

## 图像信息处理

1、图像成像的各种形式：可见光、X 射线、超声波、红外线

2、图像信息处理涵盖的方面(what is image processing)

[1]Image acquisition(图像获取)

Automatic aperture correction and color balance(自动光圈校准和颜色平衡)

Image reconstruction (图像修复)

[2]Display and print

Automatic size adjustment (自动大小调整)

Color correction (CMYK vs. RGB) (颜色校正)

[3]Storage and Transmission (存储和传输)

High efficient storing: less temporal-spatial cost (高效的存储以及较低的时空消耗)

Convenient transmission: Lossless, efficient, and secure (便利的传输)

[4]Enhancement and restoration (增强与修复)

Image restoration: Inpainting (图像的修补)

Improve visual effect: dehazing, retinex, deblur, super-resolution (增强可视性)

[5]Recognition and understanding

Character recognition: OCR, object detection: face detection (物体探测, 文字识别)

Scene understanding: image retrieval, scene classification (场景理解, 图像检索, 场景分类)

3、图像的压缩

目的：存储图像信息

图像特点：以像素为单位，矩形区域

Encoding methods: 不压缩，无损压缩，有损压缩

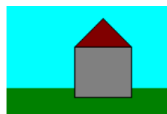
Representative formats: BMP, JPEG, TIFF, GIF, PNG

4、BMP 图像的调色板

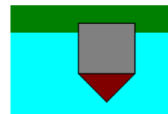
大小：N\*4 bytes, 1 byte rgbBlue to specifies the blue part of the color, 1 byte rgbGreen to specifies the green part of the color, 1 byte rgbRed to specifies the blue part of the color

1 byte rgbReserved must always be set to zero

5、BMP 图像的存储



pixels displayed on the screen



pixels stored in .bmp-file

You do not need to turn around the rows manually. The API functions which also display the bitmap will do that for you automatically.

Another important thing is that the number of bytes in one row must always be adjusted to fit into the border of a multiple of four. You simply append zero bytes until the number of bytes in a row reaches a multiple of four

6 bytes that represent a row in the bitmap: A0 37 F2 8B 31 C4

must be saved as: A0 37 F2 8B 31 C4 **00 00**

## 6、光圈对于成像的影响

光圈孔径大的时候，会导致模糊，但同时如果孔径太小，可通过的光线就会很少，导致光强太弱，同时当孔径小到一定程度时，会发生衍射现象。

## 7、弥散圆(circle of confusion)

通过使用透镜可以会聚更多的光线到一个成像点，只是只有在特殊的位置上才能达到这个目标，即大家以前就学习过的透镜聚焦。位于合适距离的物体点可以成清晰像，其他的点则会产生弥散圆。在焦点前后，光线开始聚集和扩散，点的影像变模糊，形成一个扩大的圆。如果弥散圆的直径小于人眼的鉴别能力，在一定范围内实际影像产生的模糊是不能辨认的。不能辨认的直径叫做容许弥散圆。焦点前后各有一个容许弥散圆。对于 6 寸照片，观察距离 25 - 30cm，大概是底片对角线  $1/1000-1/1500$ ，

## 8、光圈的大小与景深有密切关系

焦点前后各有一个容许弥散圆，他们之间的距离叫做景深。景深随镜头的焦距、光圈值、拍摄距离而变化

### (1)、镜头光圈：

光圈越大，景深越小；光圈越小，景深越大；

### (2)、镜头焦距

镜头焦距越长，景深越小；焦距越短，景深越大；

### (3)、拍摄距离

距离越远，景深越大；距离越近，景深越小

## 9、数码相机拍照 6 步曲

(1)摄影物时，景物反射的光线通过数码相机的镜头透射到 CCD 上。

(2)当 CCD 曝光后，光电二极管受到光线的激发而释放出电荷，生成感光元件的电信号。

(3) CCD 控制芯片利用感光元件中的控制信号线路对发光二极管产生的电流进行控制，由电流传输电路输出，CCD 会将一次成像产生的电信号收集起来，统一输出到放大器。

(4)经过放大和滤波后的电信号被传送到 ADC，由 ADC 将电信号（模拟信号）转换为数字信号，数值的大小和电信号的强度与电压的高低成正比，这些数值其实也就是图像的数据。

(5)此时这些图像数据还不能直接生成图像，还要输出到 DSP（数字信号处理器）中，在 DSP 中，将会对这些图像数据进行色彩校正、白平衡处理，并编码为数码相机所支持的图像格式、分辨率，然后才会被存储为图像文件。

(6)当完成上述步骤后，图像文件就会被保存到存储器上，我们就可以欣赏了。

## 10、色彩的分类（彩色和消色）

色彩可以分为彩色（chromatic color）和消色（achromatic color）两大类。

彩色是指红、黄、蓝等单色以及它们的混合色。彩色物体对光谱各波长的反射具有选择性，所以它们在白光照射下呈现出不同的颜色。

消色，又称非彩色，也就是我们通常所说的灰度，是指白色，黑色以及各种深浅不同的灰色。

消色物体对光谱各波长的反射没有选择性，它们是中性色。

## 11、需要了解（波长以及视网膜细胞）

色觉是指不同波长的光线作用于视网膜而在大脑中引起的感觉。人眼可见光线的波长是 390nm ~ 780nm，一般可辨出包括紫、蓝、青、绿、黄、橙、红 7 种主要颜色在内的 120 ~ 180 种不同的颜色。

视网膜是人眼中最重要的组成部分，就像相机里的感光底片，专门负责感光成像。视网膜上分布着两种视觉细胞，一种为杆状体，另一种为锥状体。杆状体细胞比较多，大约有上亿个，它对光极为灵敏，但没有区分色彩的能力。锥状体细胞则只有六、七百万个，它要在较强的照度下才能激发，它的存在使我们能够辨别各种不同的颜色。

12、三原色是目前最为代表性的色觉机制解释：

在视网膜上存在着三种分别对红、绿和蓝光线的波长特别敏感的视锥细胞或相应的感光色素，当不同波长的光线进入人眼时，与之相符或相近的视锥细胞发生不同程度的兴奋，于是在大脑产生相应的色觉；三种视锥细胞若受到同等程度的刺激，则产生消色。

