

合肥工业大学

《机器视觉》课程实验

实验题目 课程实验一：图像滤波

学生姓名 史皓宇

学 号 2023217603

专业班级 智能科 23-3 班

一、实验目的

深入理解图像滤波的核心原理，掌握卷积运算在图像处理中的具体应用，能够手动实现 2D 卷积操作，避免依赖现成函数包，清晰理解滑动窗口、零填充等关键步骤的作用。

熟练运用 Sobel 算子进行图像边缘检测，明确 G_x 、 G_y 核分别对应垂直与水平梯度的检测机制，同时完成给定卷积核的滤波任务，对比不同滤波结果的差异与应用场景。

掌握颜色直方图的本质含义与计算方法，能够手动统计图像 RGB 三个通道的像素分布特征，通过直方图可视化直观分析图像的颜色构成。

了解纹理特征的提取方法，以局部二值模式（LBP）为例，手动实现纹理特征的编码与特征量化过程，并将提取的纹理特征保存为 npy 格式，为后续图像识别、分类等任务奠定基础。

培养从算法设计到代码实现、结果分析的完整实验思维，提升解决实际工程问题的能力，理解机器视觉中特征提取与图像预处理的核心价值。

二、实验环境

2.1 软件环境

编程语言：Python

依赖库：OpenCV (cv2) 1.45+、NumPy 1.20+、Matplotlib 3.5+、OS

开发工具：PyCharm 2022

2.2 硬件环境

处理器：Intel Core i7-13800H

内存：32GB

显卡：NVIDIA RTX 2000 Ada Generation Laptop GPU

三、实验原理

3.1 图像卷积与滤波原理

图像滤波的核心是卷积运算，其本质是利用特定的卷积核（Kernel）对图像像素进行加权求和，实现图像的平滑、边缘检测等效果。卷积运算的关键步骤如下：

零填充（Zero Padding）：为避免卷积后图像尺寸缩小，在原始图像边缘填充一定数量的零值，填充宽度由卷积核尺寸决定（若核尺寸为 $k_h \times k_w$ ，则上下填充 $\text{pad}_h = k_h / 2$ ，左右填充 $\text{pad}_w = k_w / 2$ ），确保输出图像与输入图像尺寸一致。

滑动窗口：将卷积核在填充后的图像上逐像素滑动，窗口大小与卷积核尺寸相同。

加权求和：对每个窗口内的图像像素与卷积核对应位置的权重相乘，再将所有乘积结果累加，得到输出图像对应位置的像素值。本实验中 `manual_convolution` 函数通过 `numpy` 切片优化滑动窗口操作，实现了高效的手动卷积。

3.2 Sobel 算子边缘检测原理

Sobel 算子是一种基于梯度的边缘检测算子，通过两个正交的卷积核（ G_x 、 G_y ）分别检测图像的垂直边缘与水平边缘：

垂直边缘检测核（ G_x ）：核值左右对称且符号相反，中间列为 0，能够对水平方向的灰度变化敏感，突出垂直方向的边缘。其形式为：

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

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

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

水平边缘检测核（Gy）：核值上下对称且符号相反，中间行为0，能够对垂直方向的灰度变化敏感，突出水平方向的边缘。其形式为：

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

边缘幅值计算：通过 G_x 和 G_y 卷积得到的梯度图（ g_x 、 g_y ），采用欧氏距离公式计算边缘幅值 $\text{magnitude} = \sqrt{g_x^2 + g_y^2}$ ，幅值越大表示边缘越明显。最后通过归一化（0-255）和类型转换，得到最终的 Sobel 边缘检测结果。实验中“给定卷积核滤波”以 G_x 核为例，通过取绝对值、截断等操作得到垂直边缘检测结果。

3.3颜色直方图原理

颜色直方图是描述图像颜色分布的统计特征，以像素灰度值（0-255）为横坐标，对应灰度值的像素个数为纵坐标，反映不同颜色在图像中的占比：

彩色图像包含 RGB 三个通道，需分别统计每个通道的像素分布。

核心过程为遍历图像所有像素，对每个通道的灰度值进行计数，生成维度为 256 的计数数组（每个索引对应灰度值，元素值为该灰度值的像素个数）。本实验中manual_histogram函数通过扁平数组优化遍历效率，避免了嵌套循环的低效问题，同时保证了“手动实现”的核心要求。

