# 湖南科技大学课程教案
## （章节、专题首页）

授课教师：王志喜　　职称：副教授　　单位：计算机科学与工程学院

| | |
|---|---|
| 课 程 名 称 | 计算机图形图像技术 |
| 章节、专题 | 图像的基础运算 |
| 教学目标及基 本 要 求 | 掌握图像的基础运算和相应的OpenCV函数，以及图像基础运算的程序设计方法。 |
| 教 学 重 点 | 图像的算术逻辑运算和统计运算 |
| 教 学 难 点 | 图像的关系运算 |
| 教学内容与时 间 分 配 | （1）算术逻辑运算（0.4课时）<br>（2）统计运算（0.4课时）<br>（3）常用的函数运算（0.2课时）<br>共计1课时。 |
| 习　　题 | 第12.4.1节（程序设计题）。 |

# 第12章　图像的基础运算

## 12.1　算术逻辑运算

这里只介绍两个图像的算术逻辑运算、图像与数量的算术逻辑运算以及图像的关系运算。在使用C++编写OpenCV应用程序时，通常使用矩阵表达式实现这些运算。

## 12.1.1 加减运算

### 1. 计算方法

（1）图像的加减。两个大小和类型一致的图像 $f(x,y)$ 和 $g(x,y)$ 的和与差分别为

$$s(x,y)=f(x,y)+g(x,y)$$
$$s(x,y)=f(x,y)-g(x,y)$$

（2）图像与数量的加减。图像 $f(x,y)$ 和数量 $v$ 的和与差分别为

$$s(x,y)=f(x,y)+v$$
$$s(x,y)=f(x,y)-v$$

数量 $v$ 和图像 $f(x,y)$ 的和与差分别为

$$s(x,y)=v+f(x,y)$$
$$s(x,y)=v-f(x,y)$$

## 2. 矩阵表达式

可以统一写成下列形式，其中src1和src2可以是两个源数组（类型和大小必须一致），或一个源数组和一个Scalar值，结果数组的大小和类型与源数组相同。

- dst = src1 + src2
- dst = src1 - src2

【举例】这里只给出部分代码。

```
Mat2f X(4, 4), Y(4, 4), W(4, 4); // 4×4双通道单精度数矩阵
W = X + Y; // W(i) = X(i) + Y(i)
W = X - Y; // W(i) = X(i) - Y(i)
Y = X + Scalar::all(0.75); // Y(i) = X(i) + 0.75
Y = Scalar::all(0.75) - X; // Y(i) = 0.75 - X(i)
```

## 12.1.2 乘除运算

### 1. 计算方法

（1）图像的乘除。两个大小和类型一致的图像 $f(x,y)$ 和 $g(x,y)$ 的积与商分别为

$$s(x,y)=f(x,y)\times g(x,y)$$
$$s(x,y)=f(x,y)\div g(x,y)$$

（2）图像与数值的乘除。图像 $f(x,y)$ 和实数 $v$ 的积与商分别为

$$s(x,y)=f(x,y)\times v$$
$$s(x,y)=f(x,y)\div v$$

实数 $v$ 和图像 $f(x,y)$ 的积与商分别为

$$s(x,y)=v\times f(x,y)$$
$$s(x,y)=v\div f(x,y)$$

## 2. 矩阵表达式

有下列6种形式，其中src1和src2是两个类型和大小一致的源数组，scale是一个double数，结果数组的大小和类型与源数组相同。

- dst = src1.mul(src2)
- dst = src1 * scale
- dst = scale * src2
- dst = src1 / src2
- dst = src1 / scale
- dst = scale / src2

【举例】这里只给出部分代码。

```
Mat2f X(4, 4), Y(4, 4), W(4, 4); // 4×4双通道单精度数矩阵
W = X.mul(Y) * 1.5; // W(i) = X(i)×Y(i)×1.5
W = 1.5 / Y; // W(i) = 1.5 / Y(i)
```

## 12.1.3 选取运算

### 1. 计算方法

（1）对应像素的选取。两个大小和类型一致的图像 $f(x,y)$ 和 $g(x,y)$ 的较大值与较小值分别为

$$s(x,y)=\max\big(f(x,y),g(x,y)\big)$$
$$s(x,y)=\min\big(f(x,y),g(x,y)\big)$$

（2）像素与数量的选取。图像 $f(x,y)$ 和数量 $v$ 的较大值与较小值分别为

$$s(x,y)=\max\big(f(x,y),v\big)$$
$$s(x,y)=\min\big(f(x,y),v\big)$$

## 2. 矩阵表达式

可以统一写成下列形式，其中src1和src2是两个源数组（类型和大小必须一致）或一个源数组和一个double值，结果数组的大小和类型与源数组相同。

- dst = max(src1, src2)
- dst = min(src1, src2)

【举例】这里只给出部分代码。

```
Mat2f X(4, 4), Y(4, 4), W(4, 4); // 4×4双通道单精度数矩阵
W = max(X, Y); // W(i) = max(X(i), Y(i))
W = max(0.5, Y); // W(i) = max(0.5, Y(i))
```

