

# 合肥工业大学

## 2025 年《机器视觉》实验



实验内容: 图像滤波

姓 名: 折浩宇

学 号: 2023217896

完成时间: 2025/12/26

## 一、实验目的

### 1. 掌握图像滤波的基本思想与实现流程

通过对对自己拍摄的图像进行卷积运算（给定卷积核），理解“卷积核在图像上滑动并进行加权求和”的滤波机制，熟悉空间域滤波的基本步骤与边界处理方式。

### 2. 理解并实现 Sobel 算子边缘检测

使用给定的 Sobel 卷积核，对图像进行梯度计算与边缘增强，观察滤波前后图像在轮廓、纹理和细节方面的变化，理解边缘检测与梯度幅值的关系。

### 3. 完成颜色直方图的计算与可视化分析

对输入图像的 RGB 三通道分别统计像素强度分布，绘制颜色直方图，理解直方图对图像整体亮度、对比度、颜色成分的描述能力，为后续图像分析提供依据。

### 4. 实现纹理特征提取并保存特征数据

在不调用现成纹理特征提取函数的前提下，提取图像纹理描述特征（如 LBP、GLCM 等），并将最终特征向量保存为 .npy 文件，形成可复用的特征数据输入。

## 二、实验原理

### 2.1 数字图像与灰度化基础

一幅彩色图像通常表示为三通道矩阵： $I(x, y) = [R(x, y), G(x, y), B(x, y)]$ ，其中  $(x, y)$  为像素位置， $R, G, B \in [0, 255]$ 。

很多滤波和纹理分析（尤其是梯度、GLCM）通常在灰度图上进行，因此常先将 RGB 转为灰度图  $Y(x, y)$ 。常见线性灰度化为：

$$Y = 0.299R + 0.587G + 0.114B$$

这样得到的灰度值更符合人眼对不同颜色敏感度的差异。

### 2.2 空间域卷积滤波原理

#### 2.2.1 卷积的定义

空间域滤波本质是二维离散卷积（或相关运算）。给定图像  $I$  和卷积核（滤波器） $K$ ，输出图像  $O$  为：

$$O(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b K(i, j) \cdot I(x + i, y + j)$$

其中核大小为  $(2a+1) \times (2b+1)$ 。

直观理解：卷积核是一块“权重模板”，在图像上逐像素滑动，每个位置取邻域像素与权重相乘再求和，作为该点新像素值。

### 2.2.2 边界处理（边缘像素怎么卷积）

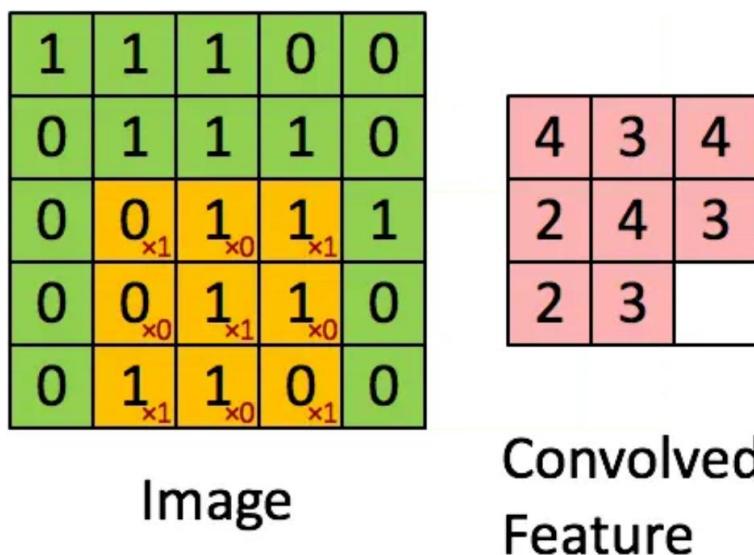
卷积核滑到图像边缘时会发生越界，需要进行边界扩展。常见方式包括：

- 零填充 (Zero Padding): 越界部分补 0 (实现简单，但边缘可能变暗)
- 复制填充 (Replicate): 用边缘像素复制填充
- 镜像填充 (Reflect): 按边缘镜像扩展 (视觉效果通常更自然)

### 2.2.3 卷积核的意义

卷积核的系数决定滤波器的作用：

- 系数相加为 1: 多用于平滑 (均值滤波、低通)
- 系数相加为 0: 多用于边缘/锐化 (高通、梯度算子)
- 权重分布具有方向性: 可增强水平/垂直/斜向边缘或纹理