13、感知的优先程度和敏感度

优先程度：同等条件下，人们往往会注意到色调 (Hue, H) 的变化，然后是饱和度 (Saturation, S)，然后是亮度 (Value, V)。

敏感度：人眼对于亮度的变化最为敏感，分辨能力最强。恰好与人眼的高动态能力相匹配。

14、颜色空间的定义

与设备相关：RGB, CMY, HSV

与设备无关：

(1) CIE XYZ: XYZ 三刺激值的概念是依据视觉的三元理论为根据的，它说明人眼具有接受三原色(红、绿、蓝)的接受器，而所有的颜色均被视作该三原色的混合色。1931 年 CIE 制定了一种假想的标准观察者，设置配色函数。XYZ 三刺激值是利用这些标准观察者配色函数计算得来的。在此基础上，CIE 于 1931 年规定了  $Y_{xy}$  颜色空间，其中 Y 为亮度，x,y 是从三刺激值 XYZ 计算得来的色坐标。它代表人类可见的颜色范围。

(2) CIE  $L^*a^*b^*$ :  $L^*a^*b^*$  颜色空间是在 1976 年制定的，它是 CIE XYZ 颜色模型的改进型，以便克服原来的  $Y_{xy}$  颜色空间存在的在 x, y 色度图上相等的距离并不相当于我们所觉察到的相等色差的问题。它的“L” (明亮度)，“a” (绿色到红色) 和“b”(蓝色到黄色)代表许多的值。与 XYZ 比较，CIE  $L^*a^*b^*$  颜色更适合于人眼的感觉。利用 CIE  $L^*a^*b^*$ ，颜色的亮度 (L)、灰阶和饱和度 (a,b) 可以单独修正，这样，图像的整个颜色都可以在不改变图像或其亮度的情况下，发生改变。

(3) CIE YUV: 在现代彩色电视系统中，通常采用三管彩色摄像机或彩色 CCD(电荷耦合器件) 摄像机，它把摄得的彩色图像信号，经分色，分别放大校正得到 RGB，再经过矩阵变换电路得到亮度信号 Y 和两个色差信号  $R-Y$ 、 $B-Y$ ，最后发送端将亮度和色差三个信号分别进行编码，用同一信道发送出去。这就是我们常用的 YUV 颜色空间。采用 YUV 颜色空间的重要性是它的亮度信号 Y 和色度信号 U、V 是分离的。如果只有 Y 信号分量而没有 U、V 分量，那么这样表示的图就是黑白灰度图。

更接近人的对颜色的感知差异：HSV

15、RGB 颜色空间

RGB 颜色模型是三维直角坐标颜色系统中的一个单位正方体，在正方体的主对角线上，各原色的量相等，产生由暗到亮的白色，即灰度。(0, 0, 0) 为黑，(1, 1, 1) 为白，正方体的其他 6 个角点分别为红、黄、绿、青、蓝和品红。RGB 颜色模型构成的颜色空间是 CIE 原色空间的一个真子集。RGB 颜色模型通常用于彩色阴极射线管和彩色光栅图形显示器 (计算机和电视机采用)。

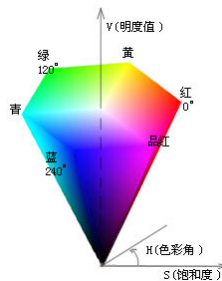
16、CMY 颜色空间

油墨或颜料的 3 种基色是以红、绿、蓝三色的补色青 (Cyan)、品红 (Magenta)、黄 (Yellow) 为基色。用 CMY 模型产生的颜色称为相减色，是因为它减掉了为视觉系统识别颜色所需要的反射光

17、CMY 空间与 RGB 空间互补，也就是用白色减去 RGB 空间中的某一颜色值就等于同样颜色在 CMY 空间中的值。RGB 是光合色，颜色是基于光线进行混合的。黑色是各种颜色的空白状态，没有任何颜色。这时若要产生颜色，就要通过增加各种色彩的光线来产生，当各类色彩都加到最大值后形成了白色。

18、HSV 颜色空间

颜色空间是从人的视觉系统出发，用色调 (Hue)、色饱和度 (Saturation) 和亮度 (Intensity, 或者 Value) 来。圆锥的顶面对应于  $V=1$ ，它包含 RGB 模型中的  $R=1$ ,  $G=1$ ,  $B=1$  三个面，故所代表的颜色较亮。色度  $H$  由绕  $V$  轴的旋转角给定。红色对应于角度  $0^\circ$ ，绿色对应于角度  $120^\circ$ ，蓝色对应于角度  $240^\circ$ 。描述颜色。在圆锥的顶点处， $V=0$ ， $H$  和  $S$  无定义，代表黑色。HSV 颜色模型构成的是一个均匀的颜色空间，采用线性的标尺，彩色之间感觉上的距离与 HSV 颜色模型坐标上点的欧几里德距离成正比。



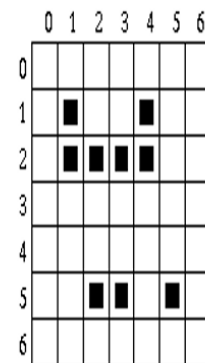
## 19、JPEG 压缩的基本思想

压缩策略：根据压缩比要求，从高频到低频逐步削减信息

好处：高频信息占用存储空间大，减少高频信息更容易获得高压缩比；低频信息可以保留物体的基本轮廓和色彩分布，最大限度维持图像质量；适合用于互联网的视觉媒体。

## 20、RLE 形成编码

- In binary images, run length coding records only areas that belong to the object in the image; the area is then represented as a list of lists.
- Each row of the image is described by a sublist, the first element of which is the row number.
- Subsequent terms are coordinate pairs; the first element of a pair is the beginning of a run and the second is the end.
- There can be several such sequences in the row.



Run length coding; the code is  $((11144)(214)(52355))$ .

