OpenCV 图像处理

HighGUI：图形用户界面

- 图像载入、显示和输出

  图像输入：

```
Mat imread(const string& filename, intflags=1);
```

图像显示：

```
void imshow(const string& winname, InputArray mat);
```

图像输出：

```
bool imwrite(const string& filename, InputArray img, const vector<int>&
params=vector<int>());
```

- 示例程序： 见教材P70

线性滤波

- 图像滤波与滤波器
  - 图像滤波
    - 尽量保留图像细节特征的条件下对目标图像的噪声进行抑制
    - 消除图像中的噪声成分叫做图像的平滑化或滤波操作
    - 信号或图像的能量大部分集中在低频和中频段
    - 在高频段，有用的信息经常被噪声淹没
    - 低通是模糊，高通是锐化
  - 目的
    - 尽量抽出对象的特征作为图像识别的特征模式；
    - 消除图像数字化时混入的噪声
  - 要求

        - 不能损坏图像的轮廓及边缘等重要信息；
        - 使得图像清晰视觉效果好。
        -

    - 线性滤波器

        - 低通滤波器：允许低频率通过
        - 高通滤波器：允许高频率通过
        - 带通滤波器：允许一定范围频率通过
        - 带阻滤波器：阻止一定范围频率通过并允许其他频率通过
        - 全通滤波器：允许所有频率通过，仅改变相位关系
        - 陷波滤波器：阻止一个狭窄频率（单频率）通过

- 平滑处理

    - 平滑处理（smoothing）也称模糊处理（bluring），可以用来减少图像上的噪点或者失真。

    - 平滑处理是低频增强的空间滤波技术，一般采用简单平均法进行，就是求近邻像素点的平均亮度值。

    - 邻域的大小与平滑的效果直接相关。邻域越大，平滑效果越好，但是也会使得边缘信息损失大，使得图像变得模糊。

    - 平滑处理的方法包括：

        - 方框滤波 —— BoxBlur函数
        - 均值滤波（领域平均滤波）—— Blur函数
        - 高斯滤波 —— GaussianBlur函数
        - 中值滤波 —— medianBlur函数
        - 双边滤波 —— bilateralFilter函数

- 邻域算子与线性邻域滤波 领域算子（局部算子）是利用给定像素周围的像素值来决定此像素的最终输出的一种算子，线性邻域滤波是一种常用的邻域算子，像素的输出决定于输入像素的加权和，即卷积运算，如下图：

45	60	98	127	132	133	137	133
46	[illegible]			126	128	131	133
47	[illegible]			119	123	135	137
47	[illegible]			113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

$f(x,y)$

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

$h(x,y)$

=

69	95	116	125	129	132
68	[illegible]	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

$g(x,y)$

线性滤波处理的输出像素值 `g(i, j)` 是输入像素值 `f(i+k, j+I)` 的加权和，如下：

$$g(i,j)=\sum_{k,I} f(i+k,j+I)h(k,I)$$

其中，h(k, l) 我们称之为“核”，是滤波器的加权系数，及滤波器的“滤波系数”。

上式可以简写为：

$$g=f\otimes h$$

- 方框滤波 （boxFilter）

方框滤波被封装在boxblur函数中，即boxblur函数的作用是使用方框滤波器来模糊一张图片。

函数原型如下：

```
void boxFilter( InputArray src, OutputArray dst, int ddepth, Size ksize, Point
anchor=Point(-1, -1), boolnormalize=true, int borderType=BORDER_DEFAULT );
```

boxFilter()函数方框滤波所用的核表示如下:

$$K=a\begin{bmatrix}1 & 1 & 1 & \cdots & 1 & 1\\ 1 & 1 & 1 & \cdots & 1 & 1\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ 1 & 1 & 1 & \cdots & 1 & 1\end{bmatrix}$$

- 均值滤波

输出图像的每一个像素是核窗口内输入图像对应像素的平均值（所有像素加权系数相等），即归一化后的方框滤波

理论简析

均值滤波是典型的线性滤波算法，主要方法为邻域平均法。 即：用一片图像区域的各个像素的均值来代替原图像中的各个像素值。

一般需要在图像上对目标像素给出一个模版（内核），该模版包括了其周围的临近像素（比如以目标像素为中心的周围8（3*3-1）个像素，构成一个滤波模版，即去掉目标像素本身）。

再用模版中的全体像素的平均值来代替原来像素值。即对待处理的当前像素点（x，y），选择一个模版，该模版由近邻的若干像素组成，求模版中所有像素的均值.

再把该均值赋予当前像素点（x，y），作为处理后图像在改点上的灰度点g(x，y)。

均值滤波的缺陷

均值滤波不能很好地保护图像细节，使图像变得模糊

OpenCV Blur函数

blur函数的作用是：对输入的图像src进行均值滤波后用dst输出。其原型如下：

```
void blur( InputArray src, OutputArray dst, int ddepth, Size ksize, Point
anchor=Point(-1, -1), boolnormalize=true, int borderType=BORDER_DEFAULT );
```