## 2.3 Sobel 算子边缘检测原理

### 2.3.1 边缘的本质：灰度突变

图像边缘是灰度（或颜色）随空间位置变化剧烈的区域，本质上对应图像一阶导数（梯度）幅值较大处：

$$\nabla I = \left[ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

### 2.3.2 Sobel 卷积核与梯度估计

Sobel 用两个  $3 \times 3$  核近似计算 x 与 y 方向梯度（你实验里给出的核属于其中一个方向）。常见形式为：

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

对灰度图 I 做卷积得到：

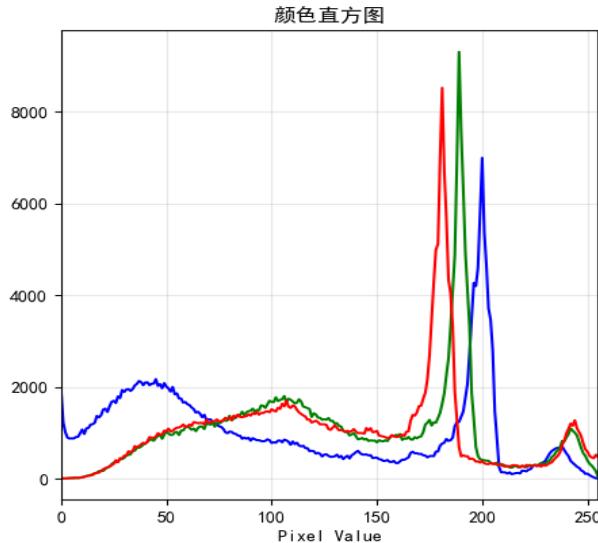
$$S_x = I * G_x, \quad S_y = I * G_y$$

梯度幅值（边缘强度）常用两种形式：

$$M = \sqrt{S_x^2 + S_y^2}$$

### 2.4 颜色直方图原理

颜色直方图描述某个通道中不同像素强度出现的频率。对 8-bit 图像，每个通道有 256 个灰度级 (0~255)。



对某通道  $C \in \{R, G, B\}$ ，其直方图定义为：

$$H_C(k) = \#\{(x, y) \mid C(x, y) = k\}, \quad k = 0, \dots, 255$$

为消除图像大小影响，常将直方图归一化为概率分布：

$$P_C(k) = \frac{H_C(k)}{\sum_{k=0}^{255} H_C(k)}$$

颜色直方图的意义：峰值位置反映亮度/主色调，分布宽窄反映对比度，多峰结构表示画面存在多种明显颜色区域。

## 2.5 纹理特征提取原理（LBP 与 GLCM）

纹理描述图像中重复结构、粗糙度、方向性等“局部模式”。本实验使用两类常用纹理特征：LBP（局部二值模式）与 GLCM（灰度共生矩阵）统计量。

### 2.5.1 LBP (Local Binary Pattern) 原理

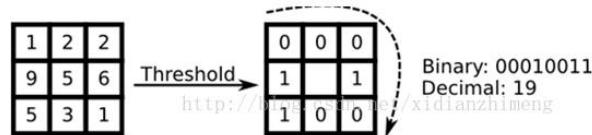
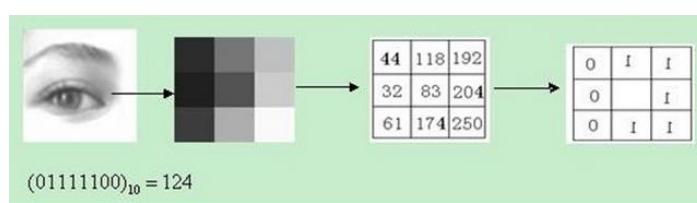
LBP 用一个像素与其邻域像素的相对大小关系编码纹理。以  $3 \times 3$  邻域为例，中心像素为  $g_c$ ，周围 8 个像素为  $g_0 \dots g_7$ 。定义比较函数：

$$s(g_i - g_c) = \begin{cases} 1, & g_i \geq g_c \\ 0, & g_i < g_c \end{cases}$$

LBP 编码：

$$LBP(x, y) = \sum_{i=0}^7 s(g_i - g_c) \cdot 2^i$$

最后对整幅图统计 LBP 值的直方图作为纹理特征向量（基础 LBP 为 256 维），通常再做归一化以便不同图像间可比。



用公式表示就是：

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

其中  $(x_c, y_c)$  是中心像素， $i_c$  是灰度值， $i_p$  是相邻像素的灰度值， $s$  是一个符号函数：

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \quad (1)$$

LBP 优点：计算简单；只比较大小，对整体光照变化具有一定鲁棒性；对微观纹理（斑点、颗粒、细密结构）描述有效。

### 2.5.2 GLCM (Gray-Level Co-occurrence Matrix) 原理

GLCM 描述灰度对在空间上共同出现的统计规律，可捕捉纹理的方向性与粗细。

给定灰度量化等级  $L$ （常将  $0 \sim 255$  压缩到  $16/32/64$  等较小等级以降低矩阵规模）与空间偏移  $\Delta = (\Delta x, \Delta y)$ （例如右邻居  $(1, 0)$ 、下邻居  $(0, 1)$ 、右下  $(1, 1)$  等）。