首先第一个数字记录行数，其次的数字分别表示一个连续序列的起始列和末尾列，如第一行， $(1, 1)$  仅一个像素，所以为 11，之后的 4 也仅为 1 个像素，所以也是 44。因此最终结果为  $(11144)$ ，而第二行的从第一个像素到第四个像素均连续，因此只需要一个 14 即可表示，所以最后第二行的输出结果为  $(214)$ 。

## 21、二值化的优点缺点

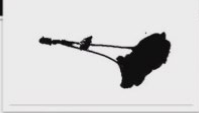
### Advantages vs. Disadvantages

Advantages:

- Less memory
- More efficient
- CAN sometimes be applied on grayscale image
- More cheap

Disadvantages:

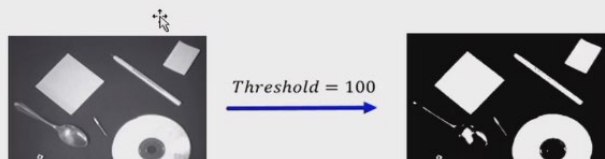
- Application field is limited;
- CANNOT be applied on 3-D data;
- Less expressive force, FAIL to convey the visual details;
- CANNOT control the contrast.



## 22、二值化图像的获得

### Acquisition of binary image

Construct a binary image: thresholding the grayscale image by reset the pixel value, i.e.

$$\begin{cases} I(x,y) = 0 & \text{if } I(x,y) < \text{Threshold} \\ I(x,y) = 255 & \text{if } I(x,y) \geq \text{Threshold} \end{cases}$$


## 23、如何选取一个好的阈值

- Step 1: 确定原始图像中像素的最大值和最小值;
- Step 2: 最小值加 1 作为 threshold 对原始图像进行二值化操作;
- Step 3: 根据对应关系确定前景和背景, 分别计算当前 threshold 下的内部协方差和外部协方差;
- Step 4: 回到 Step 2 直达到像素最大值;
- Step 5: 找到最大外部和最小内部协方差对应的 threshold.

## 24、大津算法的简单推导

### ■ Easier deduction

- $w_f = \frac{N_{Fgrd}}{N}, w_b = \frac{N_{Bgrd}}{N}, w_f + w_b = 1$
- $\mu = w_f * \mu_{Fgrd} + w_b * \mu_{Bgrd}$
- $\sigma_{between} = w_f(\mu_{Fgrd} - \mu)^2 + w_b(\mu_{Bgrd} - \mu)^2 = w_f(\mu_{Fgrd} - w_f * \mu_{Fgrd} - w_b * \mu_{Bgrd})^2 + w_b(\mu_{Bgrd} - w_f * \mu_{Fgrd} - w_b * \mu_{Bgrd})^2 \rightarrow w_b w_f (\mu_f - \mu_b)^2$

## 25、全局二值化不一定好 (选择局部二值化)

局部自适应操作设定一个局部窗口, 在整个图像上滑动该窗口; 对于每一窗口位置, 确定针

对该窗口的 threshold。

26、形态学膨胀、腐蚀、开、闭操作以及物理意义

### Dilation: enlarging the foreground

A is dilated by B

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\}$$

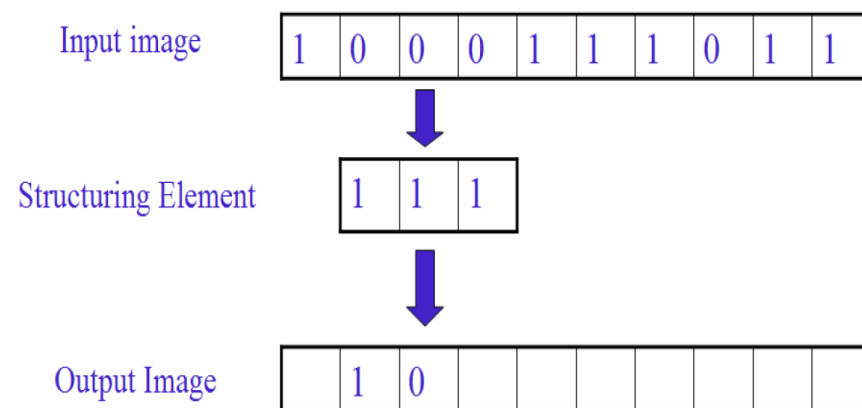
The intersection set between A and the translated B is not empty

### Physical meaning

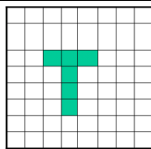
Dilation adopts the connected background pixels into the foreground, which extends its boundary and fill the holes in the foreground.

And whether "connected" is decided by the structure element.

膨胀是将与物体“接触”的所有背景点合并到该物体中，使边界向外部扩张的过程。可以用来填补物体中的空洞（其中“接触”的含义由结构元描述）。

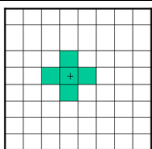


## Dilation example



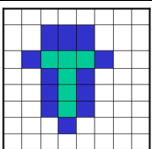
(a)

A



(b)

B



(c)

A ⊕ B

For figure (a), (0, 0) is the top left corner. For figure (b), (0,0) of the structure element is marked by "+".

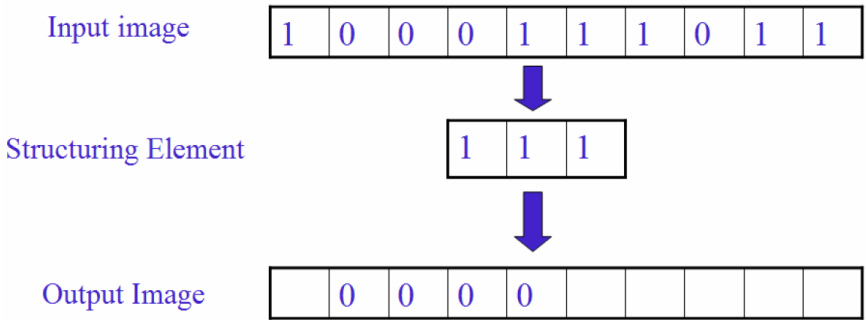
$A = \{(2,2), (2,3), (2,4), (3,3), (4,3), (5,3)\}$

