Zhiyuan College

Physics (Zhiyuan Honor Program)

数字图像处理

2023 秋

Author: Qiheng Wang

Mail: wangq1h@sjtu.edu.cn

Date: December 20, 2023

Contents

1 数字图像处理基础

1.1 光与视觉

1.1.1 三基色

彩色三要素：

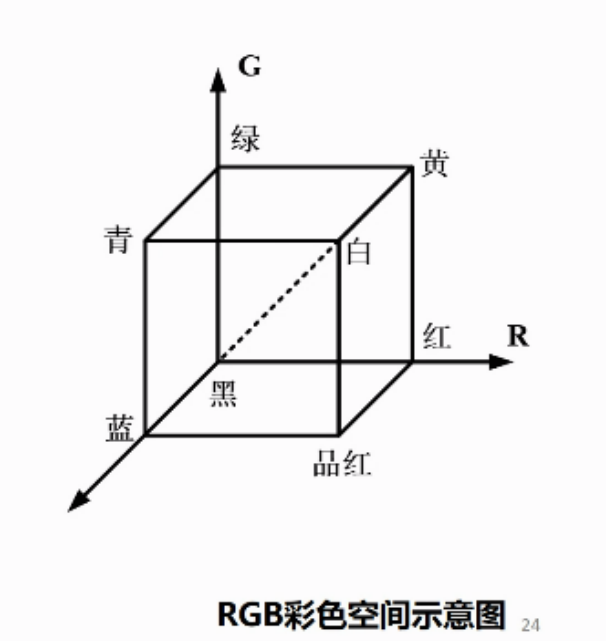
- 亮度：光强
- 色调：光谱成分
- 饱和度：彩色的深浅。与掺入白光比例相关，白光比例越高，饱和度越低

混色法：

- 加法混色
- 减法混色：青、品红、黄（CMY），相当于 RGB 的补集
- 四色打印：CMYK，加一个黑色

颜色模型：

- 面向硬件设备：RGB
- 面向彩色处理：HSI
- 印刷合电视：CMYK/YUV



RGB彩色空间示意图 24

1.2 图像获取

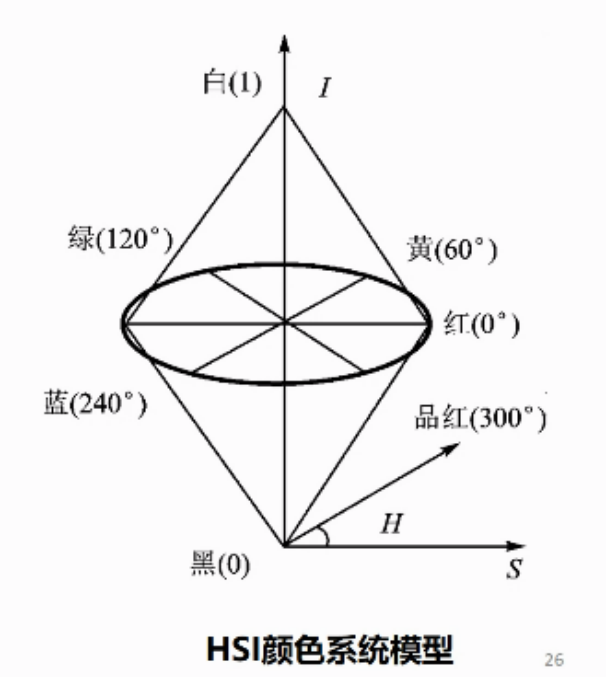
- 用一个棱镜分光，得到 RGB 三张图，成本太大
- 使用滤波，用拜尔滤镜每一个像素得到一种光，损失信息
- 使用临近像素均值法

1.3 像素

1.3.1 邻域

- 四邻域：$N_4(p)$，上下左右
- 对角邻域：$N_D(p)$
- 八邻域：$N_8(p)$

两个像素存在邻接关系，对应的为四邻接、对角邻接，八邻接。**四邻接/对角邻接必八邻**



接。

连接：在邻接的基础上，灰度值满足某种相似规则。对应有四连接，八连接和对角连接和 m-连接。

m-连接：r 和 p 在连接的定义相似集合基础上满足以下任一

- r 在 $N_4(p)$
- r 在 $N_D(p)$ 中，且集合 $N_4(p) \cap N_4(r) = \emptyset$

1.3.2 连通性

通路：连续的邻接

连通：连续的连接

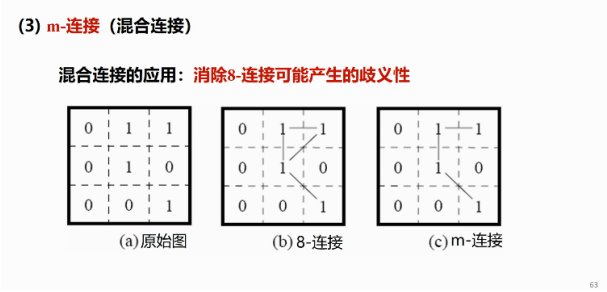


Figure 1: m-连接示意

连通域：

- 所有像素灰度级别小于或等于连通域级别
- 任意两个像素在这个连通域内连通

多连通域：和复分析相同

区域：连通域

边界：区域中邻域像素不在区域的元素集合

边沿：超过某个灰度阈值的像素点集合，图像强度的不连续性

二值图像中边界等于边沿

1.3.3 距离

欧式距离，城区距离（曼哈顿距离），棋盘距离（最大的投影距离）

2 形态学图像处理

我们所要处理的是腐蚀、膨胀、开启、闭合形态学运算。其在 MATLAB 与 OpenCV 中均有对应的函数。

2.1 数字形态学基本概念

基本思路：

- 一种邻域运算形式
- 结构单元：一种特殊的邻域，与二值图像进行特定逻辑运算
- 运算效果：大小、结构与运算性质

总结为一句话：**利用结构单元作为“探针”在图像中不断移动，收集图像的信息分析图像上各部分相互关系了解图像结构特征。**

2.2 二值形态学基本运算

2.2.1 腐蚀与膨胀

腐蚀就是用探针寻找能放下该基元的区域。

$$A \ominus B = \{x | B \in A\}$$

腐蚀运算的实质就是在目标图像中标出那些与结构元素相同的子图像的原点位置的像素。

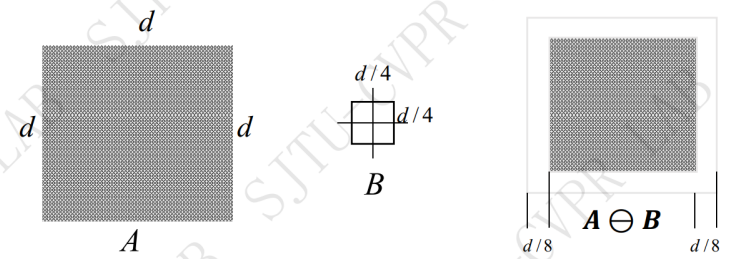
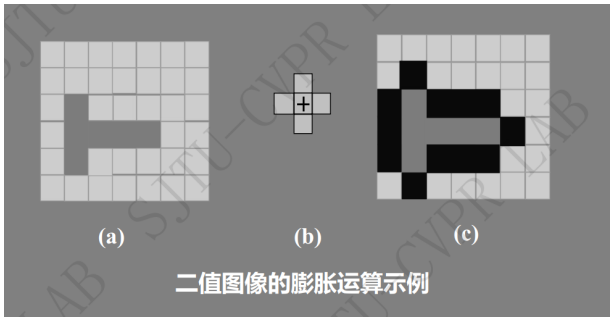


Figure 2: 腐蚀的示意

膨胀则是腐蚀的对偶运算，因此可以通过对补集的腐蚀来定义：

$$A \oplus B = \left(A^C \ominus (-B)\right)^C \tag{1}$$

文字叙述就是找到全平面上不涉及目标图像的结构元素的原点集合的补集。



2.2.2 开运算与闭运算

直观的图像是：开运算消除尖刺削弱凸起，而闭运算填补缺口削弱凹陷，均起到平滑图像的作用。

在定义了膨胀和腐蚀运算的基础上，我们可以轻松地写出开运算和闭运算的定义式。开运算为：

$$A \circ B = (A \ominus B) \oplus B \tag{2}$$

相当于先腐蚀再膨胀。

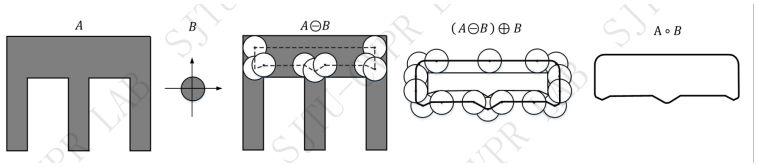


Figure 3: 开运算示意

而闭运算为：

$$A \bullet B = (A \oplus B) \ominus B \tag{3}$$

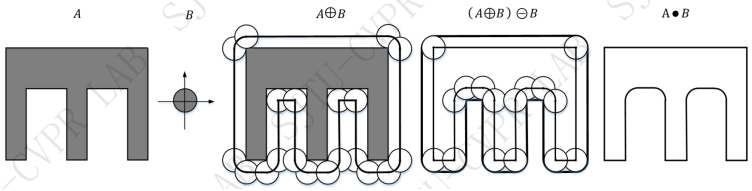


Figure 4: 闭运算示意

2.2.3 击中/击不中变换

击中/击不中变换是为了在图像的多个目标中找到特定形状的目标。这个任务分为两步，即找到图像 A 中能放下探针图像部分的位置，再找到背景 A^C 中能放下探针背景部分的位置。两者交集为最终的找到位置。其定义式为：