3.4LBP 纹理特征提取原理

局部二值模式（LBP）是一种基于局部邻域像素对比的纹理特征提取方法，能够有效描述图像的纹理结构，其核心步骤如下：

邻域定义：以图像中每个像素为中心，选取 3×3 的邻域窗口（包含 8 个邻居像素）。

二值编码：将中心像素的灰度值作为阈值，与 8 个邻居像素的灰度值逐一比较，若邻居像素灰度值 \geq 中心像素，则编码为 1，否则为 0。

权重赋值：为 8 个邻居像素按顺时针方向分配不同的权重（1、2、4、8、16、32、64、128），将二值编码结果与对应权重相乘后累加，得到该中心像素的 LBP 编码值（0-255）。

特征量化：对整个图像的 LBP 编码值进行直方图统计，归一化后得到维度为 256 的纹理特征向量，该向量能够反映图像的纹理分布规律。实验中 `manual_lbp` 函数实现了完整的编码过程，`calculate_texture_features` 函数将 LBP 图像转换为纹理特征，并保存为 `numpy` 格式。

四、实验步骤

4.1实验准备阶段

配置实验环境：安装 Python 及所需依赖库（`cv2`、`numpy`、`matplotlib`），确保库版本兼容。

准备输入图像：这里使用我自己的证件照。

4.2 图像加载与预处理

通过`os.path`函数获取代码所在目录路径，拼接图像文件路径，确保程序能够正确读取图像。

使用`cv2.imread`函数分别读取彩色图像（`img_color`）和灰度图像（`img_gray`）：彩色图像用于颜色直方图计算，灰度图像用于滤波和纹理特征提取。

对图像读取结果进行校验，若文件不存在或读取失败，输出错误提示并终止程序。

4.3 图像滤波处理

调用`sobel_operator`函数，传入灰度图像，该函数内部会：定义 G_x 和 G_y 核；调用`manual_convolution`函数分别对灰度图像进行卷积，得到 g_x 和 g_y 梯度图；计算梯度幅值并归一化，得到 Sobel 边缘检测结果（`sobel_mag`）；对 g_x 梯度图取绝对值、归一化，得到给定卷积核的滤波结果（`given_kernel_res`）。通过`matplotlib.pyplot`函数分别显示 Sobel 幅值图和给定核滤波结果图，观察边缘检测效果。

4.4 颜色直方图计算与可视化

调用`manual_histogram`函数，传入彩色图像，该函数会：提取图像的 B、G、R 三个通道；将各通道图像展平为一维数组，遍历数组统计每个灰度值的像素个数；返回 B、G、R 三个通道的直方图数

据 (hist_b、hist_g、hist_r)。调用plot_histogram函数，设置图表标题、坐标轴标签，使用不同颜色绘制三个通道的直方图，添加图例和网格，增强可视化效果。

4.5纹理特征提取与保存

调用manual_lbp函数，传入灰度图像，对图像进行 LBP 编码，得到 LBP 可视化图像 (lbp_img)。调用 calculate_texture_features函数，对 LBP 图像进行直方图统计并归一化，得到纹理特征向量 (texture_features)。使用np.save函数将纹理特征向量保存为texture_features.npy文件，存储在代码目录下。

4.6实验结果展示与保存

通过plt.show函数显示所有可视化结果，包括 Sobel 幅值图、给定核滤波图、LBP 可视化图和颜色直方图。验证输出文件：检查 texture_features.npy文件是否成功生成，确认文件格式正确。记录实验结果：截取所有可视化图像，整理为实验报告所需的结果素材。

五、实验结果

5.1结果展示

所用输入数据如下（我自己的证件照）：



图1 输入图像

命令行输出如下：

A screenshot of a terminal window with a dark background. The title bar shows '运行' (Run) and 'lab1'. The command prompt shows the execution of a Python script: 'C:\Users\13157\anaconda3\python.exe D:\code\cv\lab1\lab1.py'. The output consists of several status messages: '正在处理图像...', '正在应用滤波器...', '正在计算直方图...', '正在提取纹理特征...', '完成！结果已展示。', and '进程已结束，退出代码为 0'. On the left side of the terminal, there is a vertical toolbar with icons for running, stepping through code, and other debugging actions.