$B = \{(0,0), (-1,0), (1,0), (0,-1), (0,1)\}$

$A \oplus B = \{(2,2), (2,3), (2,4), (3,3), (4,3), (5,3), (1,2), (1,3), (1,4), (2,3), (3,3), (4,3), (3,2), (3,3), (3,4), (4,3), (5,3), (6,3), (2,1), (2,2), (2,3), (3,2), (4,2), (5,2), (2,3), (2,4), (2,5), (3,4), (4,4), (5,4)\}$

对于上图(a)的图像以左上角位置为 (0, 0)，结构元素以“+”位置为参考点 (0, 0)。则A和B分别表示为：

腐蚀的物理意义：腐蚀是一种消除边界点，使边界向内部收缩的过程。可以用来消除小且无意义的物体。



膨胀：  
 由 B 对 A 膨胀所产生的二值图象 D 是满足以下条件的点(x,y)的集合：如果 B 的原点平移到点(x,y)，那么它与 A 的交集非空。

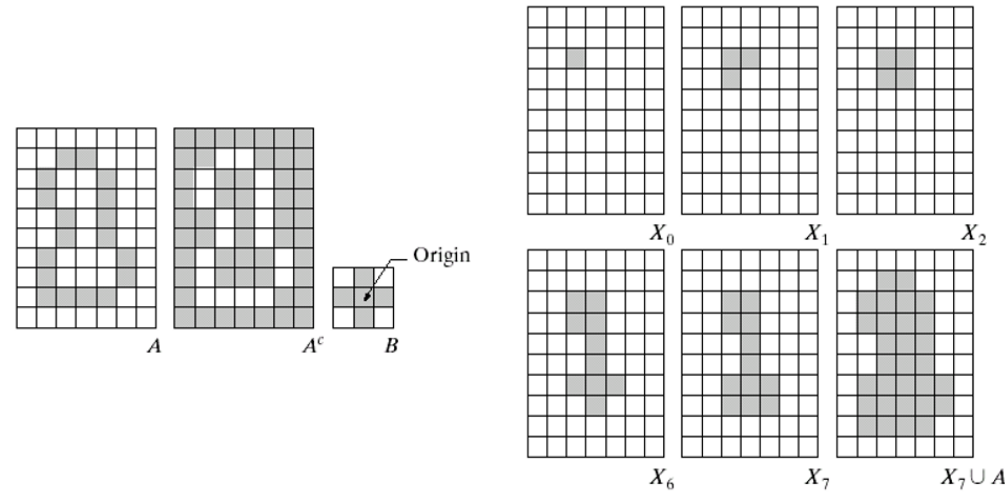
腐蚀：  
 由 B 对 A 腐蚀所产生的二值图象 E 是满足以下条件的点(x,y)的集合：如果 B 的原点平移到点(x,y)，那么 B 将完全包含于 A 中。

填洞：

Filling holes:  

$$X_k = (X_{k-1} \oplus B) \cap A^c, k = 1, 2, 3, \dots$$

$$X_0 = p, \text{ if } X_k = X_{k-1}, \text{ iteration will terminate at the } k\text{-th step. The union set of } X_k \text{ and } A \text{ contains the foreground and its boundary.}$$



开操作：  
 先腐蚀，后膨胀；用来消除小物体、在纤细点处分离物体、平滑较大物体的边界的同时并不明显改变其面积。

闭操作：  
 先膨胀，后腐蚀；用来填充物体内部细小空洞、连接邻近物体、平滑其边界的同时并不明显改变其面积。

## 27、韦伯定律在灰度图上的应用

假设连续两个灰度级之间的亮度差异就是韦伯定律中的可视临界值，那么

$$\frac{I_{max}}{I_{min}} = (1 + K_{weber})^{255}$$

$$\text{Considering } K_{weber} = 0.01 \cdots 0.02, \frac{I_{max}}{I_{min}} = 13 \cdots 156$$

The traditional display contrast:

- cathode-ray tube (CRT) :100:1
- Paper Print: 10:1

正常人眼所能看到的灰度级约为 156

## 28、可视化增强：以对数操作为例

为了增强图像的可视信息，对图像中的像素进行基于对数的操作

$$L_d = \frac{\log(L_w + 1)}{\log(L_{max} + 1)}$$

$L_d$  是显示亮度， $L_w$  是真实世界亮度， $L_{max}$  是场景中的最亮值。

这个映射能够确保不管场景的动态范围是怎么样的，其最大值都能映射到 1（白），其他的值能够比较平滑地变化。

## 29、直方图均衡化的理论与实践以及为什么没有做到真正均衡

连续性：将原图像的非均匀分布的直方图通过变换函数  $T$  修正为均匀分布的直方图，然后按均衡直方图修正原图像。图像均衡化处理后，图像的直方图是平直的，即各灰度级具有相同的出现频数。

### Assumptions:

(1) Let  $r$  and  $s$  represent the gray levels before and after the transform, and  $0 \leq r, s \leq 1$ 。

(2)  $P(r)$  and  $P(s)$  are the probability corresponding to  $r$  and  $s$ .

### Rules:

(1)  $T(r)$  is Monotonically increasing function,  $0 \leq r \leq 1$  and  $0 \leq T(r) \leq 1$ 。

(2) Inverse transform  $r = T^{-1}(s)$  is also a Monotonically increasing function。

Considering grayscale transform will not change the locations and the number of pixels, we have

$$\text{i.e. } \int_0^r P(r) dr = \int_0^s P(s) ds = \int_0^s 1 \cdot ds = s = T(r)$$

$$s = T(r) = \int_0^r P(r) dr$$



## ■Histogram equalization—find $T$ (discrete)

设一幅图像的像素总数为 $n$ ，分 $L$ 个灰度级， $n_k$ 为第 $k$ 个灰度级出现的像素数，则第 $k$ 个灰度级出现的概率为：

$$P(r_k) = n_k / n \quad (0 \leq r_k \leq 1, k = 0, 1, 2, \dots, L-1)$$

离散灰度直方图均衡化的转换公式为：

$$s_k = T(r_k) = \sum_{i=0}^k P(r_i) = \sum_{i=0}^k \frac{n_i}{n} = \frac{1}{n} \sum_{i=0}^k n_i$$

