

3.2 图像处理基本算子

数字图像处理(Digital Image Processing)是指应用计算机来合成、变换已有的数字图像,从而产生一种新的效果,并把加工处理后的图像重新输出,这个过程称为数字图像处理。因此也称之为计算机图像处理(Computer Image Processing)

In [16]:

```
pkg load image;
```

腐蚀与膨胀

In [17]:

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

SE =

Flat STREL object with 9 neighbors

Neighborhood:

```
1 1 1
1 1 1
1 1 1
```

In [21]:

```
X = rgb2gray(imread('figs/character.jpg'));
X1 = X;
```

In [22]:

```
X2 = imerode(X1, SE, 'same'); imshow([ X1, X2]);title('before and after erode');
```

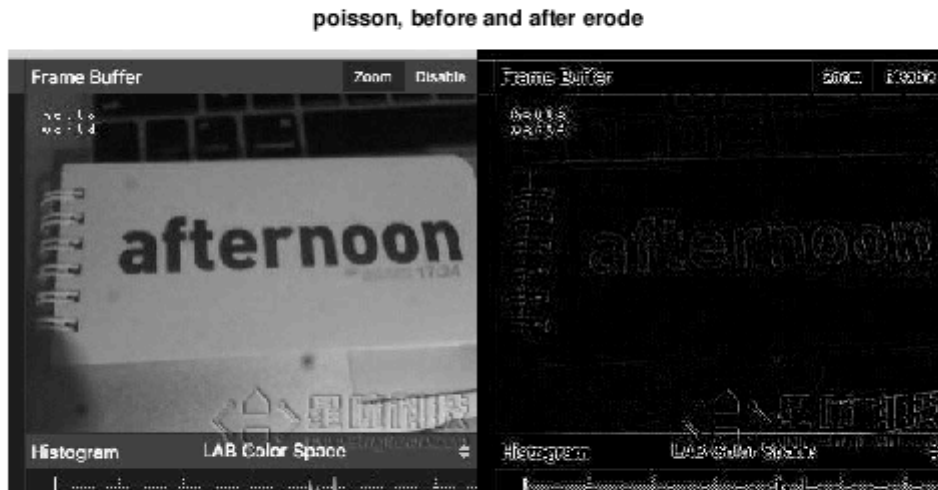


In [26]:

```
%h1 = fspecial('sobel'); % L a p l a c e 算子的五点差分格式  
h1 = [ 0 1 0 ;...  
      1 -4 1 ;...  
      0 1 0 ];
```

In [28]:

```
X3 = filter2(h1, X1); imshow([X1, X3]); title('poisson, before and after erode');
```



角点检测 - 局部特征

角点一般位于两条以上的交点上，五种类型如图所示：分别为L型、Y型、T型、X型和箭头型。



现有的角点检测算法并不是都十分的鲁棒。很多方法都要求有大量的训练集和冗余数据来防止或减少错误特征的出现。角点检测方法的一个很重要的评价标准是其对多幅图像中相同或相似特征的检测能力，并且能够应对光照变化、图像旋转等图像变化。不同的检测方法对角点有不同的意义。一个优秀的角点检测方法需要满足如下标准：

- 所有真实的角点都被发现到；
- 没有虚假的角点被当作角点检测出来；
- 角点在图像中的定位精确；
- 角点检测器必须对噪声有鲁棒性；
- 角点检测算子的效率应该是高效的。

1. 基于梯度的焦点检测: 通过计算边缘的曲率来判断角点的存在性,角点计算数值的大小不仅与边缘强度有关,而且与边缘方向的变化率有关,该方法对噪声比基于模板的角点检测方法对噪声更为敏感
2. 基于模板的焦点检测: 主要考虑像素邻域点的灰度变化,即图像亮度的变化,将与邻点亮度对比足够大的点定义为角点。常见的有Moravec、[Harris角点检测算法](https://blog.csdn.net/rqc112233/article/details/50112463) (<https://blog.csdn.net/rqc112233/article/details/50112463>)、[KLT角点检测算法](http://blog.csdn.net/tostq/article/details/49157811) (<http://blog.csdn.net/tostq/article/details/49157811>) 以及[SUSAN角点检测算法](http://blog.csdn.net/tostq/article/details/49305615) (<http://blog.csdn.net/tostq/article/details/49305615>)等
3. 基于模板梯度组合