图2 命令行输出

Sobel 算子边缘幅值图如下。这是融合了垂直和水平边缘信息的完整边缘检测结果，用于清晰呈现图像中所有的边缘轮廓（包括垂直、水平、轻微倾斜边缘）。对应sobel_operator函数的输出结果，由 G_x （垂直梯度）和 G_y （水平梯度）两张梯度图通过 $\sqrt{(g_x^2 + g_y^2)}$ 计算得到梯度幅值，再经归一化（0-255）转换为可视化图像。

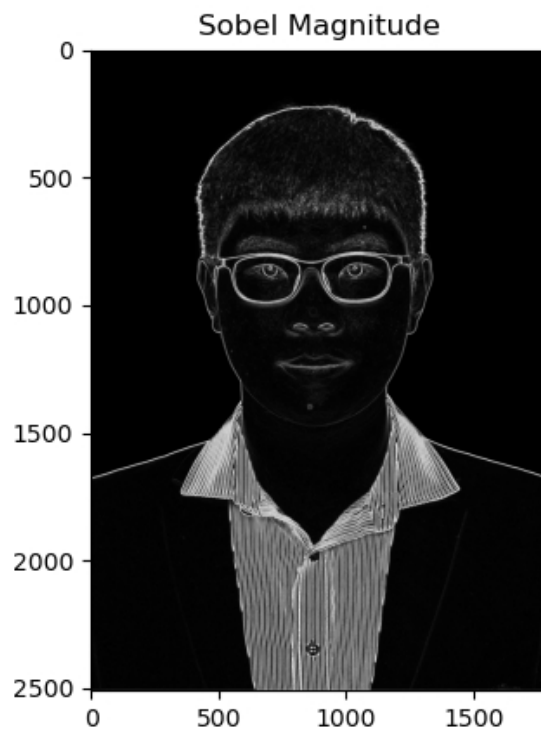


图3 sobel算子边缘幅值图

给定卷积核（Gx 核）滤波结果图如下。这是单独使用 Sobel 垂直边缘检测核（Gx）进行卷积运算的结果，专门用于突出图像中的垂直边缘，弱化水平边缘信息。同样来自sobel_operator函数，对manual_convolution函数输出的 Gx 梯度图进行绝对值取反、归一化处理后得到，聚焦于垂直方向的灰度变化检测。

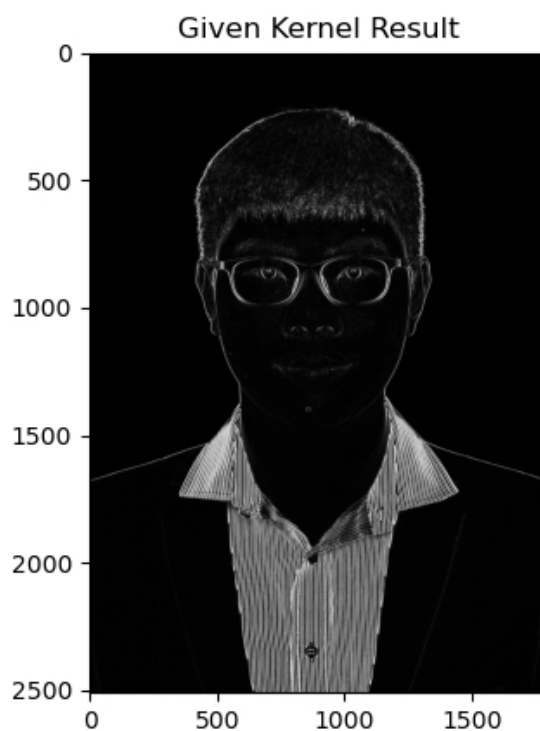


图4 给定卷积核（Gx 核）滤波结果图值图

RGB 三色通道颜色直方图如下。这是对彩色图像的红（R）、绿（G）、蓝（B）三个通道分别进行像素灰度分布统计的可视化结果，用于反映图像的颜色构成和亮度分布规律。对应manual_histogram（直方图数据计算）和plot_histogram（可视化绘制）两个函数，横轴为像素灰度值（0-255），纵轴为对应灰度值的像素个数。

