HW9: Image Processing (EE604)

Chirag Garg (210288)

November 8, 2024

Code

```
from collections import Counter
from heapq import heapify, heappop, heappush

# Image data from the PDF file
image_data = [[1, 4, 2, 5, 4, 3, 3, 3],
              [4, 3, 0, 5, 2, 0, 4, 1],
              [2, 3, 1, 2, 1, 1, 3, 2],
              [5, 1, 1, 2, 2, 2, 2, 3],
              [1, 4, 1, 2, 0, 4, 3, 4],
              [1, 3, 1, 5, 1, 0, 1, 0],
              [1, 2, 2, 2, 5, 0, 4, 5],
              [1, 3, 0, 1, 5, 1, 1, 4]]

# Flatten the image data and count frequencies
flattened_data = [pixel for row in image_data for pixel in row]
frequency = Counter(flattened_data)

# Create a Huffman tree based on frequencies
heap = [[freq, [symbol, ""]] for symbol, freq in frequency.items()]
heapify(heap)

while len(heap) > 1:
    lo = heappop(heap)
    hi = heappop(heap)
    for pair in lo[1:]:
        pair[1] = '0' + pair[1]
    for pair in hi[1:]:
        pair[1] = '1' + pair[1]
    heappush(heap, [lo[0] + hi[0]] + lo[1:] + hi[1:])

# Extract the Huffman codes
huffman_codes = sorted(heappop(heap)[1:], key=lambda p: (len(p[-1]), p))

# Calculate the average number of bits (Lavg)
total_pixels = len(flattened_data)
avg_bits = sum(frequency[symbol] / total_pixels * len(code) for symbol, code in
↪  huffman_codes)

for num, code in huffman_codes:
    print("number:", num, ", Huffman code:", code)

print("Average number of Bits required:",avg_bits)
```

Result

Huffman code:
number: 1, Huffma ncode: 10
number: 2, Huffman code: 00
number: 0, Huffman code: 010
number: 3, Huffman code: 111
number: 4, Huffman code: 110
number: 5, Huffman code: 011

Average number of Bits required: 2.515625