## 12.1.4 逻辑运算

### 1. 计算方法

（1）图像的逻辑运算。两个大小和类型一致的图像 $f(x,y)$ 和 $g(x,y)$ 的按位与、按位或以及按位异或分别为

$$s(x,y)=f(x,y)\,\text{and}\,g(x,y)$$
$$s(x,y)=f(x,y)\,\text{or}\,g(x,y)$$
$$s(x,y)=f(x,y)\,\text{xor}\,g(x,y)$$

（2）图像与数量的逻辑运算。图像 $f(x,y)$ 和数量 $v$ 的按位与、按位或以及按位异或分别为

$$s(x,y)=f(x,y)\,\text{and}\,v$$
$$s(x,y)=f(x,y)\,\text{or}\,v$$
$$s(x,y)=f(x,y)\,\text{xor}\,v$$

## 2. 矩阵表达式

可以统一写成下列形式，其中src1和src2可以是两个源数组（类型和大小必须一致），或一个源数组和一个Scalar值，结果数组的大小和类型与源数组相同。

- dst = src1 & src2
- dst = src1 | src2
- dst = src1 ^ src2
- dst = ~src

【举例】这里只给出部分代码。

```
Mat2b X(4, 4), Y(4, 4), W(4, 4); // 4×4双通道字节矩阵
W = X & Y; // W(i) = X(i) And Y(i)
W = X & Scalar::all(3); // W(i) = X(i) And 3
W = ~X; // W(i) = ~X(i)
```

## 12.1.5 关系运算

### 1. 计算方法

（1）对应像素的关系。检查两个大小和类型一致的图像 $f(x,y)$ 和 $g(x,y)$ 的对应像素是否满足指定的关系。

$$s(x,y)=\big(f(x,y)\,\mathrm{op}\,g(x,y)\big)$$

其中，关系运算符 op 为 $=$、$\neq$、$>$、$\geq$、$<$ 或 $\leq$ 之一。

（2）像素与数量的关系。检查图像 $f(x,y)$ 的每个像素与数量 $v$ 是否满足指定的关系。

$$s(x,y)=\big(f(x,y)\,\mathrm{op}\,v\big)$$

其中，关系运算符 op 为 $=$、$\neq$、$>$、$\geq$、$<$ 或 $\leq$ 之一。

## 2. 矩阵表达式

可以统一写成下列形式。

- dst = (src1 op src2)
- dst = (src1 op value)
- dst = (value op src2)

其中，两源数组的类型和大小必须一致，关系运算符op为==、!=、>、>=、<或<=之一，value是一个double数，结果数组是单通道字节数组。

【举例】这里只给出部分代码。

```
Matlb X(4, 4), Y(4, 4), W(4, 4); // 4×4单通道字节矩阵
W = (X < Y); // W(i) = (X(i) < Y(i))
W = (X < 1.5); // W(i) = (X(i) < 1.5)
```

## 12.1.6 应用示例

本示例首先使用选取运算过滤源图像中的低亮度像素（低亮度的元素改为指定值，其余不变），然后使用关系运算和乘法运算完成源图像的二值化（高亮度的元素改为255，其余改为0）。运行结果如图12-1所示。


图12-1 灰度图像的过滤与二值化

```
// OperationApplication.Cpp
#include<opencv2/opencv.hpp>
using namespace cv;
int main()
{   Mat X = imread("lena.jpg", 0); // 灰度图像
    if(X.empty()) return -1;
    imshow("源图像", X); // 显示源图像
    Mat Y = max(X, 128); // 过滤，小于128的元素改为128
    imshow("过滤图像", Y);
    Y = (X > 128) * 255; // 二值化，大于128的元素改为255，其余改为0
    imshow("二值图像", Y);
    waitKey();
}
```

# 12.2 统计运算

## 12.2.1 常用的统计运算

在本小节中，为了方便描述，对于宽度和高度分别是 $w$ 和 $h$ 的图像 $f(x,y)$，使用 $(x,y)\in f$ 表示 $(x,y)\in[0,w)\times[0,h)$，使用 $\left|f\right|$ 表示 $w\times h$，即图像 $f$ 的像素数。

### 1. 非0像素数

图像 $f(x,y)$ 的非0像素数为 $\left|\left\{f(x,y):(x,y)\in f \text{ and } f(x,y)\neq 0\right\}\right|$。

### 2. 像素和

图像 $f(x,y)$ 的像素和为 $\sum\left\{f(x,y):(x,y)\in f\right\}$。

### 3. 像素平均值和像素均方差

图像 $f(x,y)$ 的像素平均值 $\overline{f}=\left(1/|f|\right)\sum\left\{f(x,y):(x,y)\in f\right\}$，像素均方差 $\overline{f}=\sqrt{\left(1/|f|\right)\sum\left\{\left[f(x,y)-\overline{f}\right]^2:(x,y)\in f\right\}}$。

### 4. 最小像素和最大像素

图像 $f(x,y)$ 的最小像素为 $\min\left\{f(x,y):(x,y)\in f\right\}$，最大像素为 $\max\left\{f(x,y):(x,y)\in f\right\}$。