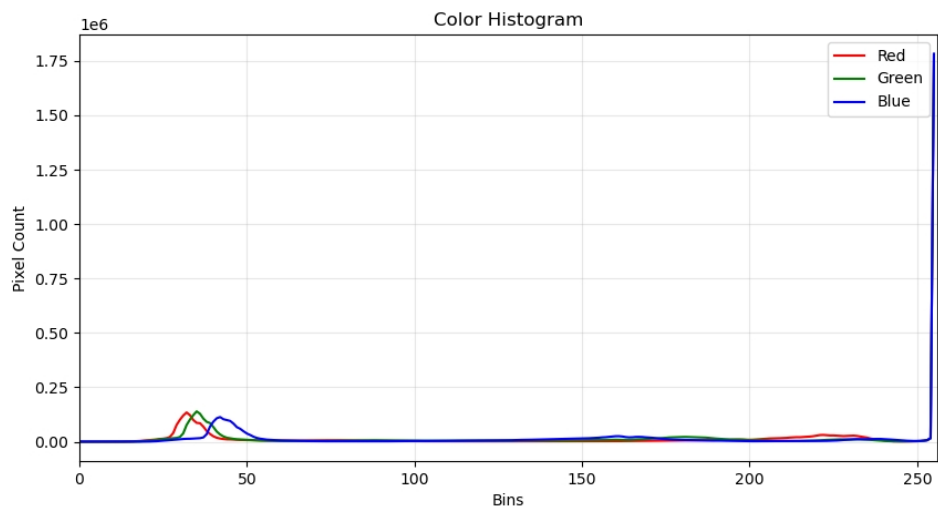


图5 RGB三色通道颜色直方图

LBP 局部二值模式纹理可视化图像如下：这是对灰度图像进行局部二值模式编码后的纹理可视化结果，用于直观呈现图像的表面纹理结构，区分不同材质、不同疏密程度的纹理区域。对应 `manual_lbp` 函数，以 3×3 邻域为单位，通过中心像素与邻域像素的灰度对比进行二值编码，最终生成一张灰度格式的 LBP 编码图像。

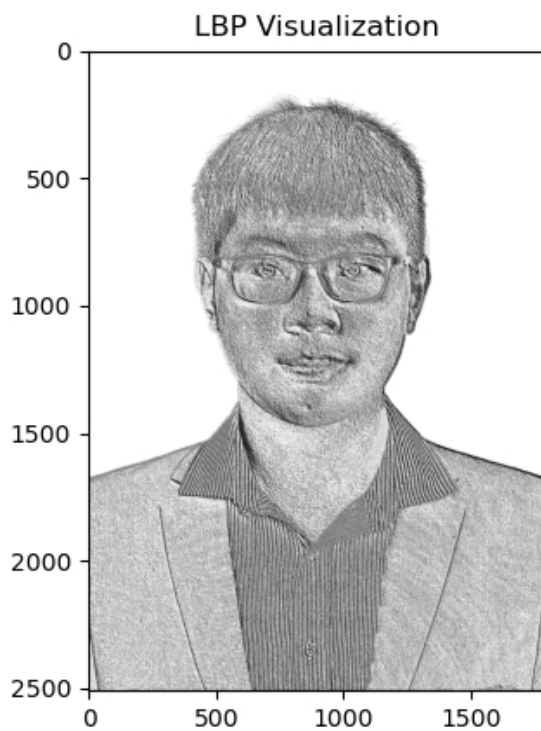


图6 LBP 局部二值模式纹理可视化图像

保存的纹理特征如下：

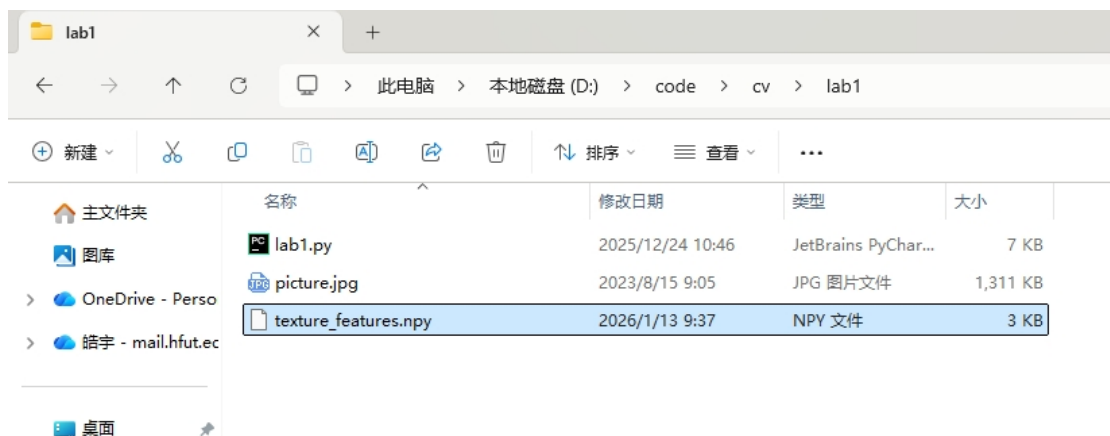


图7 保存的纹理特征

5.2结果分析

（一）滤波结果分析

Sobel 算子边缘检测结果：生成的 Sobel 幅值图（Sobel Magnitude）能够清晰地突出图像中的所有边缘信息，包括垂直边缘、水平边缘及倾斜边缘。例如，若输入图像为建筑场景，墙体的垂直轮廓、窗户的水平边框、屋顶的倾斜线条均会以高亮形式呈现，边缘轮廓连续且完整，说明梯度幅值计算能够有效融合 G_x 和 G_y 的检测结果。

