Research Task

Abstract

This task is part of V-Nova's interviewing process for engineering candidates. It consists of a background section, as well as two task modules, covering C++ programming part and a results presentation/analysis for further discussion during the interview. Some of the tasks are mandatory to complete, while some others are optional and can be used for further discussion during the face-to-face interview with the candidate. You are encouraged to complete as many parts as possible.

There is no set time limit, but we suggest as a guideline that completing tasks 2.1 and 2.2 should together take around 90-120 minutes to complete. Results should be submitted before the interview.

For any further information or questions, please contact V-Nova.

Contents

1. Background

1.1. Shannon entropy

1.1.1 Definition

Shannon entropy H of a discrete memoryless random variable X is defined as:

$$\mathbf{H}(\mathbf{X}) = -\sum_{x \in \mathcal{X}: p(x) > 0} p(x) \cdot log_2(p(x))$$

where

$$\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$$

is the set of possible values (referred to as the source alphabet) and **p(x)** the probability mass function (*pmf*).

According to Shannon's source coding theorem, entropy H is a lower bound of the code rate (average number of bits per symbol) that can be achieved by any lossless coding scheme. This theorem becomes very useful when designing new compression/encoding data techniques, as it allows us to estimate the cost of different encoding methods in bits, and choose the one that best fits our requirements.

1.1.2 Estimating the entropy of observed data

We can estimate the pmf **p(x)** from the observed data by building the histogram of the data values.

For instance, let's assume the following data sample, containing 4 "0" symbols, 8 "1" symbols, and 4 "2" symbols (16 symbols in total):

D = [0, 2, 0, 1, 0, 2, 1, 2, 2, 0, 1, 2, 1, 2, 2, 2]

Using a source alphabet X = {0, 1, 2}, we can estimate the *pmf* as:

$$p(x) = \begin{cases} \frac{1}{4}\ if\ x = 0 \\ \frac{1}{2}\ if\ x = 1 \\ \frac{1}{4}\ if\ x = 2 \end{cases}$$

In this case, the entropy is

$$H = -\frac{1}{4} \cdot log_2 \frac{1}{4} \quad -\frac{1}{2} \cdot log_2 \frac{1}{2} \quad -\frac{1}{4} \cdot log_2 \frac{1}{4} = 1.5\ bits$$

and the minimum compressed size for the whole data sample, D, would be

B = 16 symbols · 1.5 its/symbol = 24 bits.

1.2 Estimating the entropy of a 2D image

A 2D image is typically represented in memory as a 2D array of dimensions WxH, with W being the width and H the height of the array. For simplicity, in this document we assume W=H=N, i.e. we work with square images of dimensions NxN. Each pixel of the image is assumed to be an 8-bit unsigned number, with integer values in the interval [0, 255].

1.2.1 Entropy of a full image

The entropy formula (Sec. 1.1.1) can be applied on the pixel values, to estimate the cost of a compressed image file during image encoding. To this end, we would need to build the histogram of all possible 256 pixel values [0, 1, 2, …, 255], which will serve as our estimate for the pmf **p(x)**, and then apply the entropy formula on **p(x)**.

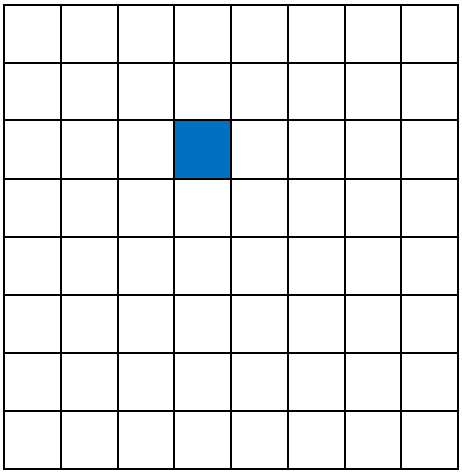
1.2.2 Entropy of image blocks

An alternative way to encode an image is by splitting it to square blocks of dimensions BxB, where B is a positive integer number, with B ≤ N. In this case, we can estimate the entropy of each image block, and then

average the entropy values of all blocks to get our estimate for the total entropy of the whole image. For simplicity, we may assume that B is always an integer factor of N, i.e. $N = B \cdot \lambda$, where λ is a positive integer number.

Each block can be referenced by its side dimension, B, and by its top-left coordinates, (i,j), where i denotes the vertical dimension and j the horizontal dimension. Obviously, a block size B=N is equivalent to covering the full image (as in Sec. 1.2.1), with the top-left coordinates being (0,0).

For instance, the following scheme shows an image of size 256x256, split in 64 rectangular blocks, each of size 32x32. The top-left coordinates of the blue block are (64, 96).