GLCM 矩阵  $P(i, j)$  的含义：

$$P(i, j) = \#\{(x, y) \mid I(x, y) = i, I(x + \Delta x, y + \Delta y) = j\}$$

归一化为概率矩阵：

$$p(i, j) = \frac{P(i, j)}{\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P(i, j)}$$

在此基础上可提取经典统计纹理特征（本实验常用四项）：

- 能量 Energy (角二阶矩 ASM): 越大表示纹理越规律、灰度对越集中。

$$Energy = \sum_i \sum_j p(i, j)^2$$

- 对比度 Contrast: 越大表示灰度变化更剧烈、纹理更粗或边界更明显。

$$Contrast = \sum_i \sum_j (i - j)^2 p(i, j)$$

- 熵 Entropy: 越大表示纹理分布越分散、越复杂。

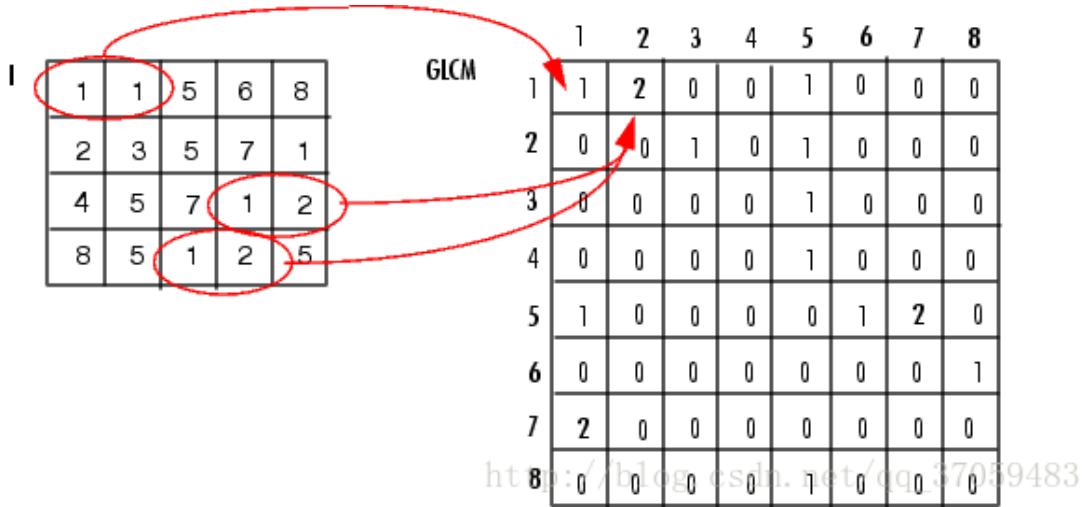
$$Entropy = - \sum_i \sum_j p(i, j) \log(p(i, j) + \epsilon)$$

- 相关性 Correlation: 体现灰度对线性相关程度与方向结构。

$$Correlation = \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j)p(i, j)}{\sigma_i \sigma_j}$$

GLCM 优点：能描述方向纹理（由偏移方向决定）；对宏观纹理（条纹、织物、

建筑窗格等) 有效。



## 2.6 特征保存 (.npy) 的原理意义

将纹理特征保存为 .npy 文件，本质是将特征向量以 NumPy 二进制格式序列化，便于后续复用（直接加载用于分类/检索/相似度计算）、保持数值精度，并能自动保留数组维度与数据类型信息。

