深圳大学 SHENZHEN UNIVERSITY

计算机视觉

南区计软328

群名称:计算机视觉2024春
群　号:731395587

深圳大学
计算机与软件学院

实验内容

序号	实验主题	实验内容	实验要求	实验时数	每组人数	实验类型
1	图像处理应用实验	1．熟悉图像的表示及基本元素、通道操作； 2．掌握基本图像增强方法； 3．掌握OpenCV计算机视觉库；	必做	6	1	讲授+实验
2	图像特征提取及综合应用实践	1．熟悉图像处理基本操作； 2．掌握图像边缘检测原理； 3．掌握图像基本特征抽取以及在实际问题中的应用；	必做	6	1	讲授+实验
3	计算机视觉系统实践	1．熟悉计算机视觉分类任务； 2．掌握数据集的准备及模型训练过程； 3．培养应用计算机视觉解决问题的能力；	必做	6	1	讲授+实验

涉及学科

- 数字图像处理
- 计算机视觉
- 模式识别
- 程序设计

实验考察形式

- 三次实验报告（在Blackboard发布与提交）
- 平时编程练习及表现
- 实验汇报

实验考察内容

- 计算机视觉与模式识别基础知识掌握能力
- 算法设计能力
- 编程及系统架构能力
- 论文写作及表达能力

实验一

实验一：图像增强应用实践

实验目的：

1. 熟悉图像的表示及基本元素、通道操作；
2. 掌握基本图像增强方法；
3. 掌握OpenCV计算机视觉库；

实验要求：

1. 实验提交文件为实验报告和相关程序代码，以压缩包的形式提交，实验报告命名规则为“计算机视觉-学号-姓名-实验报告1.doc”，其他文件打包成压缩文件，命名为“计算机视觉-学号-姓名-实验报告1-其他.zip”；
2. 所有素材和参考材料需列明出处，实验报告中的图片和程序代码建议标注个人水印或标识信息：姓名，班级，学号信息；

二、实验内容：

不调用库函数，自己动手编程实现图像增强相关方法，并与OpenCV的库函数进行效果对比分析；

三、实验时间

实验报告统一命名为：计算机视觉-学号-姓名-实验报告1.doc

如有提交其他文件或程序，统一命名为：计算机视觉-学号-姓名-实验报告1-其他.zip

实验时间：2024年3月4日至2024年4月13日（注意按时提交，不要延期）

实验报告提交时间：2024年4月13日下午5:00

实验一

计算机视觉-学号-姓名-实验报告1.doc [兼容模式] - Word

一、实验目的与要求

实验目的：

1. 熟悉图像的表示及基本元素、通道操作；
2. 掌握基本图像增强方法；
3. 掌握 OpenCV 计算机视觉库；

实验要求：

1. 实验提交文件为实验报告和相关程序代码，以压缩包的形式提交，实验报告命名规则为“计算机视觉-学号-姓名-实验报告 1.doc”，其他文件打包成压缩文件，命名为“计算机视觉-学号-姓名-实验报告 1-其他.zip”；

2. 所有素材和参考材料需列明出处，实验报告中的图片和程序代码建议标注个人水印或标识信息：姓名，班级，学号信息；

二、实验内容与方法

实验内容：不调用库函数，自己动手编程实现图像增强相关方法，并与 OpenCV 的库函数进行效果对比分析；

三、实验步骤与过程

提示（提交作业时请删除红底标注提示信息）：根据实验内容自己组织填写，内容截图插入实验报告后大小要调得适中，并且必须加上相关的说明，每页图片建议不超过 6 张，排版的美观也加入评分标准。（以后所有的实验报告都是这个要求，不再重复说明）。

