1.图像膨胀的 Matlab 实现：

可以使用 imdilate 函数进行图像膨胀,imdilate 函数需要两个基本输入参数，即待处理的输入图像和结构元素对象。结构元素对象可以是 strel 函数返回的对象，也可以是一个自己定义的表示结构元素邻域的二进制矩阵。此外，imdilate 还可以接受两个可选参数：PADOPT(padopt) ——影响输出图片的大小、PACKOPT(packopt).——说明输入图像是否为打包的二值图像(二进制图像)。举个实例如下：

步骤 1，首先创建一个包含矩形对象的二值图像矩阵。

```
>> BW=zeros(9,10);

>> BW(4:6,4:7) =1

BW =

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0
0 0 0 1 1 1 1 0 0 0
0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

步骤 2，使用一个 3×3 的正方形结构元素对象对创建的图像进行膨胀。

```
>> SE=strel('square',3)

SE =
Flat STREL object containing 9 neighbors.

Neighborhood:
1 1 1
1 1 1
1 1 1
```

步骤 3，将图像 BW 和结构元素 SE 传递给 imdilate 函数。

```
>> BW2=imdilate(BW,SE)

BW2 =

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

步骤 4，显示结果。

```
>> imshow(BW,'notruesize')

>> imshow(BW2,'notruesize')
```

2.图像腐蚀的 Matlab 实现：

可以使用 imerode 函数进行图像腐蚀。imerode 函数需要两个基本输入参数：待处理的输入图像以及结构元素对象。此外，imerode 函数还可以接受 3 个可选参数：PADOPT(padopt)——影响输出图片的大小、PACKOPT(packopt).——说明输入图像是否为打包的二值图像(二进制图像)。M——指定原始图像的行数。

以下程序示例说明了如何对某一副具体图像进行腐蚀操作，腐蚀前后的效果对比如图末。

步骤 1，读取图像 cameraman.tif （该图像是 Matlab 当前目录下自带的图片）

```
>> BW1=imread('cameraman.tif');
```

步骤 2，创建一个任意形状的结构元素对象

```
>> SE=strel('arbitrary',eye(5));
```

步骤 3，以图像 BW1 和结构元素 SE 为参数调用 imerode 函数进行腐蚀操作。

```
>> BW2=imerode(BW1,SE);
```

步骤 4，显示操作结果

```
>> imshow(BW1)
>> figure,imshow(BW2)
```

3.膨胀和腐蚀联合操作（图像开运算操作）：

下面以图像开启为例，说明如何综合使用 imdilate 和 imerode 这两个函数，实现图像处理操作。

步骤 1，创建结构元素：

```
>> clear;close all
>> SE = strel('rectangle',[40 30]); %注意:结构元素必须具有适当的大小,既可以删电流线又可以删除矩形.
```

步骤 2,使用结构元素腐蚀图像: %将会删除所有直线,但也会缩减矩形

```
>> BW1=imread('circbw.tif');
>> BW2=imerode(BW1,SE);
>> imshow(BW2)
>> figure,imshow(BW1)
```

步骤 3,恢复矩形为原有大小,使用相同的结构元素对腐蚀过的图像进行膨胀.

```
>> BW3=imdilate(BW2,SE);
>> figure,imshow(BW3)
```

4.基于膨胀与腐蚀的形态操作——骨架化和边缘检测

（1）骨架化：

某些应用中，针对一副图像，希望对图像中所有对象简化为线条，但不修改图像的基本结构，保留图像基本轮廓，这个过程就是所谓的骨架化。提供了专门的函数 bwmorph，可以实现骨架化操作。

```
>> clear;close all
>> BW1=imread('circbw.tif');
>> BW2=bwmorph(BW1,'skel',Inf);
>> imshow(BW1)
>> figure,imshow(BW2)
```

（2）边缘检测

对于一副灰度二进制图像,如果图像像素值为 1,则该像素的状态为 ON,如果其像素值为 0,则该像素的状态为 OFF。在一副图像中，如果图像某个像素满足以下两个条件：

1.该像素状态为 ON,

2.该像素邻域中有一个或多个像素状态为 OFF。

则认为该像素为边缘像素。

Matlab 中提供了专门的函数 bwperim,可以用于判断一副二进制图像中的哪些像素为边缘像素。

以下程序代码示例就是利用 bwperim 函数，对图像 circbw.tif 进行边缘检测，其边缘像素检测效果如尾图。

```
>> clear;close all
>> BW1=imread('circbw.tif');
>> BW2=bwperim(BW1);
>> imshow(BW1)
>> figure,imshow(BW2)
```

基于腐蚀和膨胀的形态操作函数如下:

bwhitmiss 图像逻辑"与"操作,该函数使用一个结构元素对图像进行腐蚀操作后,再使用第二个结构元素对图像进行腐蚀操作

imbothat 从原始图像中减去经过形态关闭后的图像,该函数可用来寻找图像中的灰度槽

imclose 闭合操作.首先对图像进行膨胀,然后再对膨胀后的图像进行腐蚀,两个操作使用同样的结构元素

imopen 开启操作,首先对图像进行腐蚀,然后再对腐蚀后的图像进行膨胀,两个操作使用同样的结构元素

imtophat 从原始图像中减去形态开启后的图像,可以用来增强图像的对比度