讨论:

其他有许多用于特征检测和提取的算法，我们将会对其中大部分进行介绍。开源图像处理库OpenCV中包含的最常使用的特征检测和提取算法有：

- Harris：该算法用于检测角点；
- SIFT：该算法用于检测斑点；
- SURF：该算法用于检测角点；
- FAST：该算法用于检测角点；
- BRIEF：该算法用于检测斑点；
- ORB：该算法代表带方向的FAST算法与具有旋转不变性的BRIEF算法；

在提取特征之后，就可以通过以下方法进行特征匹配，从而实现“智能”识别：

- 暴力(Brute-Force)匹配法；
- 基于FLANN匹配法；

Hough变换与直线检测

Remark: hough() has not yet been implemented in octave-forge, let us run it in matlab!

In []:

```
%Img = rgb2hsv(imread('RectPosePic/road.jpg')); Img=Img(500:end, :, 3);
Img = rgb2gray(imread('RectPosePic/ticket2.bmp'));
%Img = imread('circuit.tif'); Img = imrotate(Img, 33, 'crop');

I_edge = edge(Img, 'canny');
subplot(2,2,1); imshow(I_edge); title('The 5original edges');

%I_edge = imdilate(I_edge, ones(3)); % 对于sharp的边不好
subplot(2,2,2); imshow(I_edge); title('The imdilate edges');

[H,T,R] = hough(I_edge, 'Theta', 1:0.1:89);
P = houghpeaks(H, 'threshold', ceil(0.5*max(H(:))));

% Find lines and plot them
lines = houghlines(I_edge,T,R,P, 'FillGap', 25, 'MinLength', 150);

max_len = 0; subplot(2,2,3); imshow(Img); hold on;
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1), xy(:,2), 'LineWidth', 2, 'Color', 'green');

    % plot beginnings and ends of lines
    plot(xy(1,1), xy(1,2), 'x', 'LineWidth', 2, 'Color', 'yellow');
    plot(xy(2,1), xy(2,2), 'x', 'LineWidth', 2, 'Color', 'red');

    % determine the endpoints of the longest line segment
    len = norm(lines(k).point1 - lines(k).point2);
    if ( len > max_len)
        max_len = len;
        xy_long = xy;
    end
end

% highlight the longest line segment
plot(xy_long(:,1), xy_long(:,2), 'LineWidth', 2, 'Color', 'cyan');
theta = atan2(xy_long(2,1)-xy_long(1,1), xy_long(2,2)-xy_long(1,2))/pi*180;
title(['rotated angle should be ', num2str(-mod(theta,90)), ' degree']); hold off;

subplot(2,2,4);
imshow(imrotate(Img, -mod(theta,90), 'bilinear', 'crop'));
```

In []:

In []:

图像增强

现代图像应用中，有几类基本的数字图像处理任务：

- 3.2.1 图像增强：如去噪、滤波、锐化等各种“滤镜”操作
- 3.2.2 图像分割：根据灰度、颜色、纹理和形状等特征把图像划分成若干互不交迭的区域
- 3.2.3 图像修复：对受到损坏的图像进行修复重建或者去除图像中的多余物体

In []:

图像增强 1：基于直方图均衡化

函数`histeq`是`image`工具箱的一个函数，它的调用格式：

- `[I1,T] = histeq(I,K)`

其中，`I`为输入图像，`K`为指定直方图均衡化后的灰度级数，`I1`为输出均衡化之后的图像，`T`为`I`到`I1`的变换矩阵。下面这个例子可以让我们直观地感受均衡化的效果

In [38]:

```
I = imread('figs/lenna.bmp'); imshow(I)
```



In [45]:

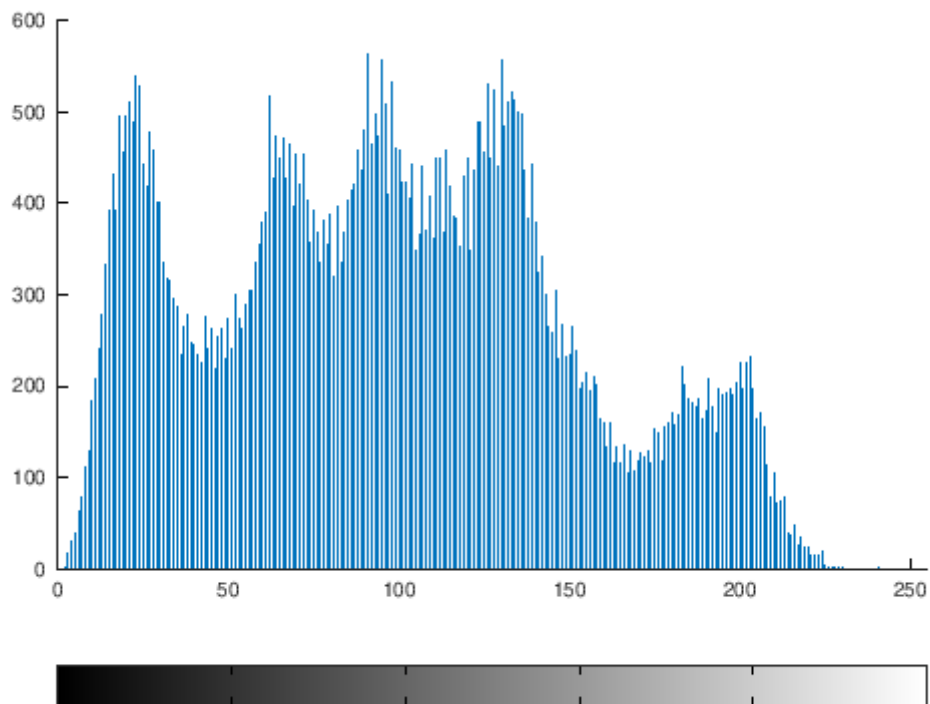
```
H = histeq(I, 10); imshow(H)
```



接着，让我们看一下两个图的直方图统计结果：

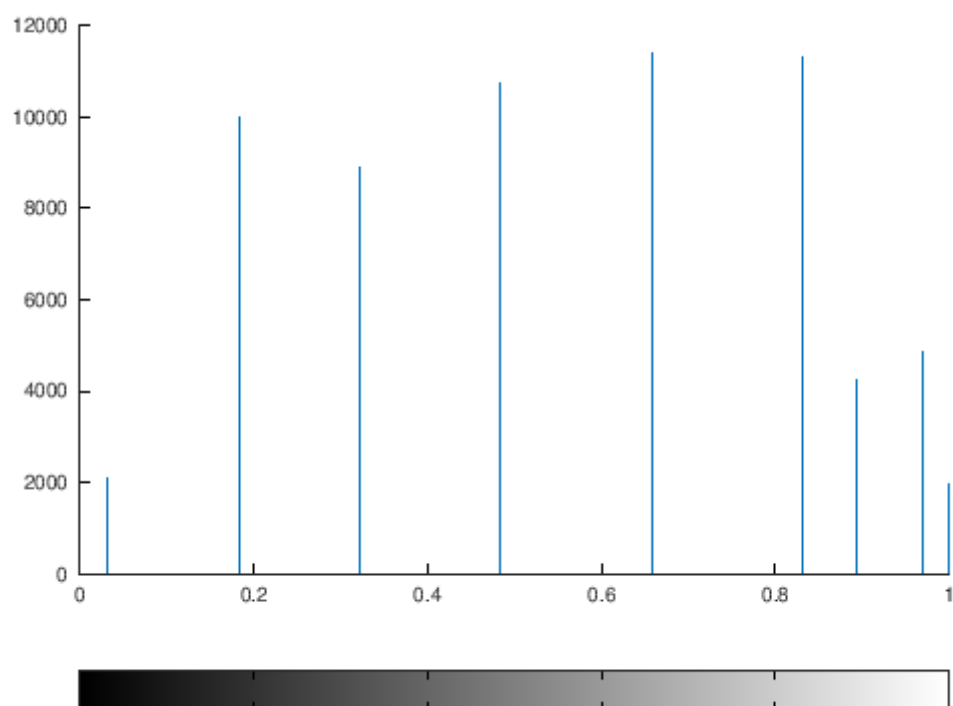
In [47]:

```
imhist(I)
```



In [50]:

```
imhist(H);
```



直方图均衡化的数学原理和其他语言实现

请参考：

- [histeq从用法到原理](http://blog.sina.com.cn/s/blog_14d1511ee0102wwyt.html) (http://blog.sina.com.cn/s/blog_14d1511ee0102wwyt.html)
- [OpenCV之图像直方图均衡化](https://zhuanlan.zhihu.com/p/61061557?from_voters_page=true) (https://zhuanlan.zhihu.com/p/61061557?from_voters_page=true)

- 更复杂的图像情形(图像来源于网络):



其他增强技巧:

读者可以用其他类型的图像, 如:[医学X光胸片](https://blog.csdn.net/moxibingdao/article/details/105985737) (<https://blog.csdn.net/moxibingdao/article/details/105985737>) 通过图像增强的方式突出肺窗。其中, 图像增强的技巧不仅包括直方图均衡化, 还有对比度、线性展宽、灰级窗、线性动态范围调整等



[这里](https://blog.csdn.net/qq_21905401/article/details/76694677?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase) (https://blog.csdn.net/qq_21905401/article/details/76694677?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase) 是一个CXR的公开数据集的介绍以及读写介绍。