## 三、实验流程与解析

### 3.1 实验整体流程

**读入图像：**从 my\_photo.jpg 读取 BGR 图像；若文件不存在则自动生成测试条纹图，保证流程可运行。

**预处理：**检测分辨率，若过大则按最大边长 600 等比例缩放；随后将 BGR 转 RGB 用于显示，并手写灰度化得到灰度图。

滤波任务：

- Sobel 边缘检测（分别卷积  $k_x$ 、 $k_y$ ，并合成梯度幅值）；
- 题目给定卷积核滤波（突出垂直边缘响应）。

**颜色直方图：**对 B/G/R 三通道分别手写统计 256 bins，并绘制曲线用于分析主灰度（颜色）分布。

**纹理特征：**① LBP（手写局部二值编码 + 256 维直方图向量）；② GLCM（手写共生矩阵 + 能量/对比度/熵/相关性）。

**保存与可视化：**保存纹理特征到 texture\_features.npy，并生成综合图展示输入图、滤波图、直方图与纹理特征。



项目流程图

### 3.2 实现二维卷积

卷积实现包含手动实现“零填充 + 滑动窗口求和”，并使用 float32 防止累加溢出。

```

def manual_convolution(img, kernel):
    h, w = img.shape
    k_h, k_w = kernel.shape
    pad_h, pad_w = k_h // 2, k_w // 2

    # 1) 零填充
    padded_img = np.pad(img, ((pad_h, pad_h), (pad_w, pad_w)),
                        'constant', constant_values=0)
    output = np.zeros((h, w), dtype=np.float32)

    # 2) 滑动窗口卷积
    for i in range(h):
        for j in range(w):
            region = padded_img[i:i+k_h, j:j+k_w]
            output[i, j] = np.sum(region * kernel)

    return output

```

要点：边界用零填充处理；卷积输出为浮点数，便于归一化与截断。

### 3.3 Sobel 边缘检测：双方向梯度 + 幅值合成

Sobel 分别计算 x、y 方向梯度 (Gx、Gy)，再合成为梯度幅值图。幅值越

大表示灰度变化越剧烈（更可能为边缘）。

```
def normalization(img_float):
    img_abs = np.abs(img_float)
    img_u8 = np.clip(img_abs, 0, 255).astype(np.uint8)
    return img_u8

def manual_sobel(img_gray):
    kx = np.array([[1, 0, -1],
                  [2, 0, -2],
                  [1, 0, -1]])
    ky = np.array([[1, 2, 1],
                  [0, 0, 0],
                  [-1, -2, -1]])

    gx = manual_convolution(img_gray, kx)
    gy = manual_convolution(img_gray, ky)

    mag = np.sqrt(gx ** 2 + gy ** 2)
    return normalization(mag)
```

要点：幅值合成可检测多方向边缘；abs+clip 保证可视化稳定（0~255）。

### 3.4 题目给定卷积核滤波：方向性增强（垂直边缘突出）

题目给定卷积核与 Sobel 的 G<sub>x</sub> 等价，因此输出更偏向垂直边缘增强。实现复用卷积函数并做统一归一化处理。

```
def manual_given_kernel_filter(img_gray):
    kernel = np.array([[1, 0, -1],
                      [2, 0, -2],
                      [1, 0, -1]])
    res = manual_convolution(img_gray, kernel)
    return normalization(res)
```

要点：与 Sobel 幅值图对比可体现“方向选择性”（单核增强特定方向边缘）。

### 3.5 纹理特征提取：LBP（局部模式）与 GLCM（共生统计）

#### 3.5.1 LBP：3×3 邻域二值编码 + 直方图向量

LBP 通过比较中心像素与 8 邻域像素大小生成二值模式，并按权重编码为 0~255；对 LBP 编码图统计 256 维直方图作为纹理特征向量。

```
def extract_lbp_texture(img_gray):
    h, w = img_gray.shape
```

```

lbp_img = np.zeros((h, w), dtype=np.uint8)

padded = np.pad(img_gray, ((1, 1), (1, 1)),
                mode='constant')

weights = np.array([[1, 2, 4],
                   [128, 0, 8],
                   [64, 32, 16]])

for i in range(1, h + 1):
    for j in range(1, w + 1):
        center = padded[i, j]
        nb = padded[i - 1:i + 2, j - 1:j + 2]
        binary = (nb >= center).astype(int)
        lbp_img[i - 1, j - 1] = np.sum(binary * weights)

hist, _ = manual_histogram(lbp_img)
return lbp_img, hist[0]

```

**要点：**使用“ $\geq$ ”可增强对整体亮度变化的鲁棒性；输出包含 LBP 编码图与 256 维特征向量。

### 3.8.2 GLCM：灰度量化 + 水平邻接统计 + 四个统计量

GLCM 描述灰度对在给定方向/距离上共同出现的概率。本实验先将灰度量化到 16 级，再统计  $0^\circ$ （水平）邻接像素对并对称化，得到概率矩阵 P，随后计算能量、对比度、熵与相关性。

```

levels = 16
img_q = (img_gray // (256 // levels)).astype(int)

glcm = np.zeros((levels, levels), dtype=float)
dx = 1 # 距离 d=1 ( $0^\circ$ 方向)

for i in range(h):
    for j in range(w - dx):
        a = img_q[i, j]
        b = img_q[i, j + dx]
        glcm[a, b] += 1
        glcm[b, a] += 1 # 对称化

P = glcm / np.sum(glcm)
I, J = np.ogrid[0:levels, 0:levels]

energy = np.sum(P ** 2)

