Homework 4

(Total 300 pts)

Due 5:00 pm, April 14, 2025 (Monday)

Submit the following 5 files on Canvas ("HW4.pdf", "Q1.m" "golomb_enco.m", "Q3.m", "Q4.m").

1. (100 pts) Load in Matlab the following MAT file (for a 1-D sequence *J*, which was converted from the 'coins.png' image) http://www.ece.uah.edu/~dwpan/course/ee614/data/coins_1d.mat. We want to implement and compare two schemes to losslessly compress the data sequence in *J* using Huffman codes.
   (a) What is the first-order entropy of the source $H(J)$?
   (b) Write a Matlab M file to code the image using Huffman code (using built-in functions such as huffmandict, huffmanenco, and huffmandeco). How many distinct symbols (pixel values) does the dictionary contain? What is the size of the coded bitstream? And on average how many bits/pixel is needed to code the sequence *J*?
   (c) Now, we want to apply Huffman code on blocks of two consecutive symbols. As an illustrating example, suppose an input sequence has eight symbols: [49 50 48 49 48 48 49 50]. Instead of Huffman coding each of these six symbols individually as we have done in (b), we will code the following two-symbol blocks: [(49 50), (48 49), (48 48), (49 50)], by treating each block as a symbol in an extended alphabet consisting of three unique block symbols: {(49 50), (48 49), (48 48)}. Note that the block (49 50) is the mostly likely block (which appears twice) in this example sequence.
   (d) What is the entropy (in terms of bits/two-symbol block) of an extended alphabet (J_2) derived from the input source *J*, by using the above two-symbol blocking scheme? Compare with the $H(J)$ obtained in (a). Which one is smaller (in terms of number of bits/pixel on average) and why?
   (e) What is the most likely two-symbol block in the input source *J*? How many times does this most likely symbol block appear?
   (f) Apply Huffman coding on the blocks of two consecutive symbols. How many distinct block-symbols does the dictionary contain? What is the size of the coded bitstream? And on average how many bits/original symbol (or pixel) is needed to code the sequence *J*? Comment on your results by comparing them with the results in (b).
   (g) Implement Huffman decoding on the coded bitstream obtained in (f) to reconstruct the original sequence *J*. Perform lossless check. Can your decoder recover the original image? Attach a screenshot showing the lossless check is passed.
   (h) Submit on Canvas the Matlab scripts ("Q1.m") you used to obtain the above results.
   (i) Fill in the following table with the results.

(a) $H(J)$	(b) # of distinct symbols (pixel values)	(b) Size of the coded bitstream (bits)	(b) Avg. # of bits/pixel

(d) $H(J_2)$	(e) Most likely two-symbol block	(e) # of times the most likely block symbol appears
	( ) followed by ( )	

(f) # of distinct block symbols	(f) Size of the coded bitstream (bits)	(f) Avg. # of bits/pixel	(g) Lossless check passed?

2. (60 pts) Golomb encoder implementation and lossless test.
   (a) Write a Matlab function to implement a generic Golomb Encoder, with the following syntax:

Note that the coding parameter *m* can be either a power of two, or not a power of two. Also, your Encoder Function should work with the Golomb Decoder function below:
http://www.ece.uah.edu/~dwpan/course/ee614/code/golomb_deco.m

```
function code = golomb_enco (n, m)
%   Encodes the input symbol n using Golomb code, based on
%   the coding parameter m;
%   Input: n is a nonnegative integer, m is the coding parameter.
%   Output: code is the codeword, a 1-D vector of binary bits.
```

(b) Save the above function as "golomb_enco.m". Enclose a copy of the source code in the main submission file "HW4.pdf". Also, submit this M file "golomb_enco.m" separately on Canvas.

(c) Test if your function can work with the Golomb Decoder function below:
http://www.ece.uah.edu/~dwpan/course/ee614/code/golomb_deco.m
To this end, run the following Matlab script:
http://www.ece.uah.edu/~dwpan/course/ee614/code/lossless_test.m
Was the Lossless Test passed? Enclose a screenshot of running the test in the main submission file "HW4.pdf".