练习: 从MIMIC数据库下载多张CXR影像, 通过调用直方图均衡化突出显示肺窗区域, 如上图所示 (标注可选)。

In []:

图像增强 2 : 基于空间滤波的图像去噪:

In [56]:

```
I_noise = rgb2gray(imread('figs/canal_noise.png'));
```

In [74]:

```
fspecial('average', [3, 3])
```

ans =

```
0.11111 0.11111 0.11111
0.11111 0.11111 0.11111
0.11111 0.11111 0.11111
```

In [67]:

```
I_H = imfilter(I_noise, fspecial('average', [3, 3]));
I_H2 = imfilter(I_H, fspecial('average', [3, 3]));
```

In [68]:

```
imshow([I_noise, 255*ones(size(I_noise,1),15), I_H, 255*ones(size(I_noise,1),15), I_H2])
```



噪声是少了，但是变糊了！

In [77]:

```
%%锐化%%  
f_sobel = fspecial('sobel')  
f_laplacian = fspecial('laplacian',0)
```

锐化 added to session magics.

f_sobel =

```
1  2  1  
0  0  0  
-1 -2 -1
```

f_laplacian =

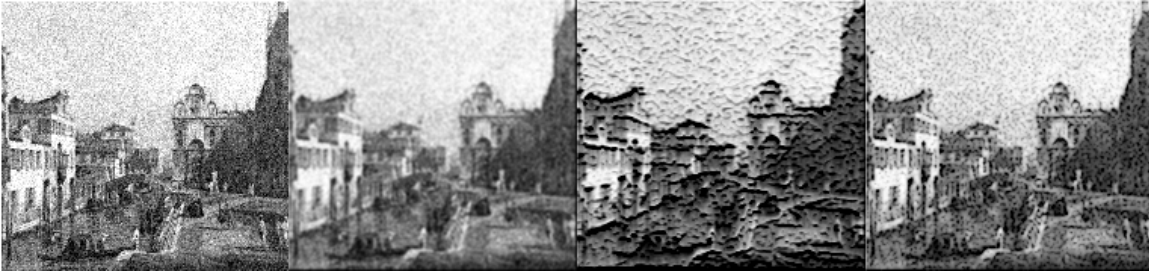
```
0  1  0  
1 -4  1  
0  1  0
```

In [78]:

```
B1 = imfilter(I_H2,f_sobel);  
B2 = imfilter(I_H2,f_laplacian);
```

In [79]:

```
imshow([I_noise, I_H2, I_H2-B1, I_H2-B2])
```



In []:

练习： 任选一张图像（黑白或彩色）,图像分辨率调整为256x256,自主添加Gauss噪声，并调用不同类型filter算子去噪，表现你的最优去噪效果，title中显示信噪比。

医学影像

已经转换好的mat格式

In [3]:

```
clear; load Ground_truth; whos
```

Variables in the current scope:

Attr	Name	Size	Bytes	Class
====	====	====	=====	=====
	Gt	181x217x181	56873096	double

Total is 7109137 elements using 56873096 bytes

In [6]:

```
% permute the data with dimension index [1 3 2]
head = permute(Gt, [1 3 2]);
```

方法一： show it in a cycyle from slice 20 to 200, it works in octave GUI

In [9]:

```
for k = 20:200; imshow(head(:, :, k)); title(num2str(k)); drawnow; pause(0.05);end;
```

200



方法二： 写入gif动画

In [8]:

```
%% output with a gif, this is not work in octave
% load Ground_truth;
% head = permute(Gt, [1 3 2]);
filename = 'head.gif';
figure(1);

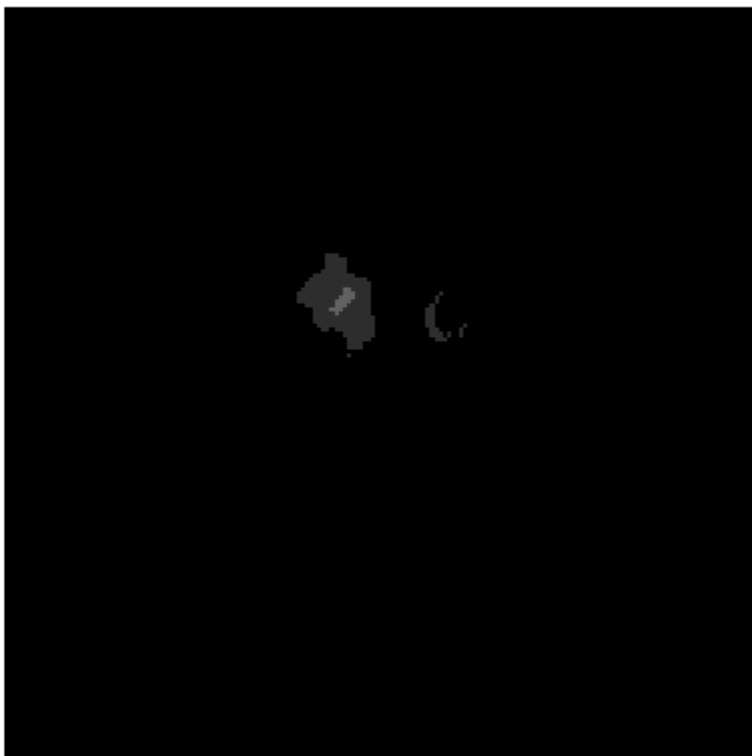
for k = 20:200

    imshow(head(:, :, k));
    title(num2str(k));

    drawnow;
    frame = getframe(1);
    im = frame2im(frame);
    [imind, cm] = rgb2ind(im, 256);
    if n == 1;
        imwrite(imind, cm, filename, 'gif', 'Loopcount', inf);
    else
        imwrite(imind, cm, filename, 'gif', 'WriteMode', 'append');
    end
end
```

error: getframe: not implemented for gnuplot graphics toolkit
error: called from
 getframe at line 68 column 5

