



学号： S202488044

姓名： 谢阳阳

# Image Filtering and Hybrid Images

## 1. 实验介绍

### 1.1 实验目标

本实验旨在探索图像滤波与混合图像的核心技术，通过处理两幅不同图像的频率特征，构建出一幅多尺度混合图像。

1. 学习并实现常用的滤波技术（高斯滤波、拉普拉斯滤波等）。
2. 通过频域分解与合成技术构建混合图像，并分析其效果。
3. 探讨混合图像在不同观察条件下的视觉特性。

### 1.2 主要工具与框架

#### 编程语言与工具

- Python：作为主要编程语言，便于实现图像处理操作。
- OpenCV：高效的图像处理库，用于读取、滤波、变换等操作。
- Matplotlib：用于可视化图像处理结果。

#### 硬件环境

- 普通个人计算机即可，无需特殊硬件加速。

## 2. 相关知识

### 2.1 图像滤波基础

#### 滤波的定义

图像滤波是通过滤波器对图像进行操作，抑制某些特定频率分量（如噪声）或增强其他特定频率分量（如边缘）。

#### 滤波的分类

- **空域滤波**：直接对图像像素值进行操作，通过卷积核与图像区域进行点积实现。
- **频域滤波**：先对图像进行傅里叶变换，将图像转化到频率域后操作，再逆变换回空间域。

#### 常见滤波器

##### 1. 低通滤波器

- 平滑图像，去除噪声和高频细节。
- 常见方法：高斯滤波（Gaussian Blur）。
- 数学表达式： $G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$
- 作用：用于保存相应的图像低频部分