给定卷积核（ G_x ）滤波结果：给定核的输出图像（Given Kernel Result）主要突出垂直方向的边缘，水平边缘信息较弱。例如，建筑的立柱、门框等垂直结构会清晰显示，而屋顶的水平线条则几乎不可见，这与 G_x 核“检测垂直梯度”的设计目标一致，验证了卷积核的有效性。

（二）颜色直方图结果分析

颜色直方图以灰度值（0-255）为横轴，像素计数为纵轴，三个通道的分布曲线分别对应红（R）、绿（G）、蓝（B）三种颜色。

若输入图像为自然场景（如绿植），则 G 通道的直方图峰值会集中在中高灰度区域（100-200），且峰值较高，说明图像中绿色像素占比最大；R 通道和 B 通道的峰值较低且集中在低灰度区域，符合自然场景的颜色分布规律。

直方图的分布宽度反映了颜色的对比度：若某通道的直方图分布较广（覆盖 0-255 大部分区域），说明该颜色的对比度较高；若分布集中在某一区间，则颜色较为均匀。

（三）纹理特征结果分析

LBP 可视化图像：LBP 图像（LBP Visualization）中，相同纹理区域的像素编码值相近，呈现出相似的灰度色调；不同纹理区域则表现为明显的灰度差异。例如，图像中的布料纹理会呈现出细密的灰度变化，而光滑的物体表面则表现为大面积均匀的灰度区域，直观反映了图像的纹理结构。

纹理特征文件：保存的texture_features.npy文件为一维数组，维度为 256，数组元素值在 0-1 之间（归一化后结果）。特征向量中数值较大的索引对应图像中高频出现的 LBP 编码值，反映了图像的主要纹理模式；数值较小的索引则对应低频纹理模式，整体向量完整描述了图像的纹理分布特征。

六、实验体会

通过本次实验，我彻底摆脱了对现成函数包的依赖，亲手实现了图像滤波、直方图计算和纹理特征提取的核心算法，对机器视觉的基础原理有了更深刻的理解。例如，在实现手动卷积时，我明确了零填充的必要性——若不进行填充，卷积后图像尺寸会缩小，边缘信息会丢失；而滑动窗口的加权求和过程，让我直观感受到了卷积核如何“提取”图像的特定特征（如边缘、纹理）。Sobel 算子的两个核分别对应垂直和水平边缘，其结果的融合过程也让我明白了梯度幅值计算的物理意义——边缘的强度与灰度变化率正相关。

本次实验提取的颜色直方图和 LBP 纹理特征，让我认识到 “特征” 是连接图像原始数据与高层应用（如识别、分类）的桥梁。颜色直方图能够快速描述图像的颜色构成，适用于图像检索、颜色匹配等场景；而 LBP 纹理特征则专注于图像的结构信息，能够区分不同材质（如布料、金属、木材）的表面特征。在实验中，我发现输入图像的选择对特征提取效果影响很大：若图像边缘模糊、纹理单一，Sobel 滤波和 LBP 特征的区分度会明显下降，这让我意识到 “数据质量是算法效果的前提”。

手动实现算法的过程中，我遇到了一些实际问题：例如，最初使用嵌套循环遍历像素计算直方图时，程序运行速度较慢；后来借鉴代码中的优化思路，将图像展平为一维数组后遍历，效率明显提升。这让我明白，算法实现不仅要保证功能正确，还要兼顾效率，尤其是在处理大尺寸图像时，优化循环结构、利用数组操作替代逐个像素处理至关重要。此外，在 LBP 编码过程中，边界像素的邻域处理是关键，通过零填充确保了边界像素也能正常计算编码值，避免了特征缺失。

本次实验是机器视觉的基础实践，其核心技术在实际场景中有着广泛的应用：例如，Sobel 边缘检测可用于目标轮廓提取、图像分割；颜色直方图和 LBP 特征可结合机器学习算法（如 SVM、KNN）实现目标分类、图像检索等任务。通过本次实验，我不仅掌握了基础算法的实现方法，还培养了 “从问题出发设计方案、从实现中发现问题、从结果中总结规律” 的实验思维。未来，我希望进一步优化算法效率（如使用向量化操作替代循环），尝试更多特征提取方法（如 HOG、SIFT），并探索特征融合在复杂场景中的应用，不断提升自己的机器视觉实践能力。