```

```

contrast = np.sum(((I - J) ** 2) * P)

mask = P > 0
entropy = -np.sum(P[mask] * np.log2(P[mask]))

mu_x = np.sum(I * P)
mu_y = np.sum(J * P)
sigma_x = np.sqrt(np.sum(((I - mu_x) ** 2) * P))
sigma_y = np.sqrt(np.sum(((J - mu_y) ** 2) * P))

numerator = np.sum((I - mu_x) * (J - mu_y) * P)
denom = sigma_x * sigma_y
correlation = 0 if denom == 0 else numerator / denom

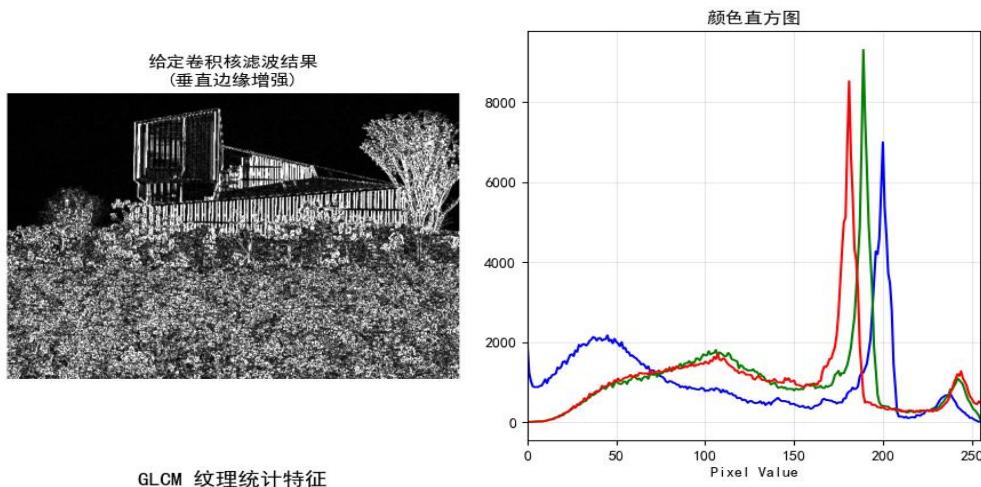
```

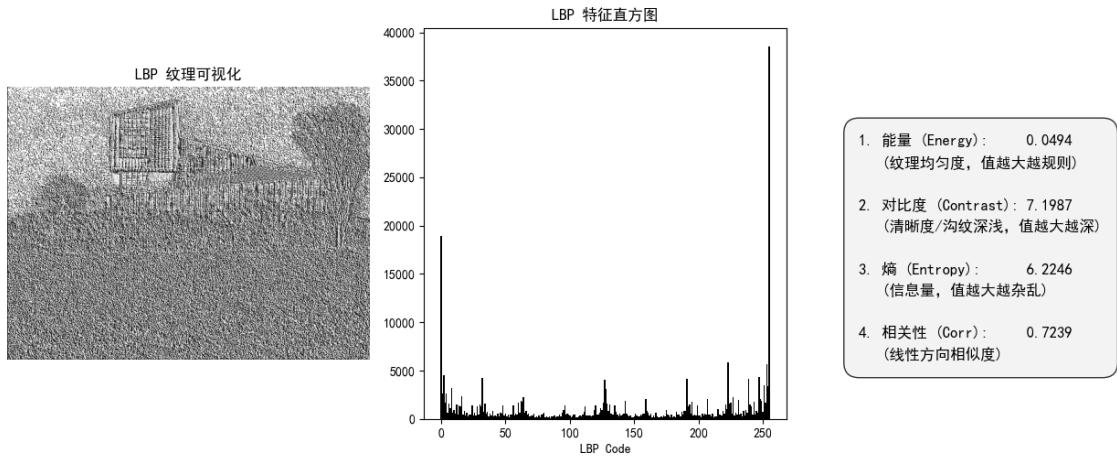
**要点：**量化降低矩阵稀疏性；熵跳过  $P=0$  项避免  $\log(0)$ ；相关性分母为 0 时需保护。

## 四、实验结果与分析

运行结果：

检测到原始分辨率过高 (4032x3024)，已缩放至：(450, 600, 3)  
 当前处理图像尺寸：(450, 600)  
 纹理特征已保存至 'texture\_features.npy'