$$A \circledast B = (A \ominus B_1) \cap (A^C \ominus B_2) \tag{4}$$

2.3 基本二值形态学图像处理算法

2.3.1 边缘提取

基于腐蚀可以缩小目标的原理，定义原图像和腐蚀图像的差为内边界。设图像 X 的边缘为 Y，则定义式为：

$$Y = X - (X \ominus B) \tag{5}$$

但同时我们根据膨胀的原理，也可以相当于在原图像外围勾勒一圈，也就是给出外边界。定义为：

$$Y = (X \oplus B) - X \tag{6}$$

外边界和内边界的和称为形态学梯度：

$$(A \oplus B) - (A \ominus B) \tag{7}$$

2.3.2 区域填充

区域填充运算比较复杂，首先其是一个迭代操作。可以理解为每一步在已填充区域的基础上通过膨胀运算延伸，并且保证延伸区域在原图像的补集里。数学定义为：

$$X_k = (X_{k-1} \oplus B) \cap A^C \tag{8}$$

借助图像5理解：

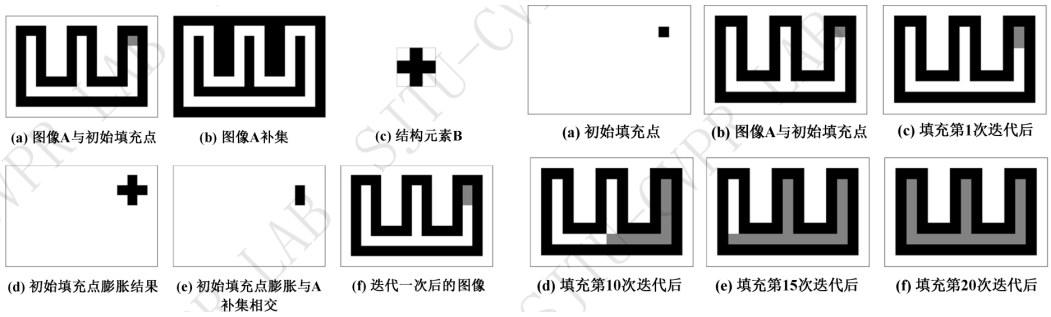


Figure 5: 区域填充示意

2.3.3 细化和厚化

细化的运算非常复杂，是用一系列结构元素去击中运算原图像，如击中则删去该点的连续过程。定义为：

$$X \otimes B = X - (X \circledast B) \tag{9a}$$

$$\{B\} = \{B_1, B_2, B_3 \dots B_n\} \tag{9b}$$

$$X \otimes \{B\} = (((X \otimes B_1) \otimes B_2) \otimes B_n) \tag{9c}$$

细化如字面意思，是将一个连通区域变换到一像素宽度。厚化可以视为细化的对偶，也

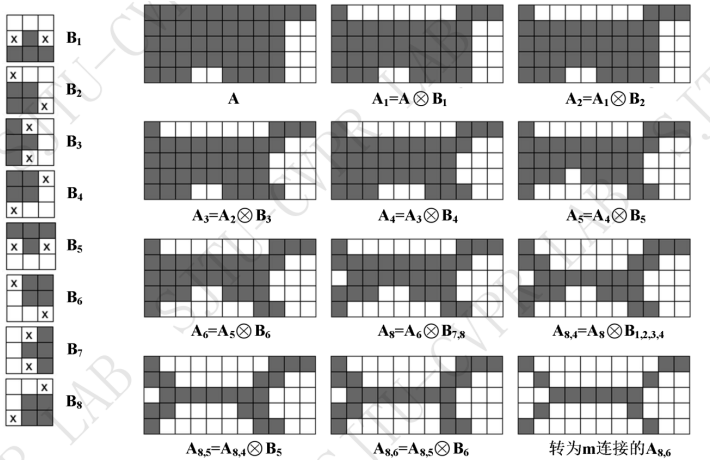


Figure 6: 细化的示意

就是说对细化图像的补集进行细化。但这个过程可能产生不连贯点，需要后处理来消除。

2.3.4 骨架提取

骨架提取是细化的一种特殊形式，其定义为：

$$S(A) = \bigcap_{i=0}^{K} S_i(A), S_i(A) = (A \ominus iB) - (A \ominus iB) \circ B \tag{10}$$

集合 A 可以通过骨架子集重建：

$$A = \bigcup_{i=0}^{K} (S_i(A) \circ iB) \tag{11}$$

2.3.5 灰度形态学基本运算

对于不是非黑即白的灰度形态学，我们需要重新定义腐蚀和膨胀运算。腐蚀运算定义为：

$$A \ominus B(s,t) = \min\{f(s+x, t+y) - b(x,y) | (s+x) \in D_f; (x,y) \in D_b\} \tag{12}$$

灰度腐蚀的效果：

- 使图像变暗（如果结构元素是正灰度）。
- 边缘部分较大灰度值点灰度值降低，边缘向灰度值更高内部收缩。
- 比结构元素小的亮区细节减弱，程度取决于结构元素的大小、幅度值及其周边灰度梯度。

灰度腐蚀的实质就是逐点计算该点局部范围内与结构元素中对应点的灰度值的差，以最小值为该点腐蚀的结果。

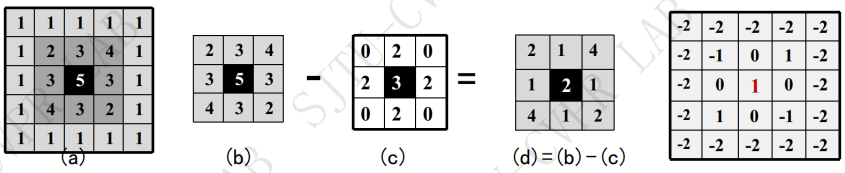
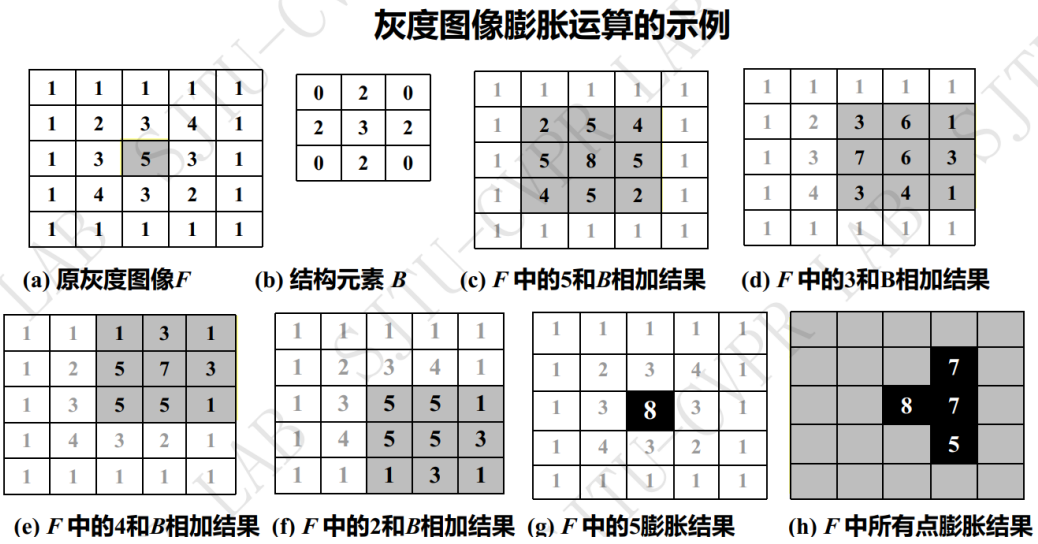


Figure 7: 灰度腐蚀示意

灰值膨胀则是保留 $f + b$ 的最大值。f 是原图像对应区域，b 是结构元素对应区域。定义

式为：

$$(f \oplus b)(s,t) = \max\{f(s-x,t-y) + b(x,y) | (s-x) \in D_f; (x,y) \in D_b\} \tag{13}$$



3 图像变换与频域滤波

3.1 傅里叶变换

Definition 3.1.1: 傅里叶变换

对于函数 $f(x)$，其傅里叶变换为

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} \, \mathrm{d}x \tag{14}$$

其中 u 为频率，$F(u)$ 为频率为 u 的正弦波在 $f(x)$ 中的分量。

因此对于图像 $f(x,y)$，其遵循二维离散傅里叶变换：

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-j2\pi(ux/M+vy/N)} \tag{15}$$

这实际上是用不同频率不同相位不同幅值的平面正弦波叠加来表示图像。

二维离散傅里叶变换具有如下性质：

- 平移和旋转协变：$f(r,\theta+\theta_0)\Leftrightarrow F(\omega,\phi+\theta_0)$
- 周期性：无限周期。$f(x,y)=f(x+k_1M,y+k_2N)\Leftrightarrow F(u,v)=F(u+k_1,v+k_2),k_1,k_2\in\mathbb{Z}$
- 可分性：可以视为正交一维傅里叶变换乘积。

3.2 离散余弦变换

由于傅里叶变换不可避免地需要涉及复数，因此在计算机中实现起来比较困难。离散余弦变换 (DCT) 能够达到相同功能但数学上只涉及实数，因此更适合计算机实现。这个变换基于一下数学事实：

$$F(u,v)=\frac{\sqrt{2}}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \tag{16}$$

其反变换为：

$$f(x,y)=\frac{\sqrt{2}}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \tag{17}$$

DCT 变换相比 DFT 变换具有更好的频域能量聚焦度，因此在图像压缩中有广泛应用。

在 JPEG 算法中，输入图像被分为 8×8 的块，每个块进行 DCT 变换，然后对变换后的系数进行量化，最后进行熵编码。保留其中的低频系数，舍弃高频系数，可以达到压缩

的效果。这一过程就是让每一模块的 DCT 系数乘以模板。

3.3 沃尔什-哈达玛变换

前两种变换都是正弦型变换，而沃尔什-哈达玛变换 (WHT) 是一种正交变换，其变换基时方波的各种变形，具有更高的计算速度。

Walsh 变换是由 $+1$ 和 -1 组成的方波的线性组合，当 $N = 2^n$ 时，其一维变换基为：

$$g(x,u) = \frac{1}{N}\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)} \tag{18}$$

其中，$b_i(x)$ 为 x 的二进制表示的第 i 位。我们还可以用矩阵来表示变换核，对于 $N = 2, 4, 8$，其变换核为：

$$\begin{aligned}
W_2 &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\
W_4 &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\
W_8 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}
\end{aligned} \tag{19}$$

对于二维情况，变换方式就会显得比较复杂。不过沃尔什变换是可分的。我们总能写成：

$$W(u,v) = \frac{1}{N^2} G \cdot f \cdot G \tag{20}$$

哈达玛变换是沃尔什变换的一种特殊情况，其变换核具有简单递推关系。其一维定义为：

$$H(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{p-1} b_i(x) b_i(u)} \tag{21}$$

Hadamard 变换核为：

$$\begin{aligned} H_2 &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ H_N &= \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix} \end{aligned} \tag{22}$$