### Histogram equalization--Example

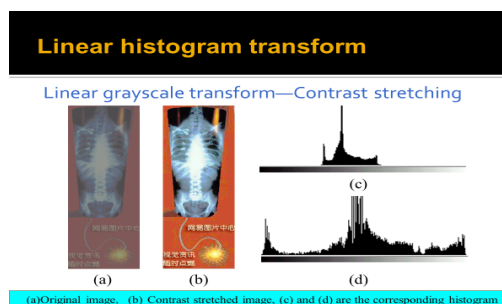
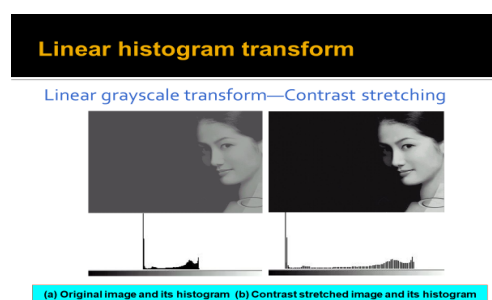
$$P(s_k) = n_k / n$$

Step 5: Compute the corresponding probability  $P(s_k)$

|            |       |       |       |       |       |       |       |       |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| $k$        | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     |
| $r_k$      | 0     | 1/7   | 2/7   | 3/7   | 4/7   | 5/7   | 6/7   | 1     |
| $n_k$      | 790   | 1023  | 850   | 656   | 329   | 245   | 122   | 81    |
| $P(r_k)$   | 0.19  | 0.25  | 0.21  | 0.16  | 0.08  | 0.06  | 0.03  | 0.02  |
| $s_k$      | 0.19  | 0.44  | 0.65  | 0.81  | 0.89  | 0.95  | 0.98  | 1.00  |
| $\sim s_k$ | 1/7   | 3/7   | 5/7   | 6/7   | 6/7   | 1     | 1     | 1     |
| $s_k$      | $s_1$ | $s_3$ | $s_5$ | $s_6$ | $s_6$ | $s_7$ | $s_7$ | $s_7$ |
| $n_k$      | 790   | 1023  | 850   | 985   |       | 448   |       |       |
| $P(s_k)$   | 0.19  | 0.25  | 0.21  | 0.24  |       | 0.11  |       |       |

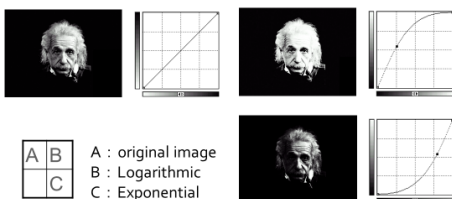
所得的  $s_k$  不可能正好等于 8 级灰度值中的某一级，因此需要就近归入某一个灰度级中。这样，相邻的多个  $s_k$  就可能落入同一个灰度级，需要将处于同一个灰度级的像素个数累加。因此，离散灰度直方图均衡化操作以后，每个灰度级处的概率密度（或像素个数）并不完全一样。直方图均衡化实质上是减少图像的灰度级以换取对比度的加大。在均衡过程中，原来的直方图上出现概率较小的灰度级被归入很少几个甚至一个灰度级中，故得不到增强。若这些灰度级所构成的图象细节比较重要，则需采用局部区域直方图均衡化处理。

30、直方图变换的一些应用



## Nonlinear histogram transform

Logarithmic transform and exponential transform



### 31、几何变换

列举简单的几何变换：平移，旋转，缩放，错切，镜像

Shear on x axis

$$\begin{cases} a(x, y) = x + d_x y \\ b(x, y) = y \end{cases}$$

Shear on y axis

$$\begin{cases} a(x, y) = x \\ b(x, y) = y + d_y x \end{cases}$$

几何变换后是否需要插值，列举一些插值的例子

行插值：按顺序寻找每一行中的空洞像素，设置其像素值与同一行中前一个像素的像素值相同。

最近邻插值：

### 最近邻插值——计算过程

为了计算几何变换后新图像中某一点 $P'$ 处的像素值，可以首先计算该几何变换的逆变换，计算出 $P'$ 所对应的原图像中的位置 $P$ 。通常情况下， $P$ 的位置不可能正好处在原图像的某一个像素位置上（即 $P$ 点的坐标通常都不会正好是整数）。寻找与 $P$ 点最接近的像素 $Q$ ，把 $Q$ 点的像素值作为新图像中 $P'$ 点的像素值。

双线性插值：

- (1) 定义双线性方程  $g(x,y)=ax+by+cxy+d$ 。
- (2) 分别将 A、B、C、D 四点的位置和灰度代入方程，得到方程组。
- (3) 解方程组，解出 a、b、c、d 四个系数。
- (4) 将 P 点的位置代入方程，得到 P 点的灰度。

### 32、Morph 的方法：

原理：让图 a 中每个像素的颜色，逐渐变成图 b 相同位置像素的颜色。

方法：根据变换的快慢，设置相应的步长，将图 a 每一点的 RGB 逐渐变成图 b 相同位置像

素的 RGB。可以选择等比或等差的方式，对于灰度图像，可以直接用等比或等差级数定义步长，使颜色从原图变到目标图。对于彩色图像，RGB 三色同时从原图变到目标图像。可以分别变化，也可考虑 RGB 的相对比例关系同时变化。

For example, if you want to produce  $n$  intermediate frames, you just use  $I_i = I_{\text{begin}} + i * \frac{I_{\text{end}} - I_{\text{begin}}}{n}$  to obtain the intermediate pixel value.

### 33、表情比例图的方法

Input: Images  $A$   $A'$   $B$

Step1: Mark feature points

Step2: For each feature point  $v_b$  in  $B$ , warp it:

$$v_{b'} = v_b + v_{a'} - v_a$$

Let  $B_g$  be the warped image of  $B$

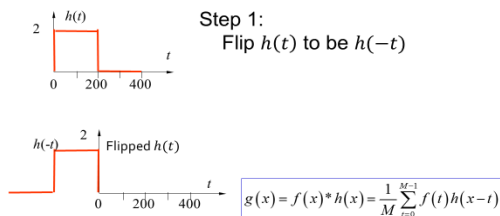
Step3: Align  $A$ ,  $A'$  with  $B_g$  by image warping.

