HW7: Image Processing (EE604)

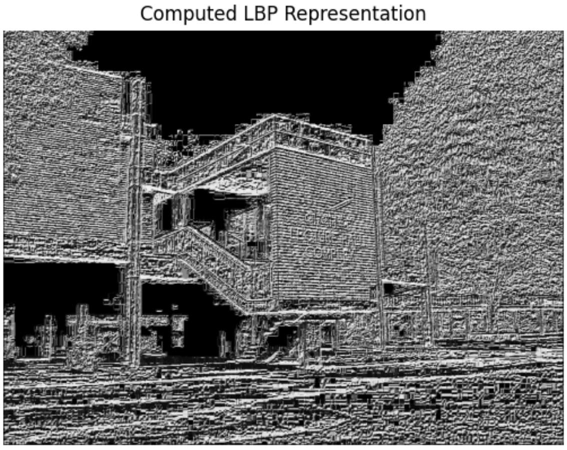
Chirag Garg (210288)

October 26, 2024

Code

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

# Load the grayscale image
image = cv2.imread('lhc.png', cv2.IMREAD_GRAYSCALE)


# Create an empty array to store the LBP values
lbp_result = np.zeros_like(image, dtype=np.uint8)

# Calculate LBP values for each pixel by comparing with neighboring pixels
for y in range(1, image.shape[0] - 1):
    for x in range(1, image.shape[1] - 1):
        center_pixel = image[y, x]
        binary_string = (
            ('1' if image[y-1, x-1] > center_pixel else '0') +
            ('1' if image[y-1, x] > center_pixel else '0') +
            ('1' if image[y-1, x+1] > center_pixel else '0') +
            ('1' if image[y, x+1] > center_pixel else '0') +
            ('1' if image[y+1, x+1] > center_pixel else '0') +
            ('1' if image[y+1, x] > center_pixel else '0') +
            ('1' if image[y+1, x-1] > center_pixel else '0') +
            ('1' if image[y, x-1] > center_pixel else '0')
        )
        lbp_result[y, x] = int(binary_string, 2)

# Define cell dimensions to split image into 2x2 grid
cell_height = lbp_result.shape[0] // 2
cell_width = lbp_result.shape[1] // 2

# Create an empty list to store histograms from each cell
concatenated_histogram = []

# Calculate histograms for each cell in the 2x2 grid and normalize each histogram
for row in range(0, lbp_result.shape[0], cell_height):
    for col in range(0, lbp_result.shape[1], cell_width):
        cell_region = lbp_result[row:row + cell_height, col:col + cell_width]
        histogram, _ = np.histogram(cell_region.ravel(), bins=256, range=(0, 256))
        normalized_hist = histogram / histogram.sum()  # Normalize the histogram
        concatenated_histogram.extend(normalized_hist)

# Display the computed LBP image
plt.imshow(lbp_result, cmap='gray')
plt.title("Computed LBP Representation")
plt.axis("off")
plt.show()

# Plot the concatenated histogram vector
plt.plot(concatenated_histogram)
plt.title("Concatenated LBP Histogram (Feature Vector)")
plt.xlabel("Histogram Bins")
plt.ylabel("Normalized Frequency")
plt.show()
```

Result



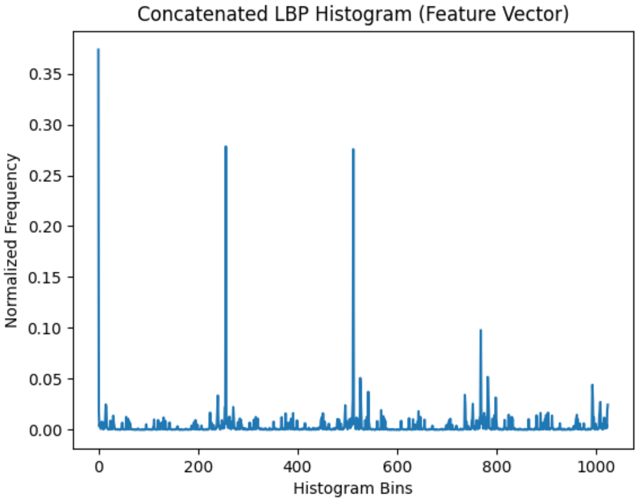


Figure 1: Histogram