WHT 易于模拟但很难分析，在图像数据压缩、滤波、编码中有广泛应用。

3.4 频域图像滤波与增强

变换增强的一般步骤是：图像 → 变换 → 滤波 → 反变换 → 增强图像。常用方法是：通带滤波、同态滤波。

3.4.1 频域平滑

首先对输入的图像做 DFT 得到变换域中的频谱，然后对频谱进行平滑，最后做反变换得到平滑后的图像。

图像中，噪声或物体边缘处灰度变化剧烈，对应傅立叶频谱的高频分量，物体内部灰度分布均匀，变化平稳，对应傅立叶频谱的低频分量。因此可以用频率域低通滤波法去除或削弱图像的高频成分，以使噪声得到消除或抑制，从而实现图像平滑。

常用的频域平滑滤波器分为低通和高通。低通滤波器有：理想低通滤波器 ILPF、巴特沃斯低通滤波器 BLPF、指数低通滤波器 ELPF、梯形低通滤波器 TLPF、高斯低通滤波器 GLPF。对应的变换函数为：

$$
\begin{aligned}
H_{ILPF}(u,v) &= \begin{cases} 1 & \text{if} \quad D(u,v) \le D_0 \\ 0 & \text{if} \quad D(u,v) > D_0 \end{cases} \\
H_{BLPF}(u,v) &= \frac{1}{1+k[D(u,v)/D_0]^{2n}}, k = 1, \sqrt{2}-1 \\
H_{ELPF}(u,v) &= e^{-kD^2(u,v)/2D_0^2}, k = 1, \ln\sqrt{2} \\
H_{TLPF}(u,v) &= \begin{cases} 1-\frac{D-D_1}{D_2-D_1} & \text{if} \quad D_1 \le D(u,v) \le D_2 \\ 1 & \text{if} \quad D(u,v) < D_1 \\ 0 & \text{if} \quad D(u,v) > D_2 \end{cases} \\
H_{GLPF}(u,v) &= e^{-D^2(u,v)/2D_0^2}
\end{aligned}
\tag{23}
$$

上式中 $D(u,v)$ 为欧式距离。ILPF 容易出现图像模糊，振铃效果。BLPF 和 ELPF 能够有效抑制振铃效应，但是会出现边缘模糊。TLPF 能够有效抑制振铃效应，但是会出现边缘模糊。GLPF 能够有效抑制振铃效应，且不会出现边缘模糊。

高通滤波器让高频顺利通过，得到高通图像，然后按前面锐化公式进行处理，可以使图像高频加强，边缘或线条变得更清楚，从而实现图像的锐化。

高通滤波器一般是低通滤波器的对偶，比如高斯高通滤波器 GHPF：

$$
H_{GHPF}(u,v) = 1 - e^{-D^2(u,v)/2D_0^2} \tag{24}
$$

此外还有带通或带阻滤波器。目的是对特定频率的信号进行增强或抑制。

3.5 图像的同态增晰

压缩照度分量的灰度范围或频域上消弱照度分量的频谱分量。增强反射分量的对比度或频域上加大反射频谱成分，使暗区细节增强，并保留亮区图像细节。

同态滤波器的基本思想是将图像分解为照度分量和反射分量，然后对照度分量进行低通滤波，对反射分量进行高通滤波，最后将两者相乘得到增强后的图像。

同态滤波器的变换函数为：

$$H(u,v)=\gamma_H\left[1-e^{-c\left(\frac{D^2(u,v)}{D_0^2}\right)}\right]\left[1-e^{-\frac{D^2(u,v)}{D_1^2}}\right]+\gamma_L \tag{25}$$

其中，$D(u,v)$ 为欧式距离，D_0, D_1 为截止频率，c 为控制滤波器斜率的常数，γ_H, γ_L 为增益常数。

4 图像压缩与编码

4.1 图像冗余

图像冗余是指图像中存在的不必要的信息，包括：

- 编码冗余：由于编码方式的不同，同一图像的编码长度不同。
- 像素间冗余：相邻像素之间的相关性。
- 视觉心理冗余：人眼对图像的感知不是很敏感。

特别说明一下编码冗余。编码某一图像时，表示不同灰度级像素需要的比特数可能不同，用平均比特数来表示该图像中这种编码方式对每个像素所需的平均比特数。定义第 r_k 个灰度级出现的概率为 p_k，则该图像的平均比特数为：

$$L_{avg}=\sum_{k=0}^{L-1}p_k l_k \tag{26}$$

其中 l_k 为第 k 个灰度级的比特数。如果用 l_k 位二进制数来表示第 k 个灰度级，则 $l_k=\lceil\log_2 L\rceil$。

不同的编码方式可能有不同的平均比特数，引入相对编码冗余和绝对编码冗余。

Definition 4.1.1: 相对编码冗余

相对编码冗余为：

$$R_{rel} = \frac{L_{avg} - H}{H} \tag{27}$$

其中 H 为熵，L_{avg} 为平均比特数。

Definition 4.1.2: 绝对编码冗余

绝对编码冗余为：

$$R_{abs} = L_{avg} - H \tag{28}$$

其中 H 为熵，L_{avg} 为平均比特数。

4.2 图像压缩

图像压缩的目的是减少图像的冗余，从而减少存储空间和传输带宽。

4.2.1 图像保真度与质量

图像压缩编码中解码图像与原始图像可能不完全相同。需要有对信息损失的测度来描述解码图像相对于原始图像的质量损失程度，这些测度一般称为保真度准则。通常的准则分为客观保真度准则和主观保真度准则。

- 客观保真度准则：用数学公式来描述解码图像与原始图像的差别。
- 主观保真度准则：用人眼对解码图像的主观感受来描述解码图像与原始图像的差别。

客观保真度准则常见两种：均方误差 MSE 和均方信噪比。其定义分别是：

$$
\begin{aligned}
MSE &= \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2 \right)^{\frac{1}{2}} \\
PSNR &= 10 \log_{10} \frac{L^2}{MSE}
\end{aligned}
\tag{29}
$$

其中，L 为图像的最大灰度级，M, N 为图像的行数和列数，$\hat{f}(x,y)$ 为解码图像，$f(x,y)$ 为原始图像。MSE 的定义为：

$$
MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2 \tag{30}
$$

4.3 行程编码

行程编码是一种最简单的，在某些场合是非常有效的一种无损压缩编码方法。行程编码的主要思路是将一个相同值的连续串用一个代表值和串长来代替。通过改变图像的描述方式，来实现图像的压缩。实质是消除了像素间冗余。

行程编码总位数 = 行程数 ×(像素灰度位数 + 串长位数)。

对于数字图像，同一幅图像某些连续的区域颜色相同，同一扫描行中许多连续的像素都具有同样的颜色值，可以对其进行行程编码。

相较于灰度图像与彩色图像，二值图像由于像素值数量更少，更有利于行程编码。

行程编码对传输差错很敏感，如果其中一位符号发生错误，就会影响整个编码序列的正确性，使行程编码无法还原为原始数据。

4.3.1 分变长行程编码

分变长行程编码是一种改进的行程编码方法，其主要思想是将行程编码中的串长位数改为变长编码，从而减少总位数。需要增加标志位来标记串长位数的结束，这样就可以将串长位数改为变长编码，从而减少总位数。

一般分变长行程编码的还原方法是从符号串左端开始往右搜索，遇到第一个 0 时停下来，计算这个 0 的前面有几个“1”。设“1”的个数为 K，则在 0 后面读 $K+2$ 个符号，这 $K+2$ 个符号所表示的二进制数加上 1 的值就是第 1 个行程的长度。

4.3.2 二维行程编码

二维行程编码是一种将行程编码推广到二维的方法，其主要思想是利用像素间二维信息的相关性，采用某种扫描路径遍历所有像素点，获得像素点之间的相邻关系后，按照一维行程编码方式进行编码。如图所示：

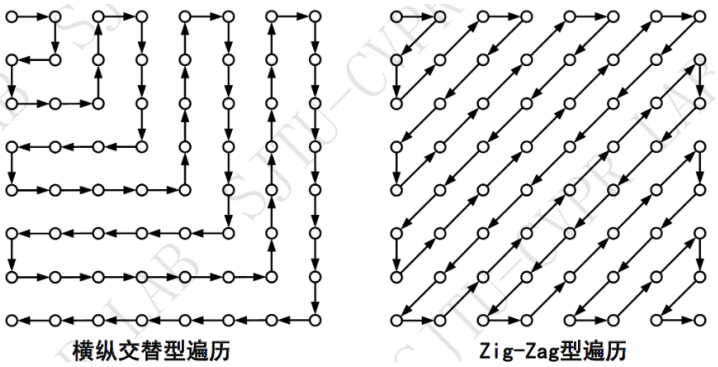


Figure 8: 二维行程编码

4.4 Huffman 编码

Huffman 编码是一种熵编码。首先定义图像的熵：

Theorem 4.4.1: 图像熵

图像熵定义为：

$$H = -\sum_{k=0}^{L-1} p_k \log_2 p_k \tag{31}$$

其中 p_k 为第 k 个灰度级出现的概率。

熵编码的效率为：

$$\eta = \frac{H}{L_{avg}} \tag{32}$$

根据信息论信源编码理论，可以证明 $R \geq H$。我们可以设计出某种无损编码方式，使得 R 尽可能接近 H，这样就可以达到最大的压缩比。

Huffman 编码是一种熵编码，其主要思想是根据信源符号出现的概率，赋予不同的编码。比如，出现概率越大的符号，其编码越短。这样就可以达到最大的压缩比。编码步骤为：

1. 求出灰度分布直方图
2. 根据概率大小，对灰度级进行排序
3. 从概率最小的两个灰度级开始，将其合并为一个新的灰度级，其概率为两者之和
4. 重复步骤 3，直到只剩下一个灰度级
5. 从最后一个灰度级开始，根据其父节点的值，赋予编码
6. 重复步骤 5，直到所有灰度级都赋予编码