1.2.3 Entropy of image residuals

An alternative way to encode an image, X, is by using another image, Y, as a reference point. In this case, we can compute the residual image, R, where

R[i,j] = X[i,j] - Y[i,j], for i=0,1,…,N-1 and j=0,1,…,N-1

During decoding, we can recover image X by using R and Y, as following:

X[i,j] = R[i,j] + Y[i,j], for i=0,1,…,N-1 and j=0,1,…,N-1

We can assume that the reference image Y will always be available in the decoder, and thus we only need to encode the residual image, R. The entropy formula can be applied as normal (Sec 1.1.1), by building the pmf **p(x)** of the residual image values.

1.2.4 Entropy of image block residuals

Finally, we can split the residual image, R, to square blocks of dimensions BxB, where B $\leq$ N, and estimate the final entropy similar to Sec. 1.2.2.

1.3 Building an adaptive coding algorithm

Let's assume that we have the original image X, a reference image, Y, and a block size, B $\leq$ N, $N = B \cdot \lambda$.

We can then define two different encoding modes, named here as *Intra* mode or in *Inter* mode. These modes are defined for this specific exercise as follows:

- In *Intra* mode encoding, we encode the block values of the original image, X.
- In *Inter* mode encoding, we encode the values of the residual image, R.

In this case, we can choose the best mode for each block by using the following algorithm:

For each block of size BxB:

1. Compute the entropy, H_x, of the original image block values (using the relevant values of image X)
2. Compute the entropy, H_R, of the residual image block values (using the relevant values of image R)
3. **If** $H_x \leq H_R$ **then:**

   Choose the *Intra* mode for this block

   else:

   Choose the *Inter* mode for this block

The *final* entropy of a block is either H_x or H_R, depending on the mode decision (*Intra* or *Inter*).

The decisions for each block can be stored in a binary array, e.g. using a "0" for an Intra mode decision and a "1" for an Inter mode decision. As this array serves as additional metadata that would need to be transmitted to the decoder, we can compute its entropy too, to get an estimate of the metadata cost.

2. Task description

This task simulates the development and analysis of a simple image compression scheme, which

- receives the input image, X
- encodes X using the adaptive coding algorithm (Sec. 1.3)
- estimates the entropy of the encoded data for various values of the block size, B.

The task includes:

- a C++ programming section
- a results presentation section for further discussion during the interview

2.1 C++ programming task

You are provided with a C++ program skeleton (*main.cpp*), which reads two square image files, X and Y, of dimensions NxN. The program also uses an integer number, B, denoting the image block size (see Sec. 1.2.2). You are also provided with two raw image files to use for testing your implementation (*imageX_512x512.raw* and *imageY_512x512.raw*).

Your task is to complete the program as follows:

A. Complete the function *compute_residual_image* to calculate the residual image, R (Sec. 1.2.3).

B. Complete the templated function *calculate_entropy_of_image_block(img, i, j, B)* to compute the Shannon entropy of an image block of dimensions BxB with top-left coordinates, (i,j). This function will be further applied on the original image, X, and the residual image, R.

C. Complete the function *adaptive_coding_algorithm* for the adaptive coding algorithm (Sec. 1.3). The algorithm needs to:

   C.1. Scan each image block iteratively.

   C.2. Compute the Intra and Inter mode entropy for each block in the image (using the C++ function defined in step B, as defined above).

   C.3. Calculate the average entropy for the whole image, defined as the average of the *final* block entropies.

   C.4. Return the average entropy of the whole image.

D. Expand the function *adaptive_coding_algorithm* to include the following:

   D.1. Store the Intra/Inter mode decision to a 2D array (termed as the *metadata* array).

   D.2. Calculate the percentage of Intra mode blocks.

   D.3. Return both the average entropy and the percentage of Intra mode blocks.

E. *[Optional / Bonus part]* Assume that you need to encode and transmit the source image, X, over a communication network (e.g. over internet) to a receiver, who has a valid data decoder and a full copy of the reference image, Y. (You may need to collect additional data to answer the following questions.)

E.1. Describe briefly what data you would need to transmit so that the decoder can accurately decode and reconstruct the encoded image. *(Bullet points are fine)*

E.2. What would be the estimated total data size (in bits)?

E.3. What would be the optimal block size, B, that minimises the estimated total data size for this specific image, X?

2.2 Report writing

Write a brief report (up to 1-2 pages) to summarise your work and any experimental results you like to present/discuss during the interview.

3. Expected deliverables

- C++ code (Sec. 2.1)
- A PDF file with a report (Sec. 2.2)