Step4: Compute ratio image:  $\mathfrak{R} = \frac{A'}{A}$

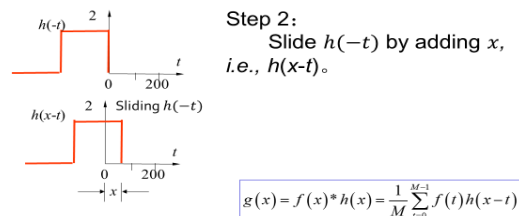
Step5:  $B' = \mathfrak{R} \cdot B_g$

### 34、一维卷积

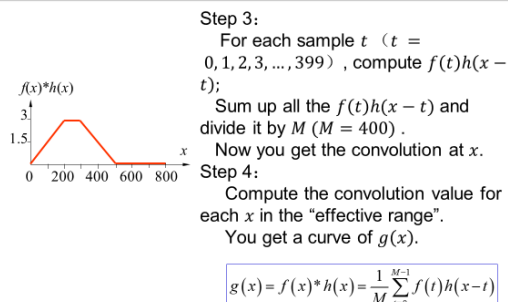
#### 1D convolution



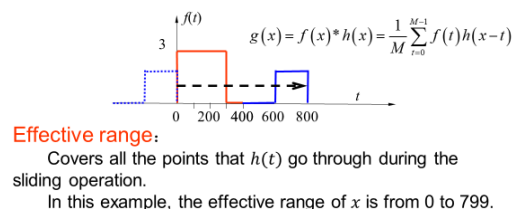
#### 1D convolution



#### 1D convolution



#### 1D convolution



### 35、卷积在图像处理中的意义（对像素加权求和）

### 36、滤波（均值滤波和中值滤波）

均值滤波：

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

中值滤波（不会变的那么糊）：

|    |    |     |     |     |  |    |    |     |     |     |
|----|----|-----|-----|-----|--|----|----|-----|-----|-----|
| 22 | 17 | 102 | 105 | 106 |  | 22 | 17 | 102 | 105 | 106 |
| 21 | 21 | 100 | 99  | 102 |  | 21 | 21 | 99  | 102 | 102 |
| 19 | 22 | 20  | 102 | 102 |  | 19 | 21 | 97  | 102 | 102 |
| 24 | 21 | 97  | 101 | 104 |  | 24 | 21 | 97  | 101 | 104 |
| 19 | 18 | 101 | 108 | 101 |  | 19 | 18 | 101 | 108 | 101 |

### 37、拉普拉斯（优点缺点）

拉普拉斯算子：

$$\begin{aligned} \nabla^2 f = & [f(x-1, y-1) + f(x, y-1) + f(x+1, y-1) \\ & + f(x-1, y) + f(x+1, y) \\ & + f(x-1, y+1) + f(x, y+1) + f(x+1, y+1)] \\ & - 8f(x, y) \end{aligned}$$

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{如果拉普拉斯掩模中心系数为负} \\ f(x, y) + \nabla^2 f(x, y) & \text{如果拉普拉斯掩模中心系数为正} \end{cases}$$

优点：锐化图像

缺点：使得噪声点更加明显

### 38、双边滤波

基本思想：

- An image has two main characteristics
  - The space domain  $S$ , which is the set of possible positions in an image. This is related to the resolution, i.e., the number of rows and columns in the image.
  - The intensity domain  $R$ , which is the set of possible pixel values. The number of bits used to represent the pixel value may vary. Common pixel representations are unsigned bytes (0 to 255) and floating point.

高斯滤波仅考虑了空间域，而双边滤波由于考虑了 intensity domain，因此具有保边的作用  
式子中每个符号的意思：

**Bilateral Filter Definition: an Additional Edge Term**

Same idea: **weighted average of pixels.**

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q$$

new:  $\frac{1}{W_p}$  (归一化因子)  
 not new:  $G_{\sigma_s}(\|p - q\|)$  (Space weight, 二维高斯核)  
 new:  $G_{\sigma_r}(\|I_p - I_q\|)$  (Intensity weight)

### 39、双边滤波的加速

## Reformulation: Summary

DigiVFX

|            |                            |     |  |
|------------|----------------------------|-----|--|
| linear:    | $(w^{bf}, i^{bf}, w^{bf})$ | $=$ | $g_{\sigma_s, \sigma_r} \otimes (wi, w)$               |
| nonlinear: | $I_p^{bf}$                 | $=$ | $\frac{w^{bf}(p, I_p) i^{bf}(p, I_p)}{w^{bf}(p, I_p)}$ |

1. Convolution in higher dimension
  - expensive but well understood (linear, FFT, etc)
2. Division and slicing
  - nonlinear but simple and pixel-wise

**Exact reformulation**

#### 40、Guided Filter (保梯度)

除了保持边两边像素灰度级的差异, 还保持梯度方向不发生变化(局部不会发生梯度逆转), 而双边滤波保留的是 difference 的绝对值

#### Conclusion

- We go from “BF” to “GF”
  - Edge-preserving filtering
  - Non-iterative
  - $O(1)$  time, fast, accurate
  - Gradient preserving
  - More generic than “smoothing”

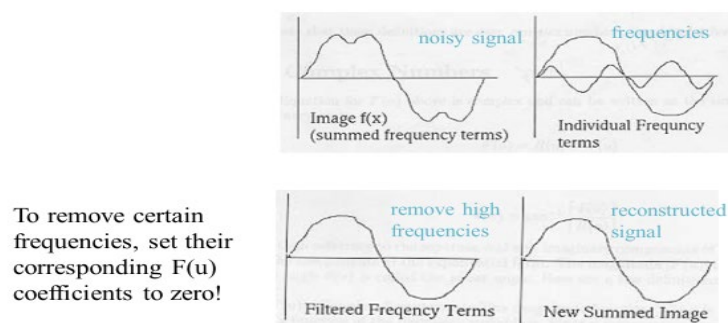
Thank you!