其中，步骤 3 到 5 构建了一棵二叉树，称为 Huffman 树。

4.5 算术编码

算术编码是一种熵编码，其主要思想是从整个符号序列出发，采取递推形式进行连续编码。

信源符号和码字之间的对应关系并不存在，用某个 0 1 的实数区间表示整个信源符号序列编码的信息，用该实数区间最短的二进制码作为编码输出。

随着符号序列中的符号数量的增加，用来代表它的区间减小，而用来表示区间所需的信息单位的数量变大。

算术编码图示：

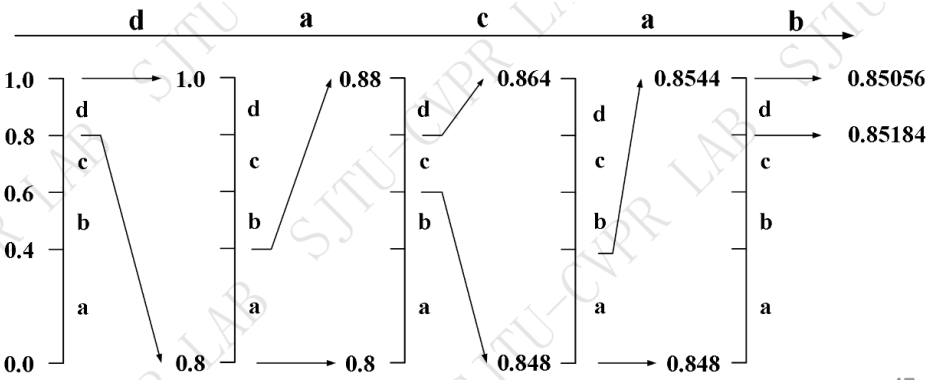


Figure 9: 算术编码

4.6 预测编码

预测编码主要思想是利用像素间的相关性，通过预测来减少冗余。

离散信号之间存在关联性，利用前面一个或多个信号预测下一个信号，对实际值和预测值的差（预测误差）进行编码。如果预测比较准确，误差就会很小，可以用比较少的比特进行编码，达到压缩数据的目的。

预测编码的性能决定于预测器的性能，最佳预测器就是在某一准则下使预测编码的性能达到最佳的预测器。常用的一些准则有：误差均方值最小准则、零（无）误差概率最大准则、误差平均分布熵最小准则等。

最佳预测器的结构与准则和信源的统计特性有关。对于平稳高斯信源，三种准则等价，

而且最佳预测器是线性预测器。对于非高斯信源，非线性预测器可以提供更好的性能，但是寻找和实现最佳的非线性预测器比较困难。

预测编码的局限性在于它认为样本之间的统计依存关系仅影响样本分布的均值而不影响分布的形状。除高斯信源外，一般信源不能满足这一条件。预测编码的最大优点在于实现方便，且对大部分实际信源相当有效，所以预测编码在实际中有广泛应用。

预测编码的一般步骤为：

1. 建立一个数学模型
2. 利用以往的样本对模型进行估计
3. 利用模型对下一个样本进行预测
4. 用预测值和实际值的差作为预测误差，对预测误差进行编码
5. 重复步骤 3 和 4，直到所有样本都编码完毕

第 n 个符号 X_n 的熵满足：

$$H(x_n) \geq H(x_n|x_{n-1}) \geq H(x_n|x_{n-1}, x_{n-2}) \geq \cdots \geq H(x_n|x_{n-1}, \cdots, x_1) \tag{33}$$

实际信源所含的实际熵，即为极限熵：

$$H_\infty = \lim_{n\to\infty} H(x_n|x_{n-1}, \cdots, x_1) \tag{34}$$

所以参与预测的符号越多，预测就越准确，该信源的不确定性就越小，数码率（即数字信号产生和传输的速率）就可以降低。

预测编码的分类有：

- 线性预测编码
- 非线性预测编码

- 帧内预测编码：只利用当前帧的信息
- 帧间预测编码：根据不同帧的信息进行预测
- 自适应预测编码：预测器和量化器参数按图像局部特性自适应变化

线性预测是一种无损压缩编码。最佳线性预测器是一种线性滤波器，其输出为预测值，输入为预测误差。最佳线性预测器的系数是根据最小均方误差准则确定的。

此外还有一种有损预测编码 DPCM。DPCM 是在无损线性预测编码的基础上加入一个量化器，步骤为：

1. 用最佳线性预测器对信源进行预测
2. 用预测值和实际值的差作为预测误差，对预测误差进行量化
3. 重复步骤 1 和 2，直到所有样本都编码完毕
4. 误差为：

$$\Delta = e_k - e_k^* \tag{35}$$

第三种线性预测模型是有损的自适应差分脉冲编码调制 ADPCM，其原理图为：

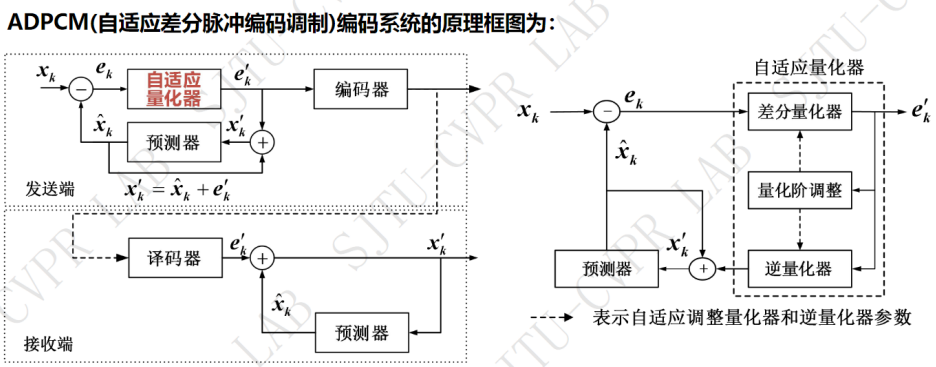


Figure 10: ADPCM 原理图

4.7 DCT 变换编码

无损压缩可以保证接收方获得的信息与发送方相同，但其压缩率有限。采用忽略视觉不敏感的部分进行有损压缩是提高压缩率的一条好的途径。

变换编码将原始数据变换到另一表示空间，使数据在新的空间上尽可能相互独立，能量更集中。

离散余弦变换（DCT）将空间数据变成频率数据，利用人的视觉对高频信息（的变化）不敏感和对不同频带数据感知特征不一样等特点，可以对多媒体数据进行压缩。

编解码示意图为：变换过程为：

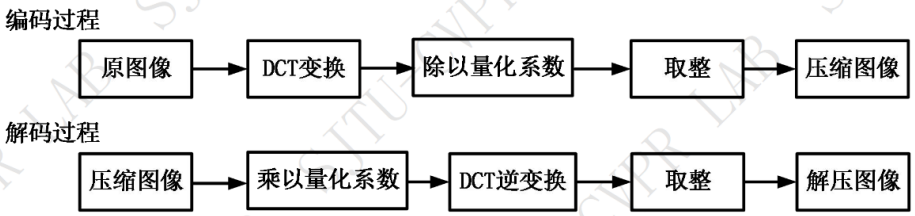


Figure 11: DCT 编解码示意图

1. 将图像分成 8×8 的小块
2. 对每个小块进行 DCT 变换
3. 对变换后的系数进行量化，一般还需要取整
4. 对量化后的系数进行编码

4.8 JPEG 图像压缩

JPEG 标准分两种：无损预测算法和基于 DCT 的有损编码算法。

4.8.1 无损预测算法

基于差分脉冲编码调制，源图像数据经过预测器后用熵编码器编码，解码时用解码器解码后经过反预测器得到原始数据。

4.8.2 基于 DCT 的有损编码算法

JPEG 压缩编码的基本步骤如图：

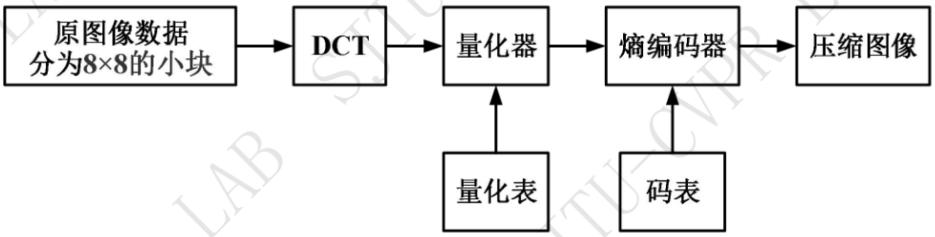


Figure 12: JPEG 压缩编码的基本步骤

5 图像增强

图像增强实质上分为两大类：空域增强和频域增强。频域增强实际就是第三章所研究的各种方法：高通、低通和同态滤波。空域增强则是对图像的像素进行操作，包括点运算和模板运算。

5.1 点运算

点运算是指对图像的每个像素进行操作，操作的结果只与该像素的灰度值有关。点运算的基本形式为：

$$B(x,y) = f\left(A(x,y)\right) \tag{36}$$

其中 $A(x,y)$ 为原图像，$B(x,y)$ 为增强后的图像，f 为点运算函数。点运算函数可以是线性的，也可以是非线性的。点运算不会改变图像内像素点之间的空间关系。

常用的两种点运算方式为灰度变换或者借助 $f(x,y)$ 的直方图进行变换。基于点操作的增强方法称为灰度变换。常用的灰度变换有：

- 图像反转
- 分段线性变换：对图像的灰度值进行分段线性变换，可以实现对图像的局部增强，相对抑制不感兴趣的区域。
- 非线性变换：常见的有 γ 校正：$s = cr^{\gamma}$，其中 c 为常数，γ 为常数，r 为原图像的灰度值，s 为增强后的灰度值。这个 γ 矫正可以实现对图像的局部增强：
  - $\gamma > 1$ 时，对低灰度值区域进行增强，对高灰度值区域进行抑制。
  - $\gamma < 1$ 时，对高灰度值区域进行增强，对低灰度值区域进行抑制。
- 直方图修正法

5.1.1 直方图修正法

直方图具有以下性质：

- 位置信息丢失：直方图只反映了图像的灰度分布，而没有反映图像的空间信息。
- 直方图叠加
- 直方图范围动态

直方图修正法包括直方图均衡化和直方图规定化。直方图均衡化的基本思想是将给定图像的直方图分布改造成均匀分布的直方图。其变换函数应满足如下两个条件：

- 单调性：$r_1 < r_2 \Rightarrow s_1 < s_2$

- 有界性：$0 \leq s_k \leq L-1$

可以发现累计分布函数 CDF 满足上述两个条件。因此，可以通过 CDF 来实现直方图均衡化：

$$t_k = EH(s_k) = \sum_{i=0}^{k} \frac{n_i}{n} = \sum_{i=0}^{k} p_s(s_i) \tag{37}$$

理论地研究这个灰度的映射关系。

$$p_s(s)\,\mathrm{d}s = p_{T(s)}(T(s))\,\mathrm{d}(T(s)) = p_t(t)\,\mathrm{d}t \tag{38}$$