四、实验结论或体会

提示（提交作业时请删除红底标注提示信息）：有重要结论与心得体会，包括技术上的或设计上的、操作上的，理论上的等等。

五、思考题

对比不同图像增强方法的优势、劣势，包括其合适的应用场景。

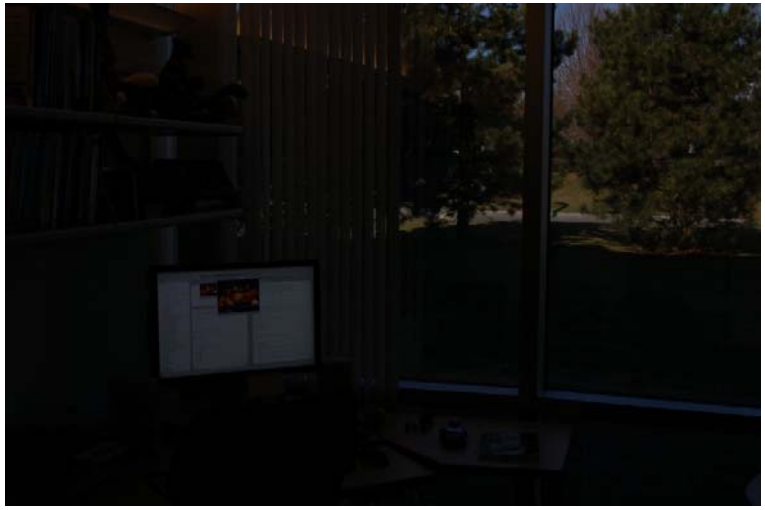
实验一

图像增强方法：

- 基于滤波的图像增强方法——局部二值模式、中值滤波、均值滤波、高斯滤波、图像锐化、Gabor滤波；
- 基于统计、函数映射的图像增强方法——直方图均衡化、直方图规定化、灰度变换、伽马变换；
- 其他图像增强方法——基于字典学习、稀疏表示、深度学习的方法。