#### 4.1 输入图像特性分析

实验输入为一幅具有明显规则结构的纹理图像,画面主要由**周期性垂直条纹与等间距水平细线构成**,且整体灰度级数量较少,属于典型的“规则纹理 + 大面积近似均匀区域”的图像。这一特性使得图像的边缘位置(条纹边界、水平线位置)具有显著的灰度突变,同时也便于验证卷积滤波与纹理特征提取算法的正确性与稳定性。

#### 4.2 Sobel 算子滤波结果分析

从“Sobel 算子滤波结果”可以观察到, **输出图像中条纹的边界位置被显著增强,呈现为清晰的高亮垂直边缘**;同时,输入图像中的水平细线也被检测为明显的边缘响应。其原因在于 Sobel 算子通过计算图像在水平与垂直方向的梯度( $(G_x)$  与  $(G_y)$ ),并进一步合成为梯度幅值图:

$$M = \sqrt{S_x^2 + S_y^2}$$

因此,凡是灰度变化较剧烈的区域(边缘、线条交界处)都会产生较大的梯度幅值并被突出显示。实验现象表明:本实验中 Sobel 梯度计算、幅值合成以及数值归一化(/截断到 0~255)过程正确,边缘提取效果明显。

#### 4.3 给定卷积核滤波结果分析 (**垂直边缘增强**)

“给定卷积核滤波结果”与 Sobel 输出相比表现出更强的方向选择性:图像中**垂直边缘**被更突出地增强,而水平细线的响应相对弱于 Sobel 幅值图。这是因为题目给定的卷积核:

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

本质上等价于 Sobel 的  $(G_x)$  ( $x$  方向梯度核)，其对“左右方向灰度突变”（即垂直边缘）最敏感，而对“上下方向突变”（水平边缘）的响应较弱。该结果与理论一致，说明手写卷积过程能够正确实现方向性滤波，且卷积核的方向结构能够直接决定增强的边缘类型。

#### 4.4 颜色直方图结果分析

颜色直方图呈现为**少数几个明显的尖锐峰值，而非连续平滑分布**。这说明图像像素值集中在有限的几个灰度/颜色等级上，符合输入图像“规则条纹 + 均匀背景块”的特点：大面积区域灰度值接近固定常数，导致对应灰度级的计数显著偏高，从而形成峰值。

#### 4.5 LBP 纹理特征结果分析

LBP 纹理可视化结果**呈现出明显的规则花纹结构，且在条纹边缘与水平线交叉区域出现更密集的纹理编码变化**；对应的 LBP 直方图中在高编码值（接近 255）处出现显著高峰。LBP=255 会在大面积平坦区域大量出现并形成直方图“主峰”；其他较小峰值则对应条纹边缘、线条交界处等灰度变化区域的编码模式。该结果表明：输入纹理具有明显的“规则结构 + 平坦区域占比大”的特征，LBP 直方图能较好地反映这一结构属性。