因为我们的理想目标是灰度均匀分布，所以有：$p_t(\omega) = \frac{1}{L-1}$。因此在离散的情况下得到：

$$t = T(s) = (L-1)\sum_{i=0}^{s} p_s(s_i) \tag{39}$$

直方图均衡化的步骤为：

1. 计算原图像的直方图 $p_s(s)$
2. 计算原图像的累计分布函数 $t = T(s)$
3. 取整扩展：$k_t = \lfloor t \rfloor$
4. 确定映射关系：$s_k = k_t$
5. 计算均衡化后的图像：$s = T^{-1}(t)$

由于以上过程是全局性质的，所以会导致图像的局部细节信息丢失。针对局部细节信息丢失的问题，发展了自适应直方图均衡化 (AHE)。AHE 一般是以点为中心，以邻域为窗口进行直方图均衡化。子块一般互有重叠，这样可以避免边缘处的突变。

在对一张明部和暗部对比度跨度过大的图像进行直方图均衡化时，会导致局部对比度经过均衡后向另一端过分偏移。而当某个区域包含的像素值非常相似时，直方图均衡化会导致该区域的像素值分布变得非常尖锐。为了改进这种情况，提出了对比度限制的直方

图均衡化 (CLAHE)。CLAHE 的基本思想是将超出阈值的像素值进行均匀地分配到其他区域。

另一种直方图修正方法就是规定化。规定化的意思是将一幅图像的直方图变换成规定的直方图。映射方式一般是单映射规则，即从小到大依次找到能使下式成立的最小的 k 和 l：

$$\left|\sum_{i=0}^{k} P_s(s_i) - \sum_{j=0}^{l} P_\mu(u_j)\right| \tag{40}$$

其中 $P_s(s_i)$ 为原图像的直方图，$P_\mu(u_j)$ 为规定的直方图。总结而言：

- 直方图均衡化：自动增强，效果不易控制，偏向全局
- 直方图规定化：有选择地增强，须给定需要的直方图，会具有特定增强的结果

5.2 模板运算

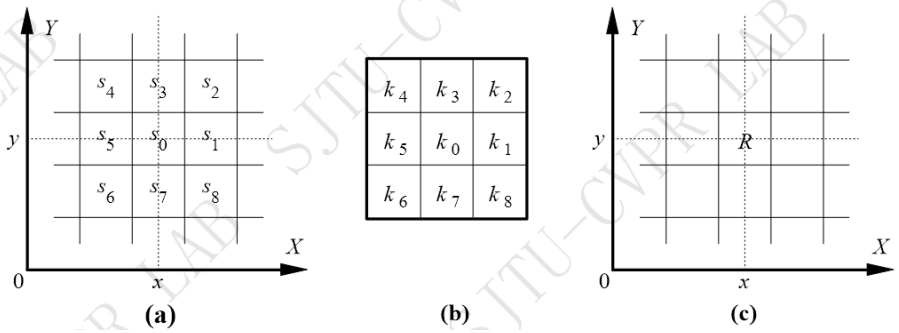
模板也称为窗口，可以看作是一幅尺寸为 $n \times n$(n 一般为奇数，通常远远小于图像尺寸)的小矩阵，其各个位置上的值称为系数值，不同的系数值及其组合决定了不同模板的功能。

模板的作用主要是将某个像素的灰度值与其相邻的像素的灰度值通过函数结合起来，赋予该像素。

函数的形式可以是线性的，也可以是非线性的，运算方式可以是卷积或者排序等。这种处理图像像素的方法常称为滤波，模板也就相当于滤波器。

5.2.1 模板卷积

卷积是一种线性运算，示意图为：图中：



$$R = k_0 s_0 + k_1 s_1 + k_2 s_2 + k_3 s_3 + k_4 s_4 + k_5 s_5 + k_6 s_6 + k_7 s_7 + k_8 s_8 \tag{41}$$

其中 R 为卷积结果，k_i 为模板的系数值，s_i 为模板对应位置的像素值。

5.2.2 模板排序

模板排序是利用模板来获得输入图像中与模板尺寸大小相一致的子图像并将其中像素按照某种顺序（一般是幅值大小）排序的运算。与模板卷积类似，其基本步骤如下：

1. 将模板在输入图像上覆盖，模板中心与输入图像中的某个像素对齐
2. 读取模板所覆盖的输入图像对应像素的灰度值
3. 将这些灰度值按照某种顺序排序
4. 根据运算的目的，取排序后的灰度值中的某个值作为输出图像中对应像素的灰度值
5. 将模板在输入图像上移动一个像素，重复上述过程，直到模板覆盖输入图像的所有像素

5.2.3 图像边界处的模板运算

在图像边界处，模板运算的处理方式有以下几种：

- 忽略边界处的像素，对较大图像的处理是有用的
- 扩展边界处的像素，即将边界处的像素值扩展到模板之外
  - 0 扩展：将边界处的像素值扩展为 0，会导致边界处的不连续
  - 将这些新增像素的灰度值赋为其在原图像的 4-邻接像素的灰度值，四个角上新增像素的灰度值赋为原图像中 8-邻接像素的灰度值
  - 周期性边界条件
  - 利用外插等其他规则

5.2.4 线性滤波

首先介绍两种空域滤波的模板：平滑滤波和锐化滤波。

平滑滤波用于模糊处理和降低噪声。平滑滤波可以减弱图像中的高频分量但不影响低频分量，降低局部灰度起伏，平滑图像。平滑滤波经常用于预处理任务中，如在大目标提取之前去除图像中的一些琐碎细节，以及桥接直线或曲线的缝隙。

一种实际的方法是邻域平均。邻域平均的基本思想是用邻域内像素的平均值来代替该像素的灰度值。邻域平均的模板为：

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \tag{42}$$

邻域平均的缺点是会使图像变得模糊，且对图像中的细节信息进行了破坏。

有平权平均，也有加权平均。比如如果认为距离中心像素越近的像素对中心像素的影响越大，那么可以使用一种加权平均的模板：

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} \tag{43}$$

高斯平均滤波器是一种特殊的加权平均滤波器，其模板上的系数值是服从高斯分布的。一个 5×5 的高斯平均滤波器模板为：

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \tag{44}$$

一个二维高斯卷积可以分解为顺序的两个一维高斯卷积，这样可以大大减少计算量。例如：

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{45}$$

锐化处理的主要目的是突出灰度的过渡部分。锐化滤波能够减弱图像中的低频分量而不影响高频分量，忽略整体对比度和平均灰度值，强调图像反差，使得图像边缘更加明显。锐化滤波的用途多种多样，应用范围有电子印刷和医学成像，工业检测和军事系统的制导等。

同样介绍几种线性锐化滤波器。首先是拉普拉斯滤波器，其模板为：

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{46}$$

拉普拉斯滤波器的缺点是会使图像中的噪声增强，因此一般不单独使用，而是与其他滤波器结合使用。

另一种是高频提升滤波器。设原图像为 $f(x, y)$，平滑后的图像为 $g(x, y)$，定义非锐化掩膜为：

$$h(x, y) = f(x, y) - g(x, y) \tag{47}$$

高频提升滤波即为：

$$h_b(x, y) = (A - 1)f(x, y) + h(x, y) \tag{48}$$

其中 A 为增益因子，$h_b(x,y)$ 为高频提升滤波后的图像。

5.2.5 非线性滤波

线性滤波器常常不能区分图像中有用的信息和无用的噪声，因为线性滤波可以被描述为原始图像的傅里叶变换和滤波模板的傅里叶变换相乘，结果是在每个频率处，有用信息和噪声都乘以相同的因子，图像信噪比没有被改变。

解决上面问题的一种方法就是引入非线性滤波。非线性滤波表示的原始数据与滤波结果是一种逻辑关系，即用逻辑运算实现，如最大-最小滤波、中值滤波等，是通过比较一定邻域内的灰度值大小来实现的。

中值滤波是一种低通滤波器。所谓中值滤波是把以某点 (x,y) 为中心的小窗口内的所有像素的灰度按从大到小的顺序排列，将中间值作为 (x,y) 处的灰度值 (若窗口中有偶数个像素，则取两个中间值的平均)。用公式表示即：

$$g(x,y) = \text{median}\{f(x,y), f(x,y), \cdots, f(x,y)\} \tag{49}$$

推广到二维，是对模板所覆盖的图像区域的像素灰度值进行排序，取得排序位于中间位置的灰度值作为模板中心点对应的灰度值：

$$g(x,y) = \text{median}_{(s,t)\in S_{xy}}\{f(s,t)\} \tag{50}$$

我们还可以设定规则，使得并非所有模板中的元素都参与排序。通常情况下，统计排序的像素个数不超过 9 13 个，有实验表明，超过 13 个时计算量的增加比噪声消除的改善更加明显，所以一般采用稀疏矩阵模板来减少计算量。

此外还有基于梯度的锐化滤波。这里介绍 roberts 算子。用绝对值来近似平方根和平方和，用差分表示梯度。roberts 算子是一种二阶微分算子，其模板为：

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{51}$$

由于偶数尺寸模板很难实现，因为没有对称中心。因此我们还有 sobel 算子，其模板为：

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{52}$$

其依据的便是三点中心差分公式。

最后介绍最大-最小滤波器。最大-最小锐化滤波是一种将最大值滤波和最小值滤波结合使用的图像增强技术，可以锐化模糊的边缘并让模糊的目标清晰起来。这种方法可以迭代进行，在每次迭代中，将一个模板覆盖区域里的中心像素值与该区域里的最大值和最小值进行比较，然后将中心像素值用与之较接近的极值 (最值) 替换。变换方式定义为：

$$g(x,y) = \begin{cases} \max_{(s,t)\in S_{xy}}\{f(s,t)\} & \text{if} \quad f(x,y) = \min_{(s,t)\in S_{xy}}\{f(s,t)\} \\ \min_{(s,t)\in S_{xy}}\{f(s,t)\} & \text{if} \quad f(x,y) = \max_{(s,t)\in S_{xy}}\{f(s,t)\} \\ f(x,y) & \text{otherwise} \end{cases} \tag{53}$$

