是一个二维数组，不论是灰度图像还是彩色图像，在计算机内存中都是以一维数组的形式存储的。用Mat存储一幅图像时，若图像在内存中是连续存储的（Mat对象的isContinuous == true），则可以将图像的数据看成是一个一维数组，而其data（uchar\*）成员就是指向图像数据的第一个字节的，因此可以用data指针访问图像的数据。不要忘了进行强制类型转换。

测试程序：

int main()

{

Mat image(2, 2, CV\_32FC3);

int rowNumber = image.rows;

int colNumber = image.cols\*image.channels();

for (int i = 0; i < rowNumber; ++i)

{

for (int j = 0; j < colNumber; ++j)

{

float \*data = reinterpret\_cast<float\*>(image.data);

data[j] = 1.5f;

}

}

for (int i = 0; i < rowNumber; ++i)

{

for (int j = 0; j < colNumber; ++j)

{

float \*data = reinterpret\_cast<float\*>(image.data);

cout << data[j]<<" ";

}

}

return 0;

}

2：判断一个图像文件是否打开成功

if (!srcImage.data)

printf("无法打开图片 dota.jpg");

if (!logoImage.data)

printf("无法打开图片 dota\_logo.jpg");

//data uchar型的指针。Mat类分为了两个部分:矩阵头和指向矩阵数据部分的指针，data就是指向矩阵数据的指针

//指向矩阵数据的指针不为0，则说明成功读取图像了，否则读取错误

---------------------

作者：菜鸡丶子航

来源：CSDN

原文：https://blog.csdn.net/xv1356027897/article/details/80019815

版权声明：本文为博主原创文章，转载请附上博文链接！