

个人资料



航子_harder

访问：2078次

积分：170

等级：BLOG > 2

排名：千里之外

原创：14篇 转载：4篇

译文：0篇 评论：0条

文章搜索

异步赠书：Kotlin领衔10本好书 SDCC 2017之区块链技术实战线上峰会 程序员9月书讯 每周荐书：Java Web、Python极客编程
(评论送书)

Opencv访问图像像素

标签：图像处理

2017-07-28 18:41

44人阅读

评论

分类：图像处理 (2)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

1.图像存储方式

图像在内存中的存储方式是连续的，每列连续存储RGB三个通道，可以用isContinuous()函数判断是否连续存储。下图为灰度图和彩色图的存储结构。

关闭

文章分类

深度学习 (7)

图像处理 (3)

C++深入 (1)

机器学习 (0)

Python深入 (5)

刷题总结 (1)

文章存档

2017年09月 (1)

2017年07月 (3)

2017年06月 (1)

2017年05月 (13)

阅读排行

rcnn-fast-rcnn--faster-rcnn (211)

kaggle--mnist--knn 比赛 (199)

python-numpy小结3 (187)

spyder--no module name (183)

Fast_rcnn+Caffe配置以及 (162)

caffe环境问题以及ubuntu (144)

ubuntu14.04 + cuda 7.5 (134)

传统物体检测 (122)

Faster-Rcnn demo.py解析 (100)

Python-Numpy小结1 (97)

	Column 0	Column 1	Column ...	Column m
Row 0	0,0	0,1	...	0, m
Row 1	1,0	1,1	...	1, m
Row,0	...,1, m
Row n	n,0	n,1	n,...	n, m

	Column 0			Column 1			Column ...			Column m		
Row 0	0,0	0,0	0,0	0,1	0,1	0,1	0, m	0, m	0, m
Row 1	1,0	1,0	1,0	1,1	1,1	1,1	1, m	1, m	1, m
Row,0	...,0	...,0	...,1	...,1	...,1, m	..., m	..., m
Row n	n,0	n,0	n,0	n,1	n,1	n,1	n,...	n,...	n,...	n, m	n, m	n, m

2.访问图像像素方式

2.1 Mat 的at方法

at< Vec3b >(i,j)[k]的方式访问，其中i,j表示图像的行列，k表示通道数，RGB图像有三个通道，分别是要访问图像每个通道，可以直接访问图像像素，则可用: at< uchar >(i,j) 方式直接去除图像特定行列的像素值。

注：图像的像素点的灰度值类型必须为 uchar (unsigned char)，因为char: -128~127; uchar: 0~255。如果为char型，在不需要对图像像素进行运算的时候是没问题的，但是一旦进行像素运算就有可能出错，为了避免使用 uchar类型。

特别的：OpenCV定义了一个Mat的模板子类为Mat_，它重载了operator()让我们可以更方便的访问图像。

```
1 Mat_<uchar> im=outputImage;
2 im(i,j)=im(i,j)/div*div+div/2;
```

2.2指针法

Mat 的 uchar*p = ptr(i)的方式可以取出图像的行指针，这样访问图像数据的时候就可以遍历每行使用p[j]取出图像特定行列的像素值。

该方法速度最快，效率最高，但是有可能出现越界问题。

2.3迭代法

采用 Mat_< Vec3b >::iterator it; 的方式声明一个迭代器，该方法比较安全，不会越界，但是效率低。

关闭

评论排行

图片集	(0)
Jupyter--闪退 解决办法	(0)
python-numpy小结3	(0)
Leetcode--Path Sum I,II,	(0)
python-numpy小结2	(0)
Python-Numpy小结1	(0)
传统物体检测	(0)
Fast_rcnn+Caffe配置以及	(0)
欢迎使用CSDN-markdown	(0)
caffe环境问题以及ubuntu	(0)

推荐文章

- * CSDN日报20170828——《4个方法快速打造你的阅读清单》
- * Android检查更新下载安装
- * 动手打造史上最简单的Recycleview 侧滑菜单
- * TCP网络通讯如何解决分包粘包问题
- * SDCC 2017之区块链技术实战线上峰会
- * 快速集成一个视频直播功能