#### 4.6 GLCM 纹理统计特征结果分析

本实验计算得到的 GLCM 统计特征为：

**Energy = 0.0494 (很低)**：灰度共现概率分布很分散，说明纹理**不规则、均匀性差，不存在明显的周期性/重复结构**。

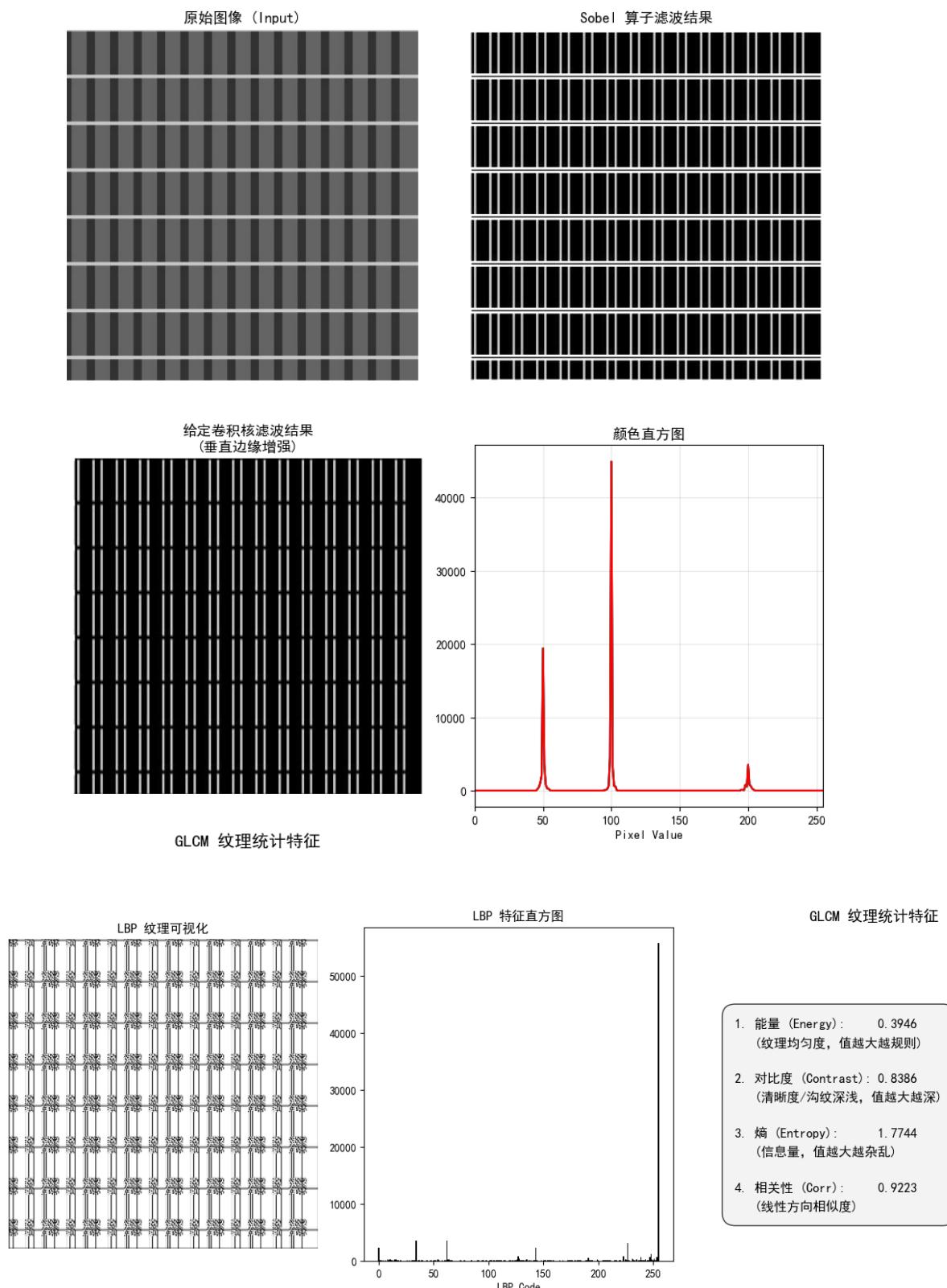
**Contrast = 7.1987 (较高)**：相邻像素灰度差异大，说明图像局部**存在较多边缘与灰度突变**，纹理起伏明显、细节清晰。

**Entropy = 6.2246 (很高)**：共现模式多、随机性强，说明**纹理复杂度高、信息量大**，符合自然场景中植被/建筑细节混合的特点。

**Correlation = 0.7239 (中等偏高)**：相邻灰度仍有一定连续性与结构相关，

说明图像存在一定方向或区域一致性（例如同一物体表面/大块区域），但整体仍受复杂纹理影响，未达到强规则纹理那种接近 1 的相关性。

## 我的下一组数据：



## 4.7 两个测试样例之间的对比

### 1) 输入图像特性对比

- 条纹测试图：**灰度级少、结构高度重复、方向性强**（垂直条纹+水平线），大面积近似均匀区域。
- 自然场景图：内容复杂（建筑/植被/天空等）、灰度变化丰富且不规则，纹理多尺度混合。

### 2) 滤波结果对比 (Sobel vs 给定核)

#### 条纹测试图

- Sobel 值幅图：垂直条纹边缘与水平线都非常强，呈现规则“网格型”高响应。
- 给定核（相当于 Gx）：更突出**垂直边缘**，横线相对弱一些，方向选择性更明显。

#### 自然场景图

- Sobel/给定核：**边缘分布更分散，响应集中在建筑轮廓、树枝、花草细节等处；整体边缘更“杂”，强弱不均。**

### 3) 颜色直方图对比

- 条纹测试图直方图：出现少数几个非常尖锐的峰（灰度值集中在几类固定值）。  
→ 反映“合成纹理/离散灰度级”特点。
- 自然场景图直方图：通常更分散、更连续（多颜色、多亮度层次）。  
→ 反映真实场景灰度/颜色丰富。