20



方法三： 在jupyter里有没有其他显示方案？

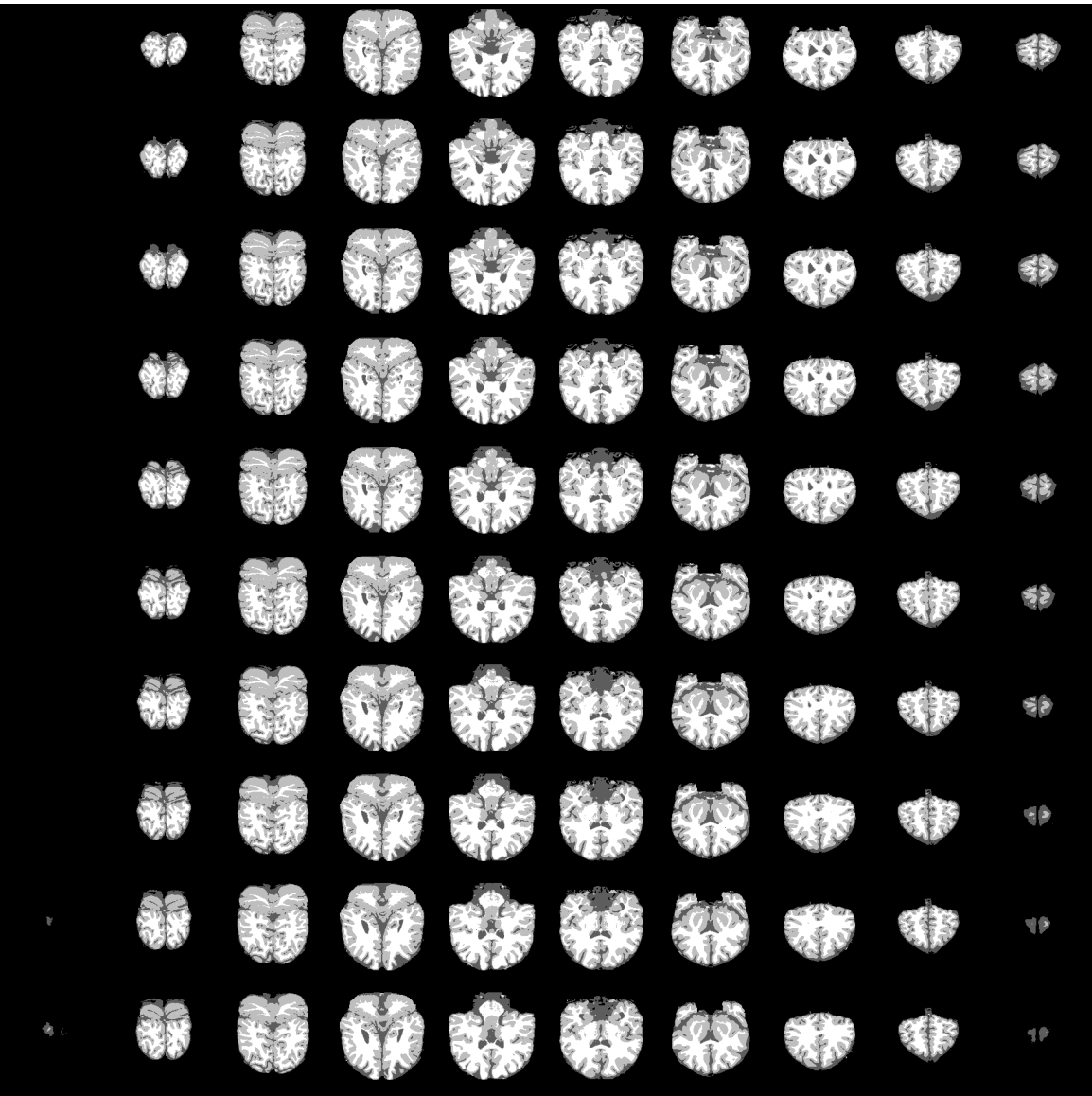
In []:

In []:

In []:

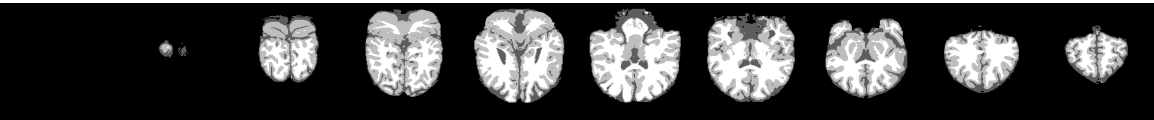
In []:

In [13]:



In [14]:

```
imshow(reshape(head(:, :, 1:20:200), 181, 1810))
```



练习： 请寻找合适的方法读入以下任一格式的影像数据：

aneurism.raw *MR-head.nrrd* *MRbrain.tar.gz*

选用前述三种方式之一可视化数据。