- 实验一

car



office

foggyInput

coins

实验一

cameraman

高斯噪声 均值0方差0.005

均值0方差0.01

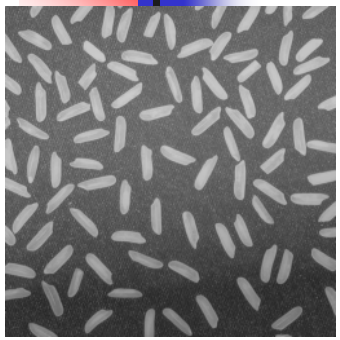
均值0方差0.015

椒盐噪声 噪声密度0.1

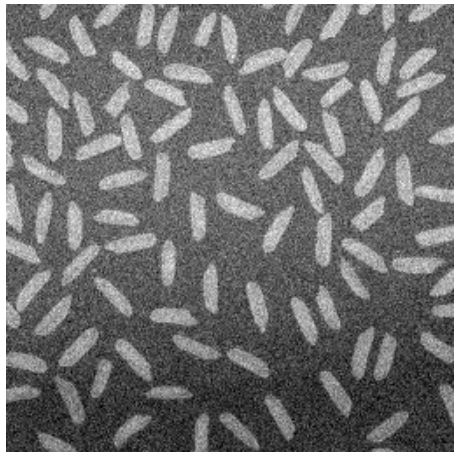
噪声密度0.2

噪声密度0.3

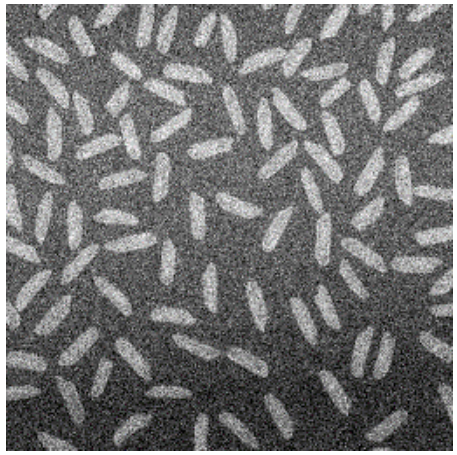
实验一



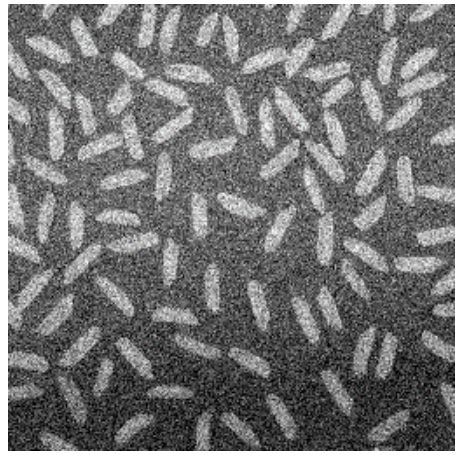
rice



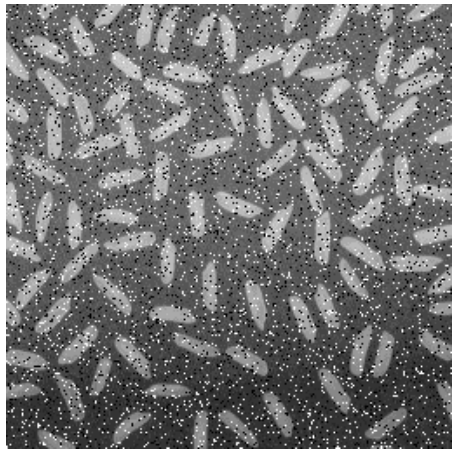
高斯噪声 均值0方差0.005



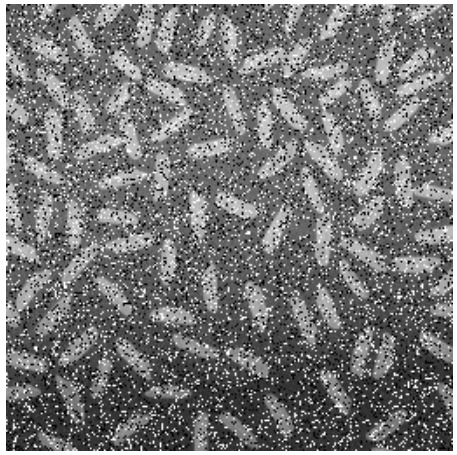
均值0方差0.01



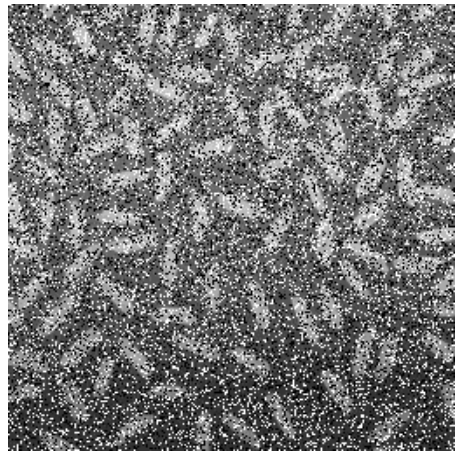
均值0方差0.015



椒盐噪声 噪声密度0.1

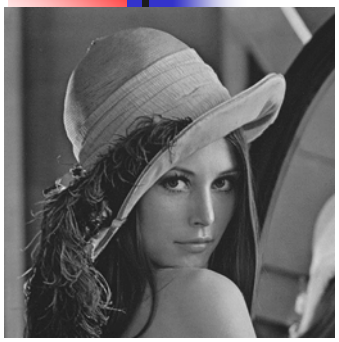


噪声密度0.2



噪声密度0.3

实验一



lena

高斯噪声 均值0方差0.005

均值0方差0.01

均值0方差0.015

椒盐噪声 噪声密度0.1

噪声密度0.2

噪声密度0.3

- 实验一

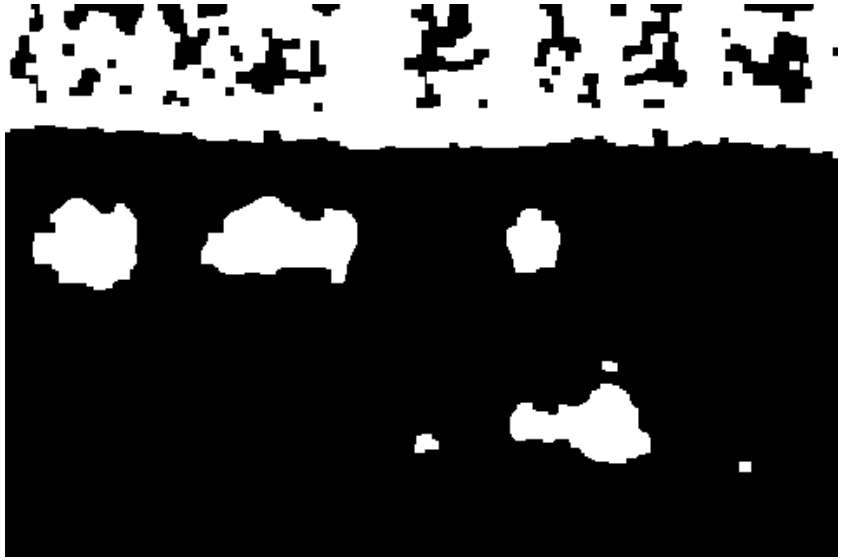
coloredChips

peppers

- 实验一

boldt

实验一



group

Group_binary

- 实验一

Road

OpenCV中关于图像操作

基本数据结构：

cv::Mat //矩阵

cv::Point //点坐标 例： cv::Point3d p(x0, x1, x2);

cv::Scalar //值向量 例： cv::Scalar red(0, 0, 255);

cv::Size //大小向量 例： cv::Size sz(w, h);

cv::Rect //矩形向量 例： cv::Rect rt(x, y, w, h);

cv::vec_ //向量 例： Vec2f v2f(x0,x1);

OpenCV中关于图像操作

图像基本信息

```
img_gray.rows; // height of image
img_gray.cols; // width of image
img_gray.channels(); // number of channels
```

图像像素

```
img_src.at<uchar>(y, x); //gray image
img_src.at<cv::Vec3b>(y, x); //color image
```

图像区域（ROI）

```
img_rgb(cv::Rect(105, 18, 50, 50)); // clip image
img_roi(roi_rect).setTo(cv::Scalar(255, 255, 255)); //roi
```