5.3 彩色图像增强

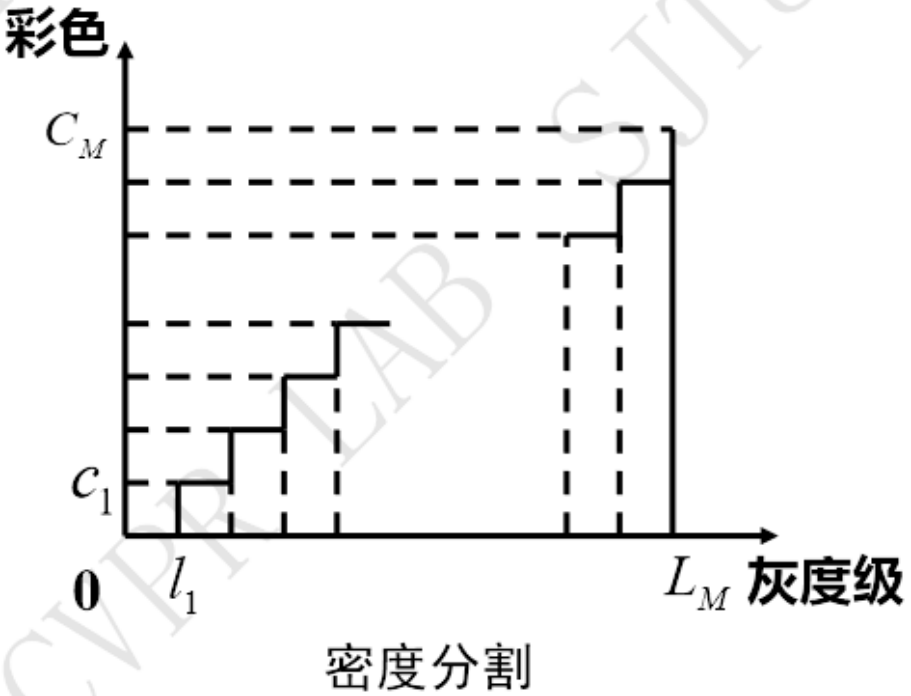
在图像外理中常可借助彩色来处理图像以得到对人眼来说增强了的视觉效果。一般来说，彩色图像增强有两大类: 伪彩色增强和真彩色增强。

伪彩色增强是把一幅黑白图像的不同灰度级映射为一幅彩色图像。伪彩色技术早期在遥感图像处理中得到广泛的应用，后来又大量地应用于医学图像处理中。

真彩色增强实际上是映射一副彩色图像为另一幅彩色图像，从而达到增强对对比度的目的。

5.3.1 伪彩色增强

伪彩色变换有两大类：密度切割法和灰度级映射法。此外还有滤波法。示意图分别列于其下：

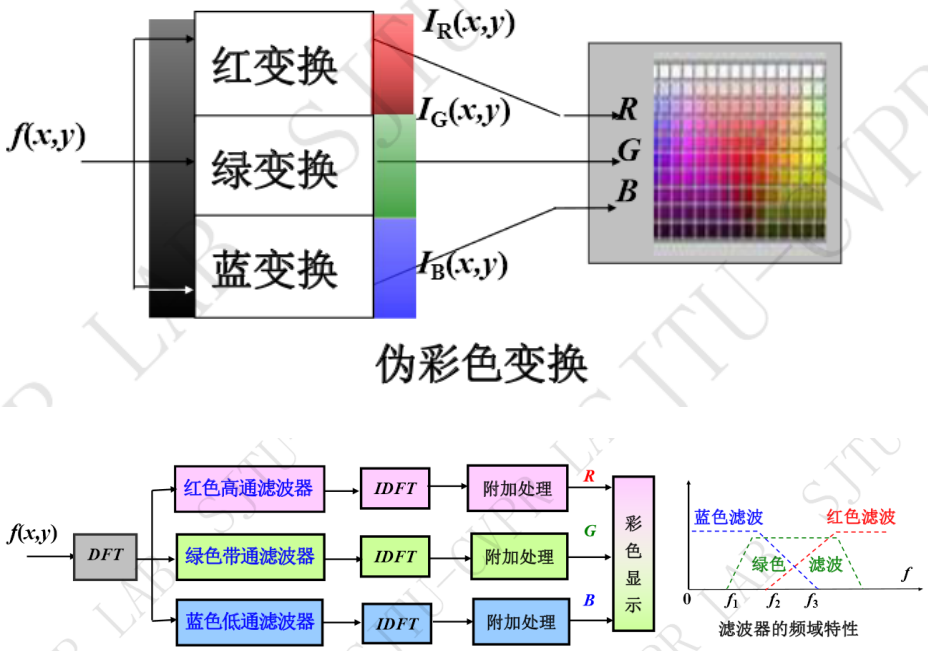


密度分割

5.3.2 真彩色增强

尽管对 R、G、B 各分量直接使用对灰度图增强的方法可以增加图中的可视细节亮度，但得到的增强图中的色调有可能完全没有意义。这是因为在增强图中对应同一个像素的 R、G、B 这 3 个分量都发生了改变，它们的相对数值与原来不同，从而导致图像颜色的较大改变。

在彩色图像模型 HSI 中，亮度分量与色度分量是分开的；色调和饱和度的概念与人的感知是紧密相关的。如果将 RGB 图转化为 HSI 图，亮度分量和色度分量就分开了。前面讨



论的对灰度图增强的方法都可以使用。一种简便常用的真彩色增强方法的基本步骤为：

1. 将 RGB 图转化为 HSI 图
2. 对亮度分量进行增强
3. 将增强后的亮度分量与原来的色调和饱和度分量组合成新的 HSI 图
4. 将新的 HSI 图转化为 RGB 图

6 图像分割

6.1 图像分割概述

6.2 边缘检测

6.3 基于阈值的分割

6.4 基于区域的分割

6.5 基于形态学分水岭的分割

分水岭分割算法是一种基于拓扑理论的数学形态学分割方法。分水岭算法的基本思想是：将图像看成拓扑平面，图像中的每个点对应于拓扑平面中的高度，局部极小值影响的区域形象称为集水盆地，边界称为分水岭。

操作中，假想水从最低点上涨，每当水将淹没分水岭时修建“水坝”，当水涨到完全没过拓扑平面时，水坝组成了分割。

分水岭算法的基本步骤如下：

- 设 $C(n-1)=C_{n-1}(M_1)\cap C_{n-1}(M_2)$，$q$ 为水位上涨至 n 时，M_1, M_2 汇水形成的连通区域。
- 对 $C_{n-1}(M_1)$ 和 $C_{n-1}(M_2)$ 进行形态学膨胀，但保证
  - 膨胀点属于 q
  - 膨胀点不使两区域连通
- 最后得到的属于 q 但不属于 $C(n-1)$ 及其膨胀区域的点集即为分水岭

分水岭算法并不能保证效果具有普遍性。因此有三种改进后的算法，分别是

- 基于距离变换的分水岭算法
- 基于梯度的分水岭算法
- 基于标记的分水岭算法

首先介绍一下距离变换：

Definition 6.5.1: 距离变换

对于二值图像 A，定义 A 中每个像素点到 A 中最近的非零像素点的距离为该像素点的欧式距离，然后取代灰度。

基于梯度的分水岭算法是先求梯度图，然后对梯度图像使用分水岭算法。由于梯度图像可能仍含有噪声，所以最好先进行平滑。

基于标记的分水岭算法是先对梯度图像进行标记，然后对标记后的图像使用分水岭算法。

- 内部标记符标记出目标区域满足某些条件的区域，这些条件一般是先验的（比如人眼观察出）。通常是灰度局部最小区域。
- 外部标记符通常标记一个连通区域，可以通过对内部标记图距离变换后分水岭分割得到
- 通过在梯度幅值图像中，将对应内外标记符的区域设置为最小值的方法对梯度图像预处理，再进行分水岭分割。

7 图像描述

图像描述方法需要满足两个要求：

- 尺度不变性
- 旋转不变性
- 光照不变性

7.1 边界描述

有两种主要的边界描述方法：

- 链码
- 傅里叶分析子

7.1.1 链码

链码是一种用于表示边界的方法，其基本思想是用一系列特定长度和方向的线段来逼近边界。

首先定义方向，有四向链码和八向链码两种。但如果直接拟合边界容易产生很长的码串，或者受到噪声影响。常用的改进方法是用外切网格重新采样，然后用四向链码或八向链码进行拟合。