3. (60 pts) Lossless compression of a sequence of non-negative integers.
http://www.ece.uah.edu/~dwpan/course/ee614/data/n.mat
   (a) Load the above MAT file in Matlab.
   (b) Plot the histogram of the *n* values. Use 'hist(n, max(n)+1)', where *max*(*n*) is the largest value in the sequence. Show the histogram.
   (c) What is the first-order entropy of this sequence?
   (d) Use Huffman Code to compress the above sequence, based on the Huffman code dictionary trained using the same sequence. What is the size of the compressed bitstream?
   (e) Use Golomb Code (the function "golomb_enco.m" you wrote in Question 1) to compress the above sequence (with coding parameter $m = 32$). What is the size of the compressed bitstream?
   (f) Use Golomb Code (the function "golomb_enco.m" you wrote in Question 1) to compress the above sequence (with coding parameter $m = 16$). What is the size of the compressed bitstream?
   (g) Use Golomb Code (the function "golomb_enco.m" you wrote in Question 1) to compress the above sequence (with coding parameter $m = 8$). What is the size of the compressed bitstream?
   (h) Fill in the table below with your results.

(c) Entropy (bits/symbol)	**(d) Huffman Coded Bitstream Size**	**(e) Golomb Coded Bitstream Size ($m = 32$)**	**(f) Golomb Coded Bitstream Size ($m = 16$)**	**(g) Golomb Coded Bitstream Size ($m = 8$)**

   (i) Submit your Matlab script "Q3.m" used to obtain the above results.

4. (80 pts) Predictive Golomb coding of an image.
The procedure of implementing a predictive coding method called differential pulse-code modulation (DPCM) is explained using an example below:
Suppose I = $\begin{bmatrix} 1 & 3 & 2 \\ 6 & 8 & 7 \end{bmatrix}$ is a 2×3 image. We first convert the 2D image into a 1D vector as

I-1D = $[1, 3, 2, 6, 8, 7]$, by scanning through the first row from left to right, then the second row, as so on.

Instead of coding the pixels values directly, we code the prediction residue sequence, which is obtained by taking the difference of the current pixel value from the previous pixel value. So starting from the second pixel, the residue sequence can be formulated as: I-Residue = $[2, -1,\ 4,\ 2, -1]$.

Since Golomb codes apply on non-negative integers, we need to convert the entries in the residue sequence into non-negative integers, by following the following rule:

Any integer k will be converted to a non-negative integer n, where $n = \begin{cases} 2k, & \text{if } k \geq 0 \\ -2k-1, & \text{if } k < 0 \end{cases}$

Thus the converted residue sequence becomes I-Residue-Convert = $[4,\ 1,\ 8,\ 4,\ 1]$. We then apply the Golomb encoder on the converted residue sequence.

Load the following MAT file (for an image X) in Matlab,
http://www.ece.uah.edu/~dwpan/course/ee614/data/coins.mat

(a) What is the first-order entropy of the image $H(X)$?
(b) Follow the same procedure as described in the above example, convert the image into a 1D vector and then obtain the prediction residue sequence. Watch out on the limit of uint8 type.
(c) Plot the normalized histogram of the residue sequence (using the option: 'normalization','pdf'). What is the first-order entropy of the residue sequence?
(d) Convert the residue sequence into a sequence of non-negative integers. Plot the normalized histogram of the converted residue sequence. What is the first-order entropy of the converted residue sequence?
(e) Determine the Golomb coding parameter m based on the converted residue sequence using $m = \left\lceil -\frac{1}{\log_2\left(\frac{E}{1+E}\right)} \right\rceil$, where $\lceil\ \ \rceil$ is the ceiling function, and E is the sample mean of the converted residue sequence. What is the value of m?
(f) Encode the converted residue sequence using the Golomb encoder you implemented in Problem 1. What is size (in bits) of the entire coded bitstream?
(g) Encode the converted residue sequence using Huffman code. How many distinct symbols are there in the Huffman code dictionary? What is the size (in bits) of the entire coded bitstream?
(h) Compare the coded bitstream sizes obtained in (f) and (g) with the coded bitstream size obtained by applying the Huffman code directly on the original image X.
(i) Fill in the following table with the results, and comment on your results. For example, compare the pros and cons of Huffman code and Golomb code, and discuss the advantages and limitations of the predictive coding method based on DPCM. Any idea on how to modify the coding parameter estimation method in (e) to improve the performance of Golomb code?
(j) Submit on Canvas the Matlab scripts ("Q4.m") you used to obtain the results.

(a) $H(X)$	(c) Entropy of the residue sequence	(d) Entropy of the converted residue sequence	(e) m	(f) Coded bitstream size (Golomb coding)	(g) Coded bitstream size (Huffman coding on the converted residue sequence)	(h) Coded bitstream size (Huffman coding on the original image)