OpenCV中关于图像操作

基本运算

最大，最小，均值，方差

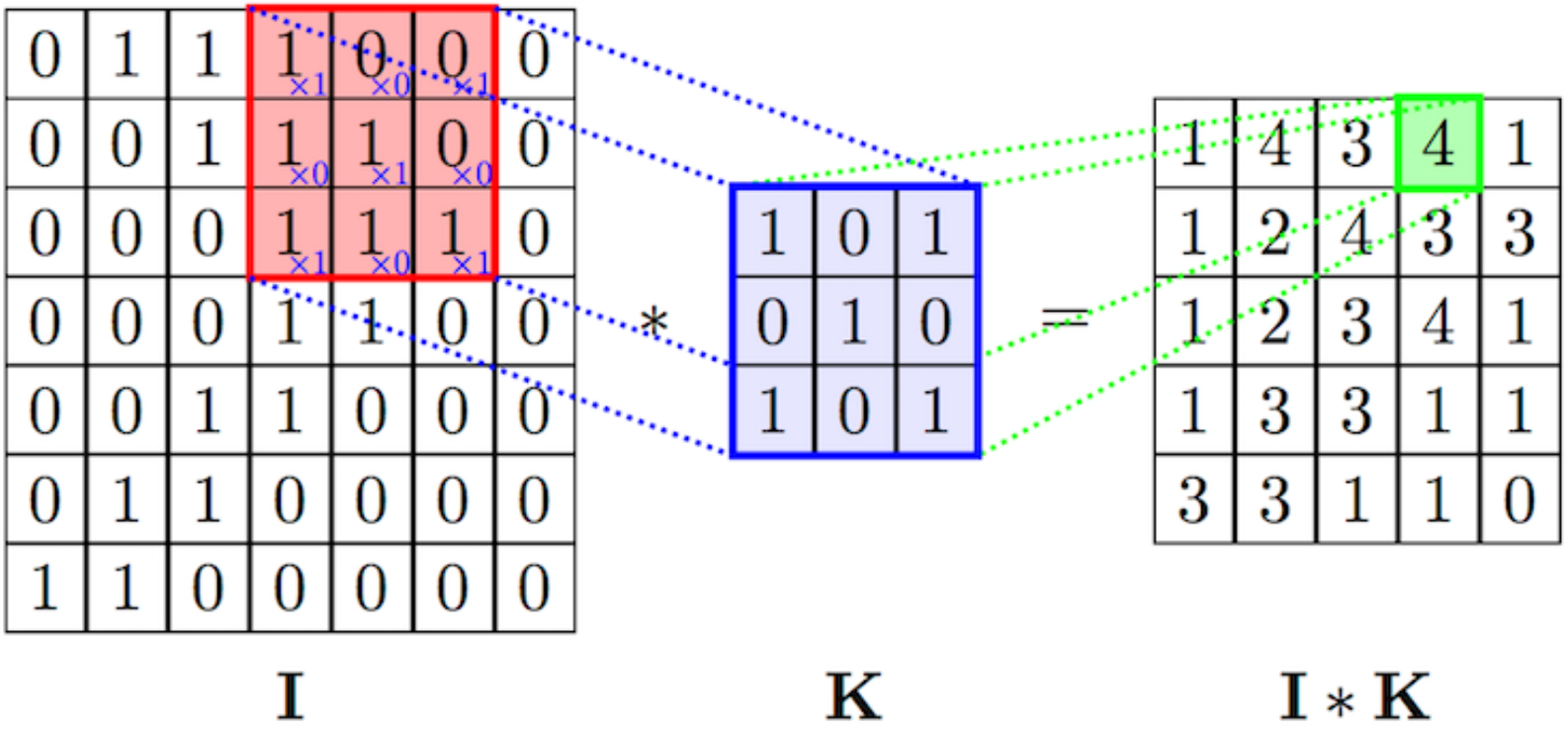
求和，非零数

加、减、乘、点乘

等其矩阵操作

```
cv::addWeighted( roi1, alpha, roi2, beta, 0.0, roi1 );
```

空间卷积及卷积核（局部操作）



卷积核：大小，值

- OpenCV中关于图像操作

空间卷积滤波器（去噪）

均值滤波

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

void cv::blur(InputArray src, OutputArray dst, Size ksize, Point anchor = Point(-1,-1), int borderType = BORDER_DEFAULT)

中值滤波

void cv::medianBlur(InputArray src, OutputArray dst, int ksize)

高斯滤波

void cv::GaussianBlur(InputArray src, OutputArray dst, Size ksize, double sigmaX, double sigmaY = 0, int borderType = BORDER_DEFAULT)

双边滤波

void cv::bilateralFilter(InputArray src, OutputArray dst, int d, double sigmaColor, double sigmaSpace, int borderType = BORDER_DEFAULT)

● OpenCV中关于图像操作

滤波操作注意事项

① 输入图像为单通道无符号整型；
② 新建两个变量，一个存储输入图像，一个存储结果图像，每次操作的滤波结果存入结果图像；
③ 注意边界，不要越界，防止内存溢出，访问出错；
④ 所有计算是在浮点型状态下进行；
⑤ 最后要把结果转换为0-255的整型，再传给结果图像。

```
void CBasicMethod::MySobel(Mat InputImg, Mat &Gray32Mat, float *Operator, int OperatorSize)
{
    //InputImg是灰度图像

    Gray32Mat.create(InputImg.size(), CV_32F);//CV_16S - 16-bit signed integers ( -32768..32767 )