链码的优点是简单，但是缺点是对噪声敏感。不过其具有尺度不变性和旋转不变性。

7.1.2 傅里叶描述子

傅里叶描述子是一种用于表示边界的方法，其基本思想是

7.2 纹理描述

纹理特征分为自然纹理与人工纹理，主要区别在于自然纹理的统计特性是随机的，而人工纹理的统计特性是有规律的。

标志三要素：

- 局部序列性在更大区域重复
- 序列元素非随机排列
- 平均有大致相同的结构尺寸

7.2.1 灰度差分统计

灰度差分统计又称为一阶统计法，通过计算灰度差分的直方图来反映纹理特征。

7.2.2 灰度共生矩阵

灰度差分统计很难考虑不同方向上的纹理特征，因此提出了灰度共生矩阵。灰度共生矩阵又称为联合概率统计法，通过计算灰度共生矩阵来反映纹理特征。

灰度共生阵 $p(d,\phi)$ 定义为：在距离 d 和方向 ϕ 上，灰度级为 i 的像素与灰度级为 j 的像素相邻的概率。

由于直接处理像素，所以没有尺度不变性。

7.3 形状上下文算法

以 Shapecontext 算法为例。其步骤首先是 Canny 边缘检测：

1. 图像平滑去噪
2. 计算梯度幅值和方向
3. 非极大值抑制
4. 重置梯度模值
5. 极值点方向重置
6. 双阈值检测

接下来要进行合适地采样轮廓点。既要有足够多的点，又要保证采样点的均匀性。

然后是基于 Shapecontext 的匹配算法：

1. 计算 Shapecontext
2. 计算匹配代价
3. 用匈牙利算法求解最优匹配

8 图像检测

8.1 V-J 检测算法

9 图像特征提取的深度网络

9.1 深度神经网络的宏观架构

深度神经网络的宏观架构为：

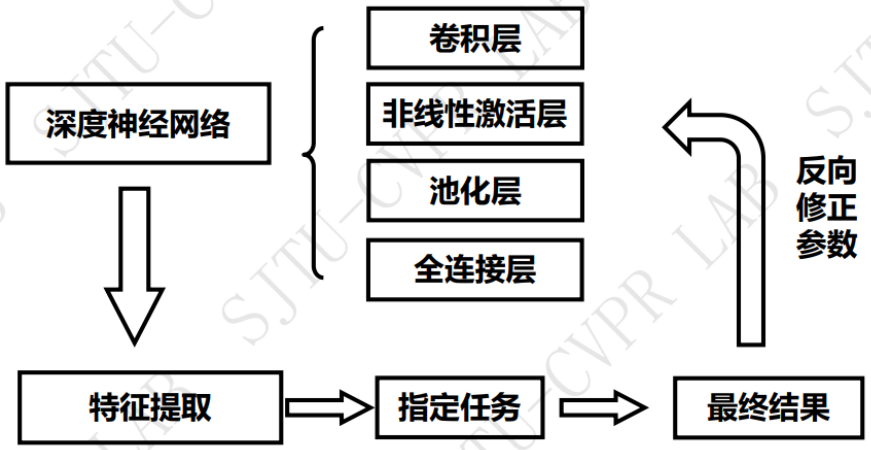


Figure 13: 深度神经网络的宏观架构

9.1.1 卷积层

卷积层的作用是提取图像的局部特征，其具有如下概念：

- 边界填充：为了保证卷积后的图像大小不变，需要在原图像的边界填充一圈像素。在 AlexNet 中使用 0 填充。

- 卷积核：一个小矩阵。
- 卷积核的步长：卷积核每次移动的距离。
- 特征图：卷积层的输出。
- 感受野：卷积神经网络每一层输出的 feature map（特征图）上的像素在原始图像上映射的区域大小.
- 共享权值：同一个隐藏层（即输出的特征图）中的所有神经元（即输出像素）都是通过同一个卷积核卷积得到，以检测同一个特征在图像的各个位置是否存在，将从输入层到隐藏层的这种映射称为特征映射。该特征映射的权重，即卷积核参数，被称为共享权值。其“共享”的含义源自于：输入图像中的每个局部感受野均被同样的卷积核进行卷积。为了进行图像识别，通常需要不止一个的特征映射，因此一个完整的卷积层包含若干个不同的特征映射，也就是若干个不同的卷积核（后面称为“通道”）

卷积层的运行过程为卷积核滑过整个图像，在每个位置上通过求取卷积核与原图对应区域（被卷积核覆盖的区域）之间的乘积和来得到卷积结果

比如：

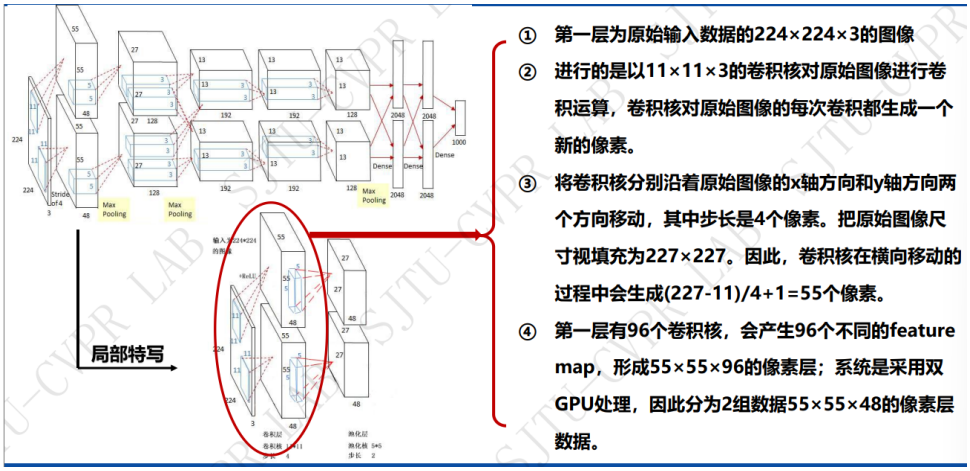


Figure 14: 卷积层的运行过程

9.1.2 非线性激活层

非线性激活层是在卷积处理之后，通过激活函数进行非线性变换，从而得到输出信号。最后输出的信号具有 $f(W \cdot \vec{x} + b)$ 的形式，其中 f 为非线性激活函数，W 为卷积核的权重矩阵，b 为偏置 bias。

非线性函数一般由有：

- sigmoid 函数：$f(x) = \frac{1}{1+e^{-x}}$igmoid 函数为值域在 0 到 1 之间的光滑函数。当需要观察输入信号数值上微小的变化时，与阶梯函数相比，平滑函数 (比如 Sigmoid 函数) 的表现更好。
- 线性整流函数（Rectified Linear Unit，ReLU）：$f(x) = max(0, x)$，ReLU 函数在 x>0 时，梯度为 1，而在 x<0 时，梯度为 0。这样的特性使得 ReLU 函数在反向传播时不会出现梯度消失的问题。由于在 ReLU 函数进行激活之前，卷积核卷积所得到的结果有正有负：正的结果说明感受野与卷积核相关性较大，倾向于找到特征，因此被 ReLU 函数所激活；而负的结果说明卷积核并未找到特征，该结果可以忽略不计的，因此用此函数进行舍弃。
- Softmax 函数：$f(x) = \frac{e^x}{\sum_{i=1}^{n} e^{x_i}}$，Softmax 函数将输入信号变换为概率分布，其输出值在 0 到 1 之间，且所有输出值的和为 1。Softmax 函数常用于多分类问题中，将神经网络的输出转换为各个类别的概率。

9.1.3 池化层

化层的主要目的是通过降采样的方式，在不影响图像质量的情况下，压缩图像、减少参数。简单来说，假设特征图像大小为 3*3，池化层采用 Max Pooling 方法，大小为 2*2，步长为 1 ，池化的步长和卷积相同，也是池化核每两次运算间移动的距离。那么图像的尺寸就会从 3*3 变为 2*2。

通常来说，池化方法有以下两种：

- Max Pooling：取池化核中的最大值作为输出。
- Average Pooling：取池化核中的平均值作为输出。

其实每一个卷积核可以看作一个特征提取器，不同的卷积核负责提取不同的特征，假如第一个卷积核能够提取出“垂直”方向的特征，第二个卷积核能够提取出“水平”方向的特征，那么池化对其进行 Max Pooling 池化操作后，提取出的是真正能够识别到特征的数值，其余被舍弃的数值，这些数值对于提取特定的特征并没有特别大的帮助。但在进行后续计算中，减小了特征图的尺寸，从而减少了参数，达到减小计算量却不损失效果的目的。

9.1.4 全连接层

到全连接层（fully-connected）这一步，其实一个完整的“卷积部分”就算完成了，如果想要叠加层数，一般也是叠加“Conv-MaxPooing”组合层，因为叠加卷积层可以通过提取更多更抽象的特征来达到更好的检测效果。

在进入到全连接层之前，需要把特征图展开成一维向量，输出到 Flatten 层（拉伸成一维向量），然后把 Flatten 层的输出输入到全连接层里，对其进行分类。Flatten 层：将所有 Feature map 拉伸成一维，可以大大减少特征位置对分类带来的影响。也就是说无论在画面哪个位置提取到特征，在进行全连接以后均可被检测到，增强进行检测任务时的鲁棒性。

9.1.5 反向传播算法

卷积深度神经网络如 CNN、FCN 等结构中包含两类参数：超参数和可调整参数。其中超参数指需要预先设定的初始化参数，如网络层数、激活函数、卷积核大小等等。可调整参数指的是在模型训练过程中被不断调整的参数，主要指隐藏层的权重和偏置项等，可调整参数直接决定了模型输出结果的精度。因此深度神经网络模型训练的目的是为了

得到最佳的模型参数组合，最常用的模型训练方法是反向传播算法（Back Propagation），又称 BP 算法。

BP 算法的基本思想是：首先随机初始化模型参数，然后通过前向传播算法计算模型的输出结果，再通过反向传播算法计算模型的梯度，最后通过梯度下降算法更新模型参数，如此反复迭代，直到模型收敛。BP 算法的流程是：

- 随机初始化模型参数
- 通过前向传播算法计算模型输出结果
- 通过反向传播算法计算模型梯度
- 通过梯度下降算法更新模型参数
- 重复步骤 2-4，直到模型收敛

反向传播过程与梯度下降过程如图：

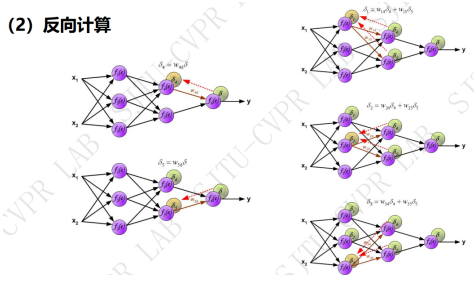


Figure 15: 反向传播过程

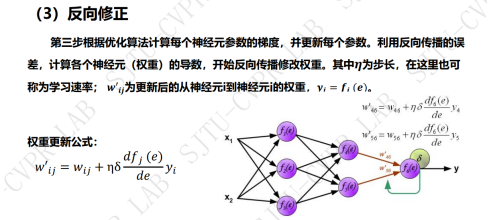


Figure 16: 梯度下降过程

9.2 VGGNet

VGG 网络的最大特点，就是它在 AlexNet 网络基础上的改进：将卷积核全部替换为 3×3（另有：1×1），从而开启了小卷积核的深度网络模型时代。

堆叠 3×3 卷积核的原理与作用有以下几点：首先可以增大感受野，其本质是使用 2 个 3×3 的卷积核堆叠等价于 1 个 5×5 的卷积核；3 个 3×3 的卷积核堆叠等价于 1 个 7×7 的卷积核，如图所示。假设输入、输出通道均为 K 个通道时，几个小卷积核（3×3）卷积

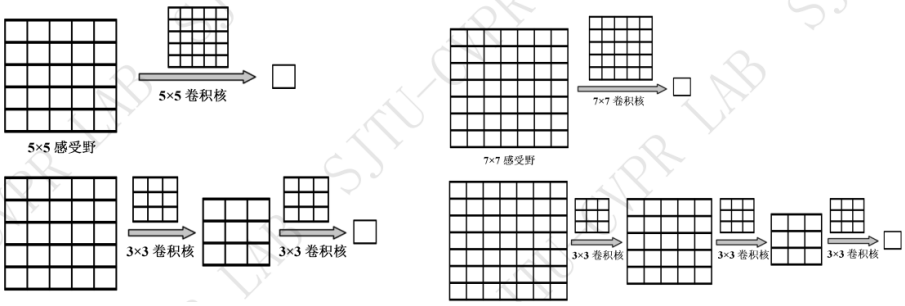


Figure 17: 堆叠 3×3 卷积核的原理

层的组合比一个大卷积核（5×5 或 7×7）卷积层好，并且可以减少训练参数。

此外，与 AlexNet 有尺寸为 3x3 的池化核不同的是，VGG 全部改为使用尺寸为 2×2 的池化核。池化层数增加，采用较小尺寸池化核依旧可以有比较好的降采样效果。

