Image Processing Assignment

Krithika , 22011101046

February 5, 2025

1 Aim

To apply various image processing techniques such as convolution, correlation, mean filtering, and median filtering on an image and analyze their effects using histograms.

2 Source Code

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Load image in color (BGR)
img = cv2.imread('icecream.jpg')

# Convert BGR to RGB for proper display in matplotlib
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Function to plot histograms for each color channel
def plot_histogram(image, title):
    plt.figure(figsize=(12, 6))
    r_channel, g_channel, b_channel = cv2.split(image)
    plt.subplot(1, 2, 1), plt.imshow(image)
    plt.title(title)
    plt.subplot(1, 2, 2)
    plt.hist(r_channel.ravel(), bins=256, range=(0, 256),
        color='red', alpha=0.7, label='Red')
    plt.hist(g_channel.ravel(), bins=256, range=(0, 256),
        color='green', alpha=0.7, label='Green')
    plt.hist(b_channel.ravel(), bins=256, range=(0, 256),
        color='blue', alpha=0.7, label='Blue')
    plt.title(f'{title} Histogram')
    plt.legend()
```

```
    plt.show()

# Convolution and Correlation Filters
kernel = np.array([[1, 1, 1],
                   [1, -7, 1],
                   [1, 1, 1]])
convolution_result = cv2.filter2D(img, -1, kernel)
kernel_flip = np.flip(kernel)
correlation_result = cv2.filter2D(img, -1, kernel_flip)
plot_histogram(convolution_result, 'Convolution Result')
plot_histogram(correlation_result, 'Correlation Result')

# Mean Filter
mean_filter = np.ones((3, 3), np.float32) / 9
mean_filtered = cv2.filter2D(img, -1, mean_filter)
plot_histogram(mean_filtered, 'Mean Filtered Image')

# Median Filter
median_filtered = cv2.medianBlur(img, 3)
plot_histogram(median_filtered, 'Median Filtered Image')
```

3 Output Images

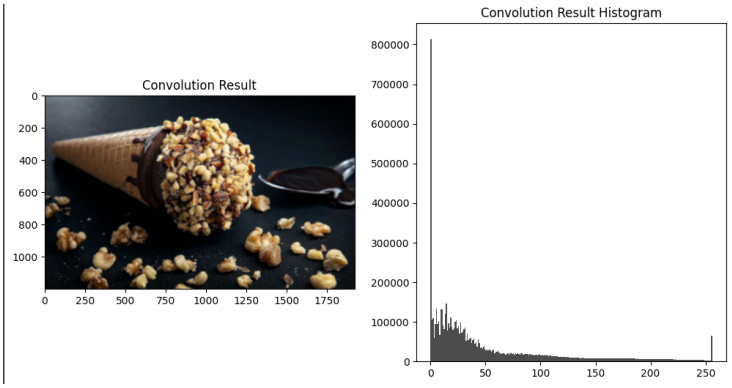

Figure 1: Convolution Result

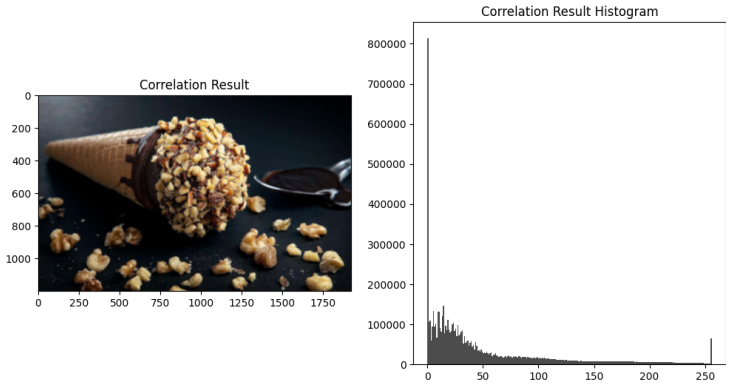

Figure 2: Correlation Result

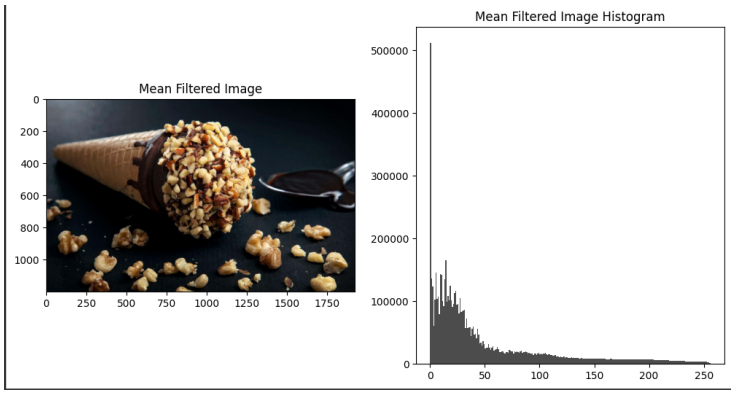

Figure 3: Mean Filtered Image

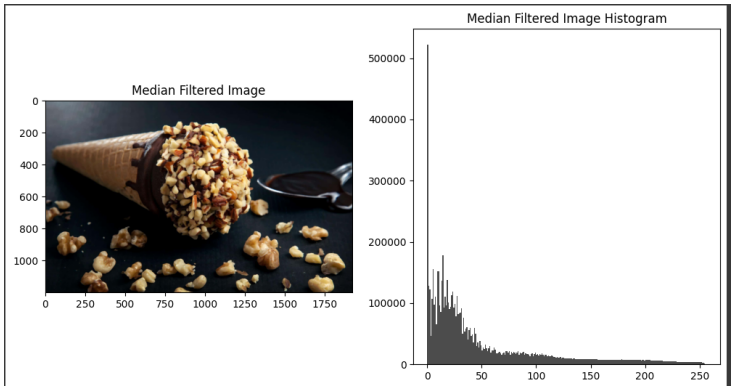

Figure 4: Median Filtered image