##### 2. 高通滤波器

- 提取边缘和细节信息，抑制低频部分。
- 常见方法：拉普拉斯滤波（Laplacian）、Sobel 滤波等。
- 拉普拉斯滤波的离散形式：

$$\Delta f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

#### 滤波的应用：

- 图像增强（如提高对比度）
- 边缘检测（如提取对象轮廓）
- 噪声抑制（如消除随机噪声）

## 2.2 混合图像的原理

### 频率分解

图像可以视为不同频率信息的组合：

- **低频成分**：描述图像中的整体结构和光滑区域。
- **高频成分**：描述图像中的细节和快速变化部分。

利用频率分解，可以分别提取一幅图像的低频和另一幅图像的高频部分，最终进行融合

### 构建混合图像的步骤

对第一幅图像应用低通滤波器提取低频成分。

对第二幅图像应用高通滤波器提取高频成分。

将两部分按权重相加，形成混合图像：

$$H(x, y) = L_1(x, y) + H_2(x, y)$$

其中， $L_1(x, y)$ 是第一幅图像的低频部分， $H_2(x, y)$ 是第二幅图像的高频部分。

## 3. 实验流程

### 3.1 实验环境准备

软件环境：

安装必要的 python 库： opencv-python 、 matplotlib、 numpy

数据准备

选择两幅高质量的图像，分别作为低频图像和高频图像的输入源

### 3.2 图像滤波的实现

低频成分的提取：

使用高斯滤波对第一幅图像进行处理提取相应的低频信息

高频成分提取：

1. 首先使用高斯滤波器对第二幅图像进行低频平滑处理
2. 再者之后利用原图减去低频部分，得到高频信息

### 3.2 混合图像的生成

通过上述操作获取到的低频图像和高频图像信息，对两者进行加权叠加

$$H(x, y) = L_1(x, y) + H_2(x, y)$$

## 4. 实验结果分析

原始的高频图像和低频图像如下图 1、图 2 所示



图 1 待提取低频图像



图 2 待提取高频图像

经过图片滤波后得到的高低频信息如下图 3 所示

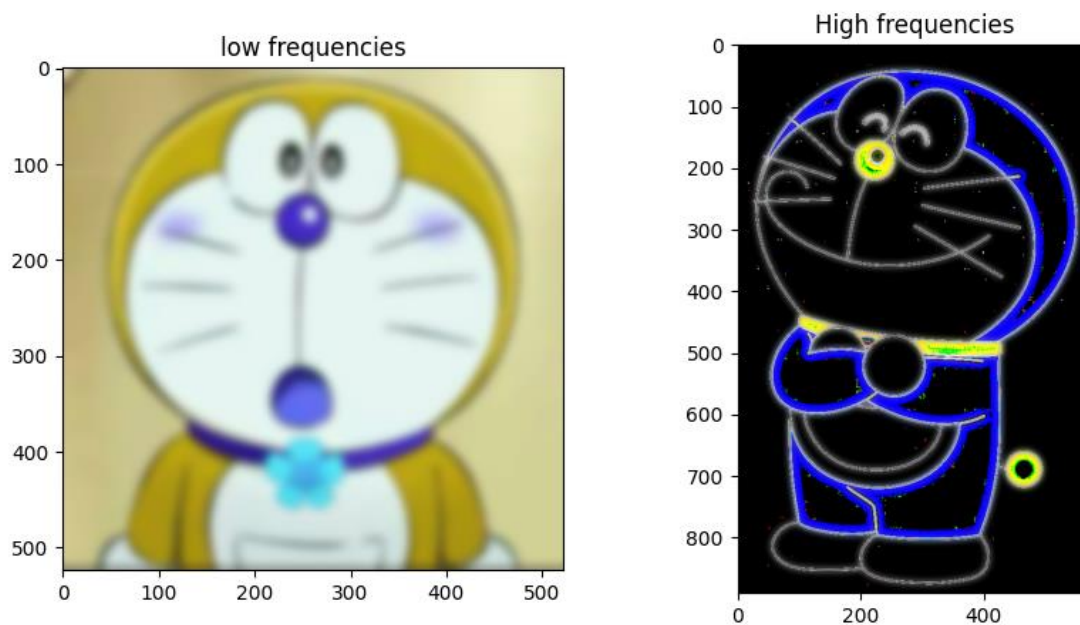


图 3 高低频信息

经过高低频信息混合后得到的图像如图 4 所示

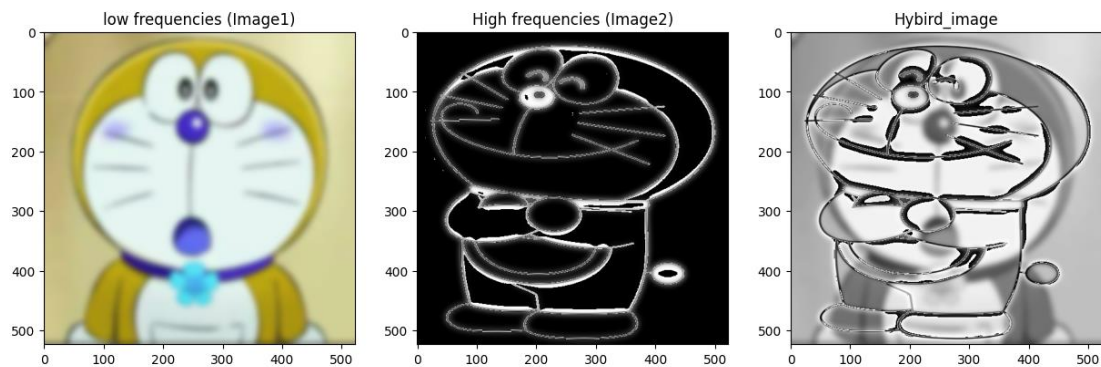


图 4 混合图像

## 附录

```
image1=cv2.imread("./pic_data/image1.jpg",0)
image2=cv2.imread("./pic_data/image2.jpg",0)
height, width = image1.shape[:2]
# Resize image2 to match the size of image1
resized_image2 = cv2.resize(image2, (width, height), interpolation=cv2.INTER_LINEAR)
# 提取 image1 的低频
low_frequencies_image_1=cv2.GaussianBlur(image1,(31,31),0)
#提取图像 B 的高频
low_frequencies_image_2=cv2.GaussianBlur(resized_image2,(31,31),0)
high_frequencies_image_2=resized_image2-low_frequencies_image_2
# 混合: 图像 1 与图像 2
hybird_image=low_frequencies_image_1+high_frequencies_image_2
# 显示相应的结果
plt.figure(figsize=(15,5))
plt.subplot(1,3,1)
plt.imshow(low_frequencies,cmap="gray")
plt.title("low frequencies (Image1)")

plt.subplot(1,3,2)
plt.imshow(high_frequencies_image_2,cmap="gray")
plt.title("High frequencies (Image2)")

plt.subplot(1,3,3)
plt.imshow(hybird_image,cmap="gray")
plt.title("Hybird_image")
plt.show()
```