而且，VGG 网络的通道数增多。VGG 网络第一层的通道数为 64，后面每层都进行了翻倍，最多到 512 个通道。通道数的增加，意味着有更多不同的卷积核对图像输入进行处理，使得更多的特征信息可以被提取出来。

VGG 网络增加非线性激活函数，增加特征抽象能力，在全连接层也增加了 ReLU 函数进行激活这一步骤，即对所有隐藏层都使用了 ReLU 非线性激活函数进行激活。VGG19 的结构类似于 VGG16，性能略好于 VGG16，但 VGG19 需要消耗更多资源，因此实际应用中 VGG16 的使用率更高。由于 VGG-16 网络结构十分简单，并且很适合迁移学习，很多深度网络都是在 VGG16 网络的基础上进行改进，获得广泛使用。

9.3 GoogLeNet

一般来说，提升网络性能最直接的办法就是增加网络深度和宽度，深度指网络层次数量、宽度指神经元数量。但是，增加网络深度和宽度会导致网络参数数量的急剧增加，

从而导致网络训练时间的大幅度增加，同时也会导致梯度弥散或者爆炸。

一般来说解决这些问题的方法就是在增加网络深度和宽度的同时。而为了减少参数，将全连接变成稀疏连接是一种不错的思路。稀疏连接是受到神经科学中的现象启发：在神经科学中，研究人员发现每个细胞只对视觉区域中的一个极小部分敏感，而对其他部分区域则可以做到“视而不见”。因此，研究人员在设计网络结构时尝试通过减少不必要的节点间的相互连接关系，从而减少参数量。但是研究人员在实现的过程中发现，全连接变成稀疏连接后实际计算量并不会有质的提升，因为大部分硬件是针对密集矩阵计算优化的。稀疏矩阵虽然数据量少，但是计算所消耗的时间却很难减少。

大量的文献表明可以将稀疏矩阵聚类为较为密集的子矩阵来提高计算性能，就如人类的大脑是可以看作是神经元的重复堆积。因此，研究人员提出了 Inception 模块，将稀疏矩阵聚类为较为密集的子矩阵，从而提高计算性能。

GoogleNet 的完整结构为：Inception 模块可以看作是一个有 4 个分支的小型网络，每个

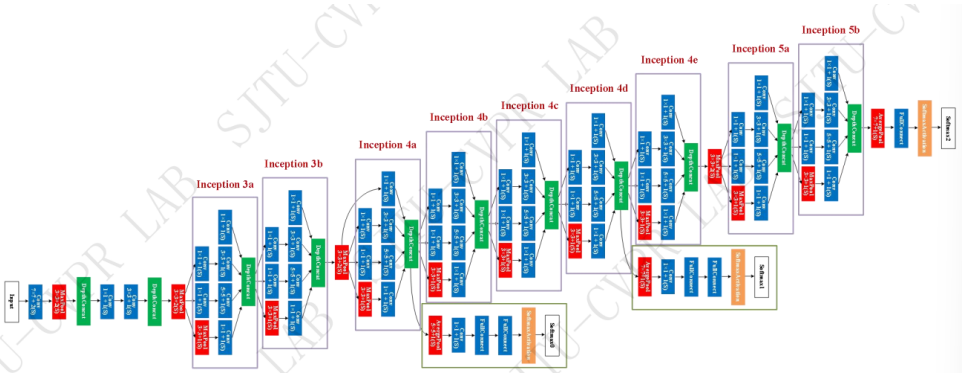


Figure 18: GoogleNet 的完整结构

分支都是一个卷积层，但是卷积核的尺寸不同，分别为 1×1、3×3、5×5，以及一个最大池化层。这样的设计可以使得网络在不同尺度下都能够提取到特征，从而提高网络的泛化能力。比如，一种可能的 Inception 模块如图所示：

其输出可能为：

此外，GoogleNet 还加入了 Dorpout 操作。Dropout 是作为训练深度神经网络的一种减少过拟合现象出现的方法。在每个训练批次中，通过忽略一定的特征检测器（如让一半的

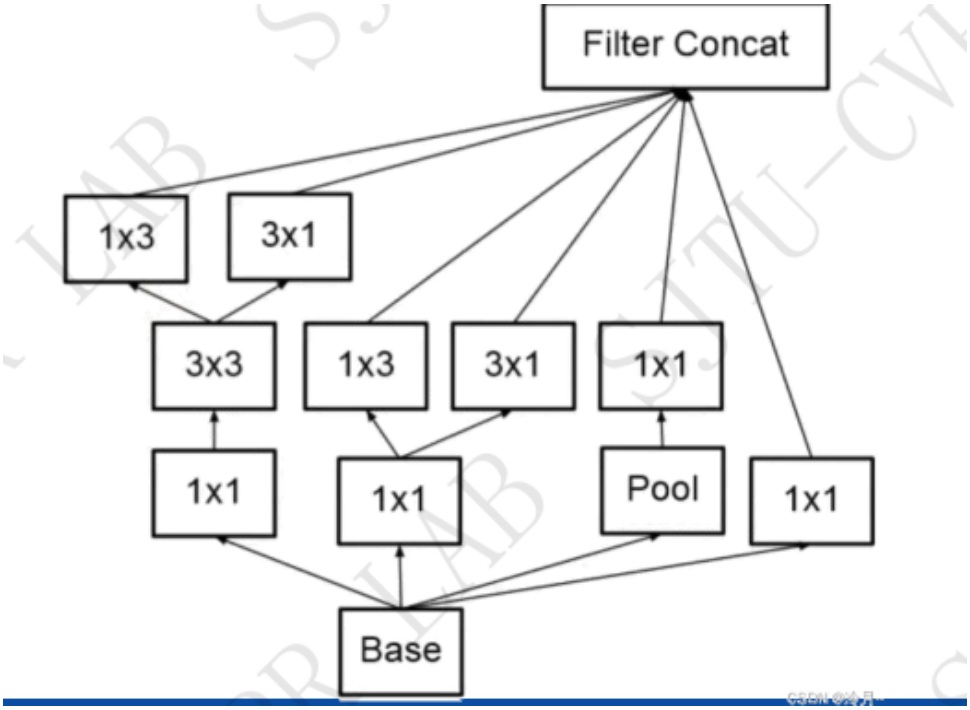


Figure 19: Inception 模块

隐藏层节点值为 0），可以明显地减少过拟合现象。这种方式可以减少特征检测器（隐藏层节点）间的相互作用，也就是指某些检测器依赖其他检测器才能发挥作用。简而言之就是：网络在前向传播的时候，让某个神经元的激活值以一定的概率 p 停止工作，这样可以使模型泛化性更强，因为它不会太依赖某些局部的特征。为了避免梯度弥散，网络额外增加了 2 个辅助的 Softmax 层用于向前传导梯度，也就是右边两个标注浅绿色框的辅助分类器。Softmax 和梯度弥散前文中讲述过，而这里的辅助分类器也是解决梯度弥散问题的方案之一。辅助分类器是将中间某一层的输出用作分类，并按一个较小的权重（如 0.3）加到最终分类结果中，这样相当于做了模型融合，同时给网络增加了反向传播的梯度信号，也提供了额外的正则化，对于整个网络的训练很有帮助。而在实际测试的时候，这两个额外的 Softmax 分类器会被去掉，仅帮助训练使用。

GoogleNet 沿用了 AlexNet 网络中率先提出并使用的局部响应归一化层（Local Response Normalization, LRN），在之前并没有进行详细介绍，因为在与 GoogleNet 同年参赛的网络，也是上一节所介绍的 VGGNet 中，经实验表明，在 ILSVRC 数据集上使用该层并不能提升网络的表现，反而会提升内存消耗和运算时间。因此在这里作简要介绍。在 AlexNet 模型中首次提出 LRN 这一概念，目的是进行局部对比度增强，以便使局部最大

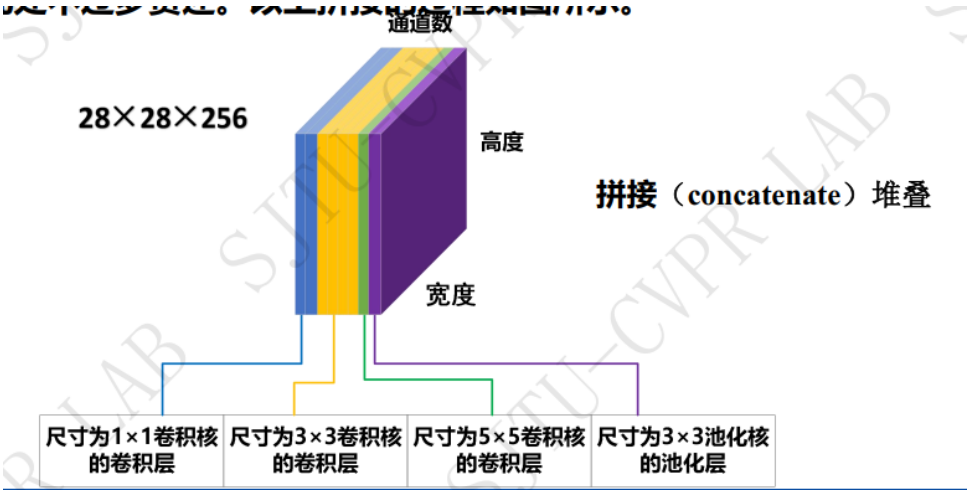


Figure 20: Inception 模块的输出

像素值用作下一层的激励。Alexnet 网络给出的具体计算公式如下：

$$b_{x,y}^{i}=a_{x,y}^{i}\Bigg/\left[k+\alpha\sum_{j=\max\left(0,i-\frac{n}{2}\right)}^{\min\left(N-1,i+\frac{n}{2}\right)}\left(a_{x,y}^{i}\right)^{2}\right]^{\beta} \tag{54}$$

9.4 ResNet

ResNet，又称残差网络，是由来自 Microsoft Research 的 4 位华人学者提出的卷积神经网络。特点是容易优化，并且能够通过增加相当的深度来提高准确率。ResNet 提出了残差块（Residual block）的概念，并对内部的残差块使用了一种叫做“shortcut connection”的连接方式，顾名思义，shortcut 就是“抄近道”的意思。这种跳跃连接有效缓解了在深度神经网络中增加深度带来的梯度弥散问题。