#### 41、sparse norm filter

$l=1$  时候为中值滤波  $l=2$  时候高斯滤波, 本质上求最小化

#### 42、用傅里叶变换解释发生了什么事情

#### Example: Removing undesirable frequencies



1. Take the FT of  $f(x)$ :  $F(f(x))$

2. Remove undesired frequencies:  $D(F(f(x)))$

3. Convert back to a signal:  $\hat{f}(x) = F^{-1}(D(F(f(x))))$

## 43、FFT

### FFT Principle

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-\frac{j2\pi kn}{N}}$$

■ Let  $W_N^{n,k} = e^{-j2\pi nk/N}$

then DFT is:  $F(k) = \sum_{n=0}^{N-1} f(n) W_N^{n,k}$

- Assuming  $N$  is the positive integral power of 2, we have

$$N = 2^H \Rightarrow N = 2M$$

$$\begin{aligned} F(k) &= \frac{1}{2M} \sum_{n=0}^{2M-1} f(n) W_{2M}^{n,k} \\ &= \frac{1}{2} \left[ \frac{1}{M} \sum_{n=0}^{M-1} f(2n) W_{2M}^{2n,k} + \frac{1}{M} \sum_{n=0}^{M-1} f(2n+1) W_{2M}^{2n+1,k} \right] \end{aligned}$$

### FFT Principle

$$W_{2M}^{2n,k} = e^{-j2\pi \cdot 2nk / 2M} = e^{-j2\pi nk / M} = W_M^{n,k}$$

$$\begin{aligned} W_{2M}^{2n+1,k} &= e^{-j2\pi (2n+1)k / 2M} = e^{-j2\pi nk / M} \cdot e^{-j2\pi k / 2M} \\ &= W_M^{n,k} \cdot W_{2M}^k \end{aligned}$$

$$F(k) = \left[ \sum_{n=0}^{M-1} f(2n) W_M^{n,k} + \sum_{n=0}^{M-1} f(2n+1) W_M^{n,k} W_{2M}^k \right]$$

Let : 
$$\begin{cases} F_e(k) = \sum_{n=0}^{M-1} f(2n) W_M^{n,k} \\ F_o(k) = \sum_{n=0}^{M-1} f(2n+1) W_M^{n,k} \end{cases} \quad k = 0, 1, \dots, M-1$$

• then :  $F(k) = [F_e(k) + F_o(k) W_{2M}^k]$

### FFT Principle

- Then how about  $k = M, M+1, \dots, 2M-1$ ?

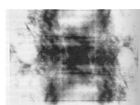
$$\begin{aligned} W_M^{n,k+M} &= e^{-j2\pi n(k+M)/M} \\ &= e^{-j2\pi nk/M} \cdot e^{-j2\pi n} \\ &= e^{-j2\pi nk/M} \\ &= W_M^{n,k} \end{aligned}$$

and  $W_{2M}^{k+M} = e^{-j2\pi (k+M)/2M} = e^{-j2\pi k/2M} \cdot e^{-j\pi} = -W_{2M}^k$

Hence  $F(k+M) = [F_e(k) - F_o(k) W_{2M}^k]$

## 44、相位和幅值的重要性（相位更重要）

### Magnitude and Phase of DFT (cont'd)



Reconstructed image using  
magnitude only  
(i.e., magnitude determines the  
contribution of each component!)



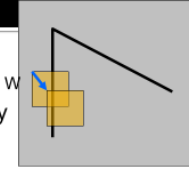
Reconstructed image using  
phase only  
(i.e., phase determines  
which components are present!)

#### 45、能量函数推导 H 矩阵的过程

### Feature detection: the math

Consider shifting the window  $W$  by  $(u, v)$

- how do the pixels in  $W$  change?
- compare each pixel before and after by summing up the squared differences
- this defines an "error" of  $E(u, v)$ :

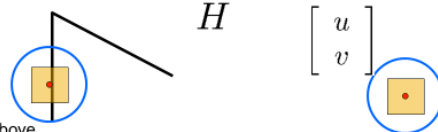


$$\begin{aligned}
 E(u, v) &= \sum_{(x,y) \in W} [I(x+u, y+v) - I(x, y)]^2 \\
 &\approx \sum_{(x,y) \in W} [I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x, y)]^2 \\
 &\approx \sum_{(x,y) \in W} \left[ [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2
 \end{aligned}$$

### Feature detection: the math

This can be rewritten:

$$E(u, v) = \sum_{(x,y) \in W} [u \ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_H \begin{bmatrix} u \\ v \end{bmatrix}$$



For the example above

- You can move the center of the yellow window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest  $E$  values?
- We can find these directions by looking at the eigenvectors of  $H$

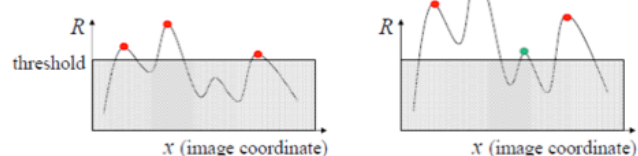
46、Harris-corner Detector 对于灰度的变化，位移的变化，旋转的变化具有不变性，但对于尺度敏感，因而提出尺度不变检测如下 47

#### ■ Invariance to image intensity change?

##### ■ Partial invariance to additive and multiplicative intensity changes

✓ Only derivatives are used  $\Rightarrow$  invariance to intensity shift  $I \rightarrow I + b$