    Mat_<uchar> dataMat = InputImg;
    Mat_<float> dataResultMat = Gray32Mat;
    int oh = OperatorSize / 2;//算子Operator的半径
    int ow = OperatorSize / 2;
    for (int j = 0; j < InputImg.rows; j++)
    {
        for (int i = 0; i < InputImg.cols; i++)
        {
            float datavalue = 0;
            for (int p = -oh; p <= oh; p++)
            {
                for (int q = -ow; q <= ow; q++)
                {
                    if (p + j<0 || p + j>InputImg.rows - 1 || q + i<0 || q + i>InputImg.cols - 1)
                    {
                        continue;
                    }
                    datavalue += dataMat(j + p, i + q)*Operator[(p + oh)*OperatorSize + (q + ow)];
                }
            }
            dataResultMat(j, i) = saturate_cast<float>(datavalue);
        }
    }
}
```

OpenCV中关于图像操作

图像二值化操作

dst=threshold(src, thresh, maxval, type)

src： 输入图，只能输入单通道图像，通常来说为灰度图
dst： 输出图
thresh： 阈值
maxval： 当像素值超过了阈值（或者小于阈值，根据type来决定），所赋予的值
type：二值化操作的类型，包含以下5种类型：
cv2.THRESH_BINARY； cv2.THRESH_BINARY_INV；
cv2.THRESH_TRUNC； cv2.THRESH_TOZERO；
cv2.THRESH_TOZERO_INV