## 5. 几种常用的图像范数

- $C$-范数：也称为$L_\infty$范数，表示像素绝对值的最大值，即图像 $f(x,y)$ 的$C$-范数定义为 $\|f\| = \max\{|f(x,y)| : (x,y) \in f\}$。
- $L_1$-范数：表示像素绝对值的和，即图像 $f(x,y)$ 的$L_1$-范数定义为 $\|f\| = \sum\{|f(x,y)| : (x,y) \in f\}$。
- $L_2$-范数：就是欧几里得范数，表示像素平方和的平方根，即图像 $f(x,y)$ 的$L_2$-范数定义为 $\|f\| = \sqrt{\sum\left\{[f(x,y)]^2 : (x,y) \in f\right\}}$。

## 12.2.2 OpenCV中的相关函数

### 1. 非0元素数

【函数原型】int countNonZero(InputArray src);
【功能】计算数组中的非零元素数目。
【参数】src是输入数组，为单通道数组。

### 2. 数组元素的和

【函数原型】Scalar sum(InputArray src);
【功能】独立地为每一个通道计算数组元素的和。
【参数】src是输入数组。

### 3. 数组元素的平均值

【函数原型】Scalar mean(InputArray src, InputArray mask = noArray());
【功能】独立地为每一个通道计算数组元素的平均值。
【参数】src是输入数组，mask是操作掩码。

## 4. 数组元素的均方差

【函数原型】void meanStdDev(InputArray src, OutputArray mean, OutputArray stddev, InputArray mask = noArray());

【功能】独立地为每一个通道计算数组元素的平均值和均方差。

【参数】

- src：输入数组。
- mean：保存平均值的变量。
- stddev：保存标准差的变量。
- mask：操作掩码。

## 5. 数组元素的最小值和最大值

【函数原型】void minMaxLoc(InputArray src, double *minVal, double *maxVal = 0, Point *minLoc = 0, Point *maxLoc = 0, InputArray mask = noArray());

【功能】寻找数组元素的最小值和最大值。

【参数】

- src：输入数组，单通道数组。
- minVal和maxVal：指向保存最小值和最大值的变量。
- minLoc和maxLoc：指向保存最小值位置和最大值位置的变量。
- mask：操作掩码。

## 6. 数组的范数

【函数原型】double norm(InputArray src1, int normType = NORM_L2, InputArray mask = noArray());

【功能】计算src的范数。多通道数组视为单通道处理。

【参数】

- src：输入图像。
- normType：范数类型，通常选用NORM_INF（$C$-范数）、NORM_L1（$L_1$-范数）或NORM_L2（$L_2$-范数）。
- mask：操作掩码。

# 12.3 常用的函数运算♫

只介绍OpenCV中对像素值进行非线性变换的几个常用的数学函数。

## 1. 绝对值

【函数原型】MatExpr abs(const MatExpr &src);

【功能】对数组内每个元素求绝对值。

【参数】src是源数组。

## 2. 幂函数

【函数原型】void pow(InputArray src, double power, OutputArray dst);

【功能】对数组内每个元素求幂。

【参数】

- src：源数组。
- power：幂指数。
- dst：输出数组，类型和大小与源数组一致，必要时重建。

【说明】

- dst($i$) = src($i$) ^ power。
- 若幂指数不是整数，则使用输入元素的绝对值进行计算，相当于调用pow(abs(src), power, dst)。

### 3. 指数函数

【函数原型】void exp(InputArray src, OutputArray dst);
【功能】对数组内每个元素求以$e$为底的指数函数值。
【参数】src和dst分别是源数组和输出数组。
【说明】$dst(i) = exp(src(i))$。

### 4. 对数函数

【函数原型】void log(InputArray src, OutputArray dst);
【功能】计算每个数组元素的绝对值的自然对数。
【参数】src和dst分别是源数组和输出数组。
【说明】$dst(i) = log(|src(i)|)$。0对应一个大负数。

# 12.4 练习题

## 12.4.1 程序设计题

1. 首先使用OpenCV装入一幅灰度图像，然后使用函数min()过滤掉源图像中亮度大于指定值（例如128）的像素，并显示源图像和结果图像以便对比。

2. 首先使用OpenCV装入一幅灰度图像并显示该图像，然后计算出该图像的最小像素值min和最大像素值max，最后将每个像素都减去min再乘以255/max以后显示结果图像。

## 12.4.2 阶段实习题

1. 首先使用OpenCV装入一幅灰度图像，并创建一个滑块（初始值为255）。然后使用关系运算和Mat的成员函数copyTo()过滤掉源图像中亮度大于滑块位置的像素（过滤掉的像素亮度值改为0），并显示结果图像。

2. 编写一个简单的向源图像加入噪音的程序。首先随机选择若干噪声位置，然后对每个噪声位置随机产生噪声值，将噪声值与源图像相应像素直接平均作为结果像素值，源图像中非噪声位置的像素值保持不变。要求考虑灰度图像、彩色图像、字节图像和浮点数图像等情况。