上述方法的代码例子如下：

```

1  /*
2  主函数，声明了一个void colorReduce()函数，该函数分别用三种方式访问图像像素，运行例子的时候将不用的函
3  颜色空间缩减
4  对于三通道图像，一个像素对应的颜色有一千六百多万种，用如此多的颜色可能会影响算法性能。颜色空间缩减E
5  该例子没有采用查找表，而是直接对每个像素进行乘除操作，方便理解图像访问方式。
6  */
7  #include <opencv2/opencv.hpp>
8  #include<iostream>
9  using namespace std;
10 using namespace cv;
11 void colorReduce(Mat& inputImage, Mat& outputImage, int div);
12
13 int main()
14 {
15
16     Mat srcImage = imread("D:\\wvoo\\lena.jpg");
17     imshow("原始图像", srcImage);
18     Mat dstImage;
19     dstImage.create(srcImage.rows, srcImage.cols, srcImage.type());//效果图的大小、类型与原图片相同
20
21     //记录起始时间
22     double time0 = static_cast<double>(getTickCount());
23     //调用颜色空间缩减函数
24     colorReduce(srcImage, dstImage, 128);
25     //计算运行时间并输出
26     time0 = ((double)getTickCount() - time0) / getTickFrequency();
27     cout << "此方法运行时间为： " << time0 << "秒" << endl; //输出运行时间
28     //显示效果图
29     imshow("效果图", dstImage);
30     waitKey(0);
31     return 0;
32 }
33 //使用指针的方式访问

```

关闭

```
34 //void colorReduce(Mat& inputImage, Mat& outputImage, int div){
35 // outputImage = inputImage.clone();
36 // int rows = outputImage.rows;
37 // int cols = outputImage.cols;
38 // for (int i = 0; i < rows; ++i){
39 //     uchar* dataout = outputImage.ptr<uchar>(i);
40 //     for (int j = 0; j < cols; ++j){
41 //         dataout[j] = dataout[j] / div*div + div / 2;
42 //     }
43 // }
44 //}
45 //使用at方法访问
46 //void colorReduce(Mat& inputImage, Mat& outputImage, int div){
47 // outputImage = inputImage.clone();
48 // int rows = outputImage.rows;
49 // int cols = outputImage.cols;
50 // for (int i = 0; i < rows; ++i){
51 //     for (int j = 0; j < cols; ++j){
52 //         outputImage.at<Vec3b>(i, j)[0] = outputImage.at<Vec3b>(i, j)[0] / div*div + div / 2;
53 //         outputImage.at<Vec3b>(i, j)[1] = outputImage.at<Vec3b>(i, j)[1] / div*div + div / 2;
54 //         outputImage.at<Vec3b>(i, j)[2] = outputImage.at<Vec3b>(i, j)[2] / div*div + div / 2;
55 //     }
56 // }
57 //}
58 //使用迭代器的方式访问
59 void colorReduce(Mat& inputImage, Mat& outputImage, int div)
60 //{
61 // outputImage = inputImage.clone();
62 // //模板必须指明数据类型
63 // Mat_<Vec3b>::iterator it = outputImage.begin<Vec3b>();
64 // Mat_<Vec3b>::iterator itend = outputImage.end<Vec3b>();
65 // //也可以通过指明cimage类型的方法不写begin和end的类型
66 // /*Mat_<Vec3b> cimage = outputImage;
67 // Mat_<Vec3b>::iterator itout =outputImage.begin<Vec3b>();
68 // Mat_<Vec3b>::iterator itoutend = cimage.end();*/
```

关闭

```
69 // for (; it != itend; it++/*, it++*/)
70 // {
71 //     (*it)[0] = (*it)[0] / div*div + div / 2;
72 //     (*it)[1] = (*it)[1] / div*div + div / 2;
73 //     (*it)[2] = (*it)[2] / div*div + div / 2;
74 // }
75 //}
```

参考博客：

<http://www.cnblogs.com/Xiaoyan-Li/p/5792796.html>

http://blog.csdn.net/qq_29540745/article/details/52443697

顶

0

踩

0

上一篇 [Faster-rcnn voc数据集制作](#)

下一篇 [图片集](#)

相关文章推荐

- [opencv学习（5）图像像素的访问、颜色通道的分...](#)
- [Opencv3编程入门学习笔记（三）之访问图像像素...](#)
- [【免费】深入理解Docker内部原理及网络配置--王...](#)
- [Android入门实战](#)
- [OpenCV访问图像像素的方法](#)
- [Opencv访问图像像素的三种方法](#)

- SDCC 2017之区块链技术实战线上峰会--蔡栋
- opencv中访问图像像素方式
- php零基础到项目实战
- opencv学习笔记（十四）——图像像素的访问
- C语言及程序设计入门指导
- 5天搞定深度学习框架Caffe
- OpenCV访问图像像素的方法收集以及自己实践中...
- opencv之访问图像像素的 三种方法
- opencv(8)---访问图像像素
- opencv学习三种常用方法访问操作图像像素

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-660-0108

| 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知信息技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



关闭