形态学膨胀腐蚀操作

dilate(const Mat &src, Mat &dst, Mat kernel, Point anchor=Point(-1,-1), int iterations=1)
src：输入图，可以多通道，深度可为CV_8U、CV_16U、CV_16S、CV_32F或CV_64F。
dst：输出图，和输入图尺寸、形态相同。
kernel：结构元素，如果kernel=Mat()则为预设的3×3矩形，越大膨胀效果越明显。
anchor：原点位置，预设为结构元素的中央。
iterations：执行次数，预设为1次，执行越多次膨胀效果越明显。

erode(const Mat &src, Mat &dst, Mat kernel, Point anchor=Point(-1,-1), int iterations=1)
src：输入图，可以多通道，深度可为CV_8U、CV_16U、CV_16S、CV_32F或CV_64F。
dst：输出图，和输入图尺寸、形态相同。
kernel：结构元素，如果kernel=Mat()则为预设的3×3矩形，越大膨胀效果越明显。
anchor：原点位置，预设为结构元素的中央。
iterations：执行次数，预设为1次，执行越多次腐蚀效果越明显。

深圳大学 SHENZHEN UNIVERSITY

- OpenCV中关于图像操作

形态学膨胀腐蚀操作

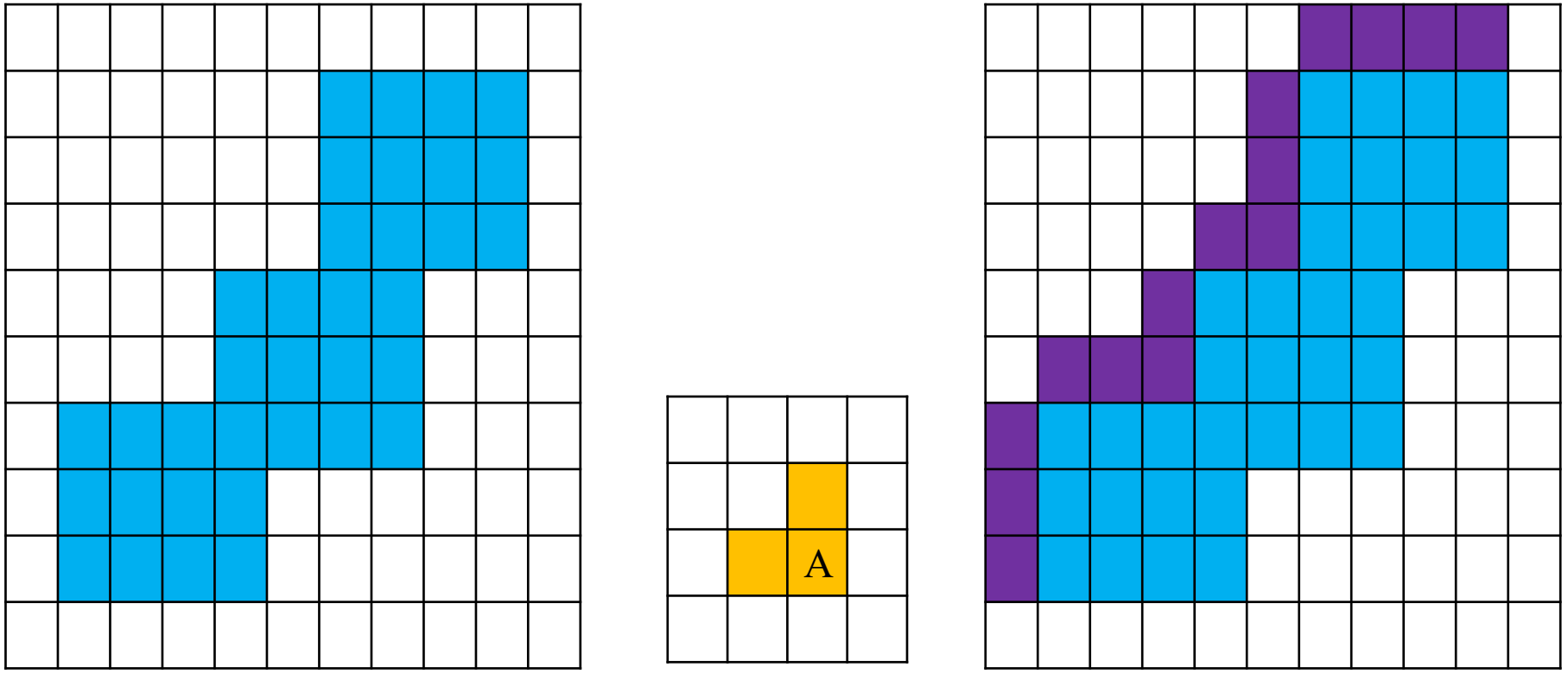
```
void CForFreshMen::DilateErodeSelfStruc()
{
    Mat img=imread("thresh.bmp",1);

    int sizeH=5;
    int sizeW=5;
    Mat element (sizeH,sizeW,CV_8U,Scalar(0));
    int element_val[5][5]={0 ,0 ,1, 0, 0,
    0, 0, 1, 0, 0,
    1, 1, 1, 1, 1,
    0, 0, 1, 0, 0,
    0, 0, 1, 0, 0};
    //结构元素赋初值，也可以通过element.at<uchar>(j,i)=x;来赋值
    for(int j=0;j<element.rows;j++)
    {
        uchar *data=element.ptr<uchar>(j);
        for(int i=0;i<element.cols;i++)
        {
            cout<<element_val[j][i];
            //*data++=saturate_cast<uchar>(element_val[j][i]);
            *data++=element_val[j][i];
        }
        cout<<endl;
    }
    cout<<element<<endl;
    Mat eroded;  //腐蚀图像
    Mat dilated; //膨胀图像
    erode(img,eroded,element,Point(-1,-1),2);//腐蚀两次
    dilate(img,dilated,element,Point(-1,-1),2);//膨胀两次
    imshow("original",img);
    imshow("erode",eroded);
    imshow("dilate",dilated);
    cvWaitKey(0);

}
```