### 4) LBP 结果对比

- 条纹测试图 LBP:LBP **可视化呈现强周期结构**；直方图在某些编码（尤其 255）处有明显尖峰。  
→ 原因：大面积均匀区域 + 规则边界，邻域比较结果高度一致。
- 自然场景图 LBP: LBP 图更“噪声化/颗粒化”，直方图分布更广。  
→ 原因：纹理细节多且随机，局部模式种类多。

### 5) GLCM 四特征数值对比

#### 条纹测试图：

Energy = 0.3946, Contrast = 0.8386, Entropy = 1.7744, Correlation =

0.9223

自然场景图：

Energy = 0.0494, Contrast = 7.1987, Entropy = 6.2246, Correlation = 0.7239

(1) Energy: 0.3946 (条纹)  $\gg$  0.0494 (自然)

- 条纹图能量高：共生概率集中，纹理高度规则、重复强。
- 自然图能量低：共生概率分散，纹理不规则、均匀性差。

(2) Contrast: 0.8386 (条纹)  $\ll$  7.1987 (自然)

- 条纹图对比度低：量化后相邻灰度差较固定且不大，变化模式单一。
- 自然图对比度高：边缘/细节多（树枝、窗格、花草），局部灰度差更大、更频繁。

(3) Entropy: 1.7744 (条纹)  $\ll$  6.2246 (自然)

- 条纹图熵体现“有序”：共生模式少，信息量低  $\rightarrow$  熵低。
- 自然图熵体现“复杂”：共生模式多，随机性强  $\rightarrow$  熵高。

(4) Correlation: 0.9223 (条纹)  $>$  0.7239 (自然)

- 条纹图相关性更高：方向一致性强，相邻像素关系稳定。
- 自然图相关性中等偏高：仍有连续区域，但整体受复杂纹理干扰，方向一致性不如条纹明显。

与自然场景图相比，条纹测试图表现出**能量更高、熵更低、相关性更高、对比度更低**，说明条纹图纹理更**规则且方向一致**；而自然图呈现**能量低、熵高、对比度高**，说明其纹理**更复杂、不规则且灰度变化剧烈**。这也解释了为什么条纹图的 Sobel/卷积滤波输出更“干净、结构化”，自然图输出更“分散、细节化”。

## 五、心得体会

### 五、实验体会（心得体会）

通过本次“图像滤波与特征提取”实验，我对空间域滤波、边缘检测与纹理分析的完整流程有了更系统的理解，并在“自实现二维离散卷积”的实现细节上获得了较多收获。首先，手动实现卷积（零填充 + 滑动窗口加权求和）让我直观认识到：**卷积核的系数分布决定了滤波的方向性与功能**，同时边界处理方式会直接影响输出图像的边缘区域质量。实验中采用零填充实现简单，但也能观察到

边缘处可能出现响应偏暗或不连续的现象，因此后续若追求视觉效果或测量精度，可以进一步尝试复制/镜像填充并对比输出差异。

其次，在 Sobel 边缘检测部分，通过分别计算 ( $G_x$ )、( $G_y$ ) 并做梯度幅值合成，**我更清楚地理解了“边缘=灰度突变”的本质：**幅值合成能够兼顾多方向边缘，而单方向核（题目给定核与 ( $G_x$ ) 等价）则体现出明显的方向选择性，这种差异在输出图上非常直观。与此同时，我也意识到**归一化/截断策略**会影响结果观感与后续特征统计，因此需要统一规范处理流程以保证可比性。

在纹理特征部分，LBP 与 GLCM 从“局部模式”与“统计共现”两个角度刻画纹理，互补性很强。特别是 GLCM 的四个统计量 (Energy/Contrast/Entropy/Correlation) 能够把“规则纹理”和“自然复杂纹理”的差异用数值量化表达：在自然场景图中呈现能量低、熵高、对比度高（纹理复杂且灰度变化剧烈），而条纹测试图则表现为能量更高、熵更低、相关性更高（纹理规则且方向一致）。这种基于两组样例的对比分析让我更确信：**特征的意义必须结合图像结构来解释，单看数值没有说服力。**

最后，从工程实现角度看，本实验还训练了我对程序鲁棒性与可复现性的关注：例如输入图像缺失时自动生成测试图、控制图像尺寸以降低计算量、将最终纹理特征保存为 .npy 便于复用等做法，使得实验结果更稳定、流程更完整。后续若进一步提升实验质量，我计划：

- 1) 增加不同边界填充策略与不同量化等级/方向/距离参数对 GLCM 的影响对比；
- 2) 对 LBP 直方图做归一化并尝试旋转不变/等价模式等改进版本；
- 3) 在保持“自实现核心计算”的前提下，对卷积与统计步骤进行适度优化（如向量化/减少重复计算），提高效率与可扩展性。