✓ Intensity scale:  $I \rightarrow a I$

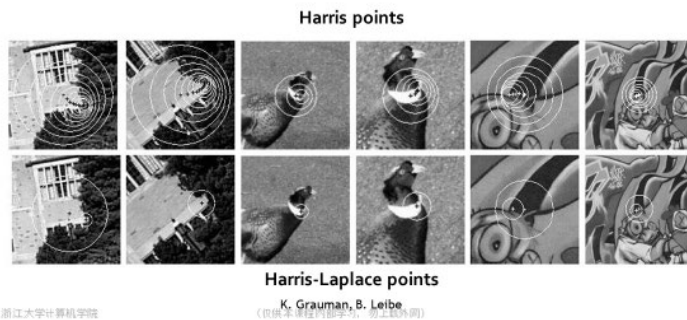




## 47、Harris-Laplace 两步

### Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian

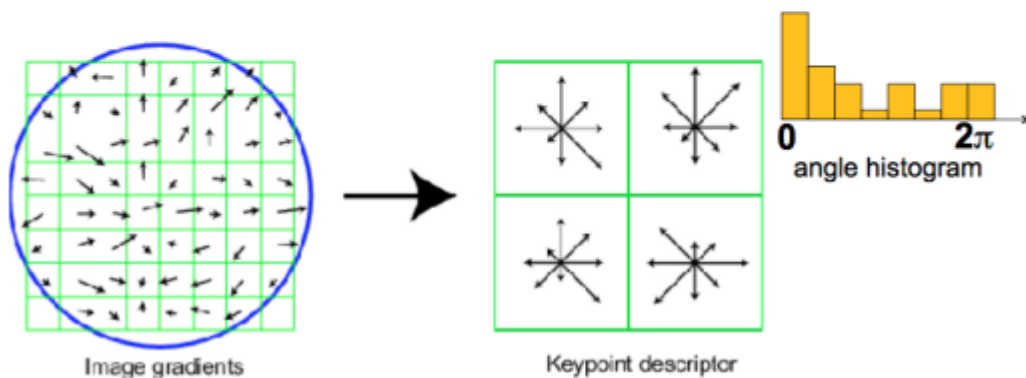


## 48、SIFT 特征的旋转不变性 (计算直方图, 投票最多的作为 dominant orientation)

- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

## 49、SIFT 的尺度不变性

1. 获得特征点
2. 计算每个像素的梯度方向
3. 抛弃梯度太小的像素
4. 为剩下的像素建立8格的直方图
5. 分为4\*4的格子, 每个格子内是直方图的统计结果, 每个特征转化为4\*4\*8=128维的高维向量, 作为descriptor, 完



第一步取 16\*16 的窗口, 第 5 步将窗口分为了 4\*4 的格子, 每一个直方图是八维向量

## 50、SIFT 的图像拼接

### 基本步骤

- Detect key points
- Build the SIFT descriptors
- Match SIFT descriptors
- Fitting the transformation
- RANSAC
- Image Blending

## RANSAC 的基本步骤

**RANSAC** [Fischler & Bolles 1981]

RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)
2. Compute transformation from seed group
3. Find *inliers* to this transformation
4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers

## 51、SIFT 的优缺点

**Conclusion of SIFT**

- Merit
  - Desired property in invariance in changes of scale, rotation, illumination, etc.
  - Highly distinctive and descriptive in local patch.
  - Especially effective in rigid object representation.
- Drawback
  - Time consuming for extraction
    - About one second in average for an image with size of 400 by 400.
  - Poor performance for un-rigid object.
    - Such as human face, animal, etc.
  - May fail to work in severe affine distortion.
    - The local patch is a circle, instead of an ellipse adjusted to the affine distortion.

## 52、SURF detectors and descriptors

### SURF algorithm

#### Interest point *detector*:

- **Compute integral image**
- **Apply 2<sup>nd</sup> derivative (approximate) filters to image**
- **Non-maximal suppression (Find local maxima in (x,y,σ) space)**
- **Quadratic interpolation**

- 1、计算积分图
- 2、计算二阶导
- 3、非极大值抑制
- 4、最小二乘插值提升精度

## SURF algorithm

Interest point **descriptor**:

- Divide window into 4x4 (16 subwindows)
- Compute Haar wavelet outputs
- Within each subwindow, compute

$$v_{subregion} = \left[ \sum dx, \sum dy, \sum |dx|, \sum |dy| \right]$$

- This yields a 64-element descriptor  
(Only implement USURF – no rotation)

1、窗口分为 4\*4，16 个 cell

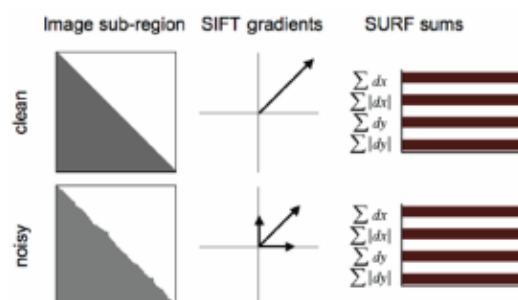
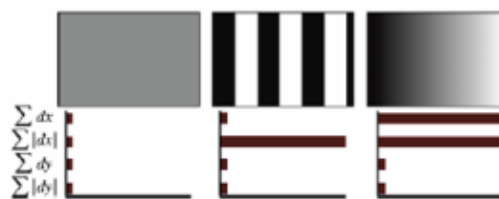
2、计算类 haar 小波特征

3、

SURF 是 64 维的向量，每一维的特征 SIFT 主要为方向，而 SURF 分别为 1 阶导和幅值，SIFT 用赋值来筛选，SURF 用了类 haar 小波特征。

53、SURF 相较于 SIFT 的优势

## Why SURF is better than SIFT



SURF 可以用于加速特征提取，SIFT 特征不好算条纹类型，以及灰度值渐变的类型，SURF 都可以计算，SIFT 在计算有噪声时可能受到干扰，SURF 抗干扰能力更强。

54、BP

## ■ Back-propagation (BP)

- Step 1: Randomly initialize weight, compute  $h_W(X)$ ;
- Step 2: Compute error  $E = (h_W(X) - y)^2$
- Step 3:  $W_k = W_{k-1} - \epsilon \frac{\partial E}{\partial W}$ , specifically,  $w_{ij}^{(k)} =$

$$w_{ij}^{(k-1)} - \epsilon \frac{\partial E}{\partial w_{ij}^{(k-1)}}$$

55、为什么要引进卷积神经网络（如何减少参数）

1、采用了局部连接，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的信息综合起来就得到了全局的信息。

2、权值共享，把每个卷积核当作一种特征提取方式，而这种方式与图像等数据的位置无关。这就意味着，对于同一个卷积核，它在一个区域提取到的特征，由于图像的特征是稀疏的，也能适用于其他区域。

3、池化，降采样，只提取一些重要的特征，经过池化层之后，模型参数会减少很多。

56、要具体了解池化的含义以及作用

57、考试第一题还考到数字图像为什么是现在主流的信息载体还是啥巴拉巴拉的，隐约记得老师说过是要点，但当时人懒就没在意。