- OpenCV中关于图像操作

二值图像上的噪声去除——形态学滤波

膨胀：填补分割后目标中的孔洞

$$D = X \oplus B = \{(x, y) \mid [\hat{B}_{x,y} \cap X] \subseteq X\}$$

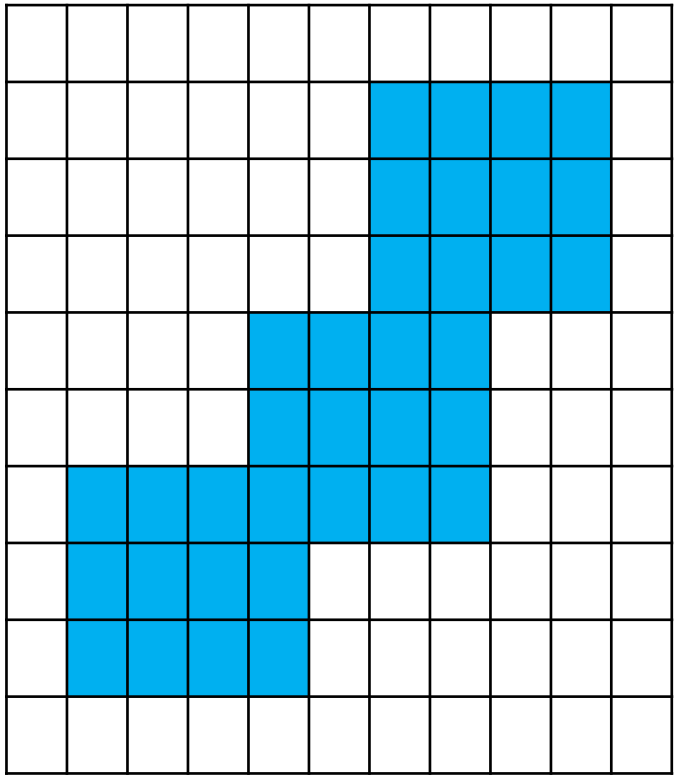


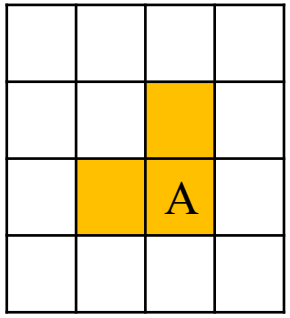
- OpenCV中关于图像操作

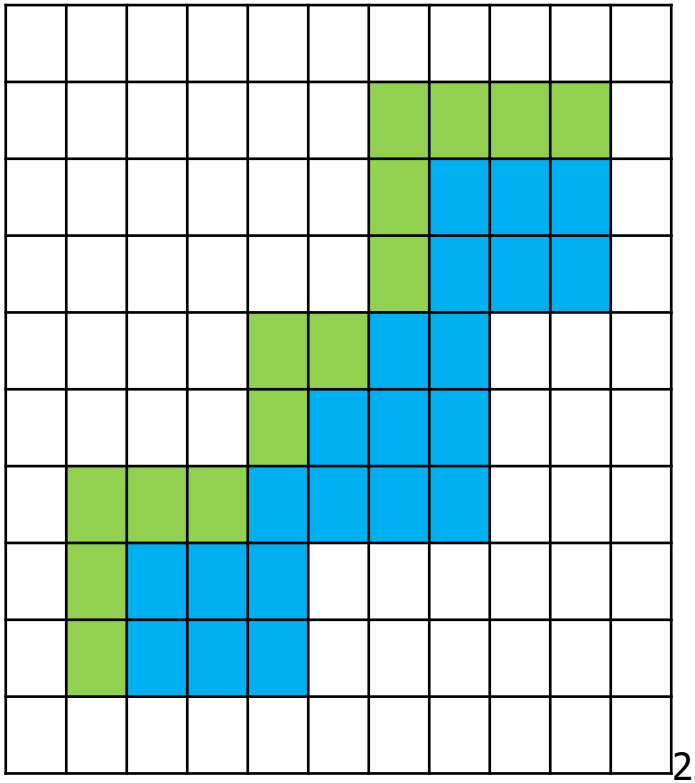
二值图像上的噪声去除——形态学滤波

腐蚀：消除目标图像孤立噪声点

$$E = X \odot B = \{(x,y) \mid \hat{B}_{x,y} \subseteq X\}$$







- OpenCV中关于图像操作

二值图像上的噪声去除——形态学滤波

开运算：消除细小目标、在纤细点处分离目标、平滑大边界，而不明显改变其面积

$$X \circ B = (X \odot B) \oplus B$$

闭运算：填充目标内部细小孔洞、连接临近目标，在不明显改变面积的情况下平滑其边界

$$X \bullet B = (X \oplus B) \odot B$$

- OpenCV中关于图像操作

形态学滤波器（OCR，指纹识别）

void cv::**morphologyEx**(InputArray src, OutputArray dst, int op, InputArray kernel, Point anchor = Point(-1,-1), int iterations = 1, int borderType = BORDER_CONSTANT, const Scalar & borderValue = morphologyDefaultBorderValue())

```
enum   cv::MorphTypes {
         cv::MORPH_ERODE = 0,
         cv::MORPH_DILATE = 1,
         cv::MORPH_OPEN = 2,
         cv::MORPH_CLOSE = 3,
         cv::MORPH_GRADIENT = 4,
         cv::MORPH_TOPHAT = 5,
         cv::MORPH_BLACKHAT = 6,
         cv::MORPH_HITMISS = 7
       }
```

```
enum   cv::MorphShapes {
         cv::MORPH_RECT = 0,
         cv::MORPH_CROSS = 1,
         cv::MORPH_ELLIPSE = 2
       }
```

上机练习内容：

1. OpenCV环境配置；
2. 实现对图像的读取；
3. 把一副彩色图像的三个通道变成3个单通道图像存储到硬盘上并显示；
4. 计算一幅单通道图像的直方图；
5. 编程实现对一幅单通道图像的边缘检测。

上机练习内容：

6. 对一幅灰度图像进行直方图均衡化

7.对图像进行二值化操作；

8. 图像形态学操作；