Digital Image and Video Processing

Lab3: The Haar Transform & Image Compression

Student Name:
Hugh Jordan

Student Number:
16321743

The Haar Transform & Image Compression

1. Recalling your knowledge of Variable Length Coding

1a. Calculate the Entropy of this quantised image.

The entropy of the quantised image was equal to 1.7516. This was found using the following lines:

```
h = -p.*log2(p);
h = (h(~isnan(h)));
entropy = sum(h);
```

1b. Using the Huffman Coding method, design a set of variable length codewords for this image. Include your Huffman Coding Tree in your report.

The variable length codewords were designed using the line `[dict, avglen] = huffmandict(symbols, p)`. The corresponding Huffman Coding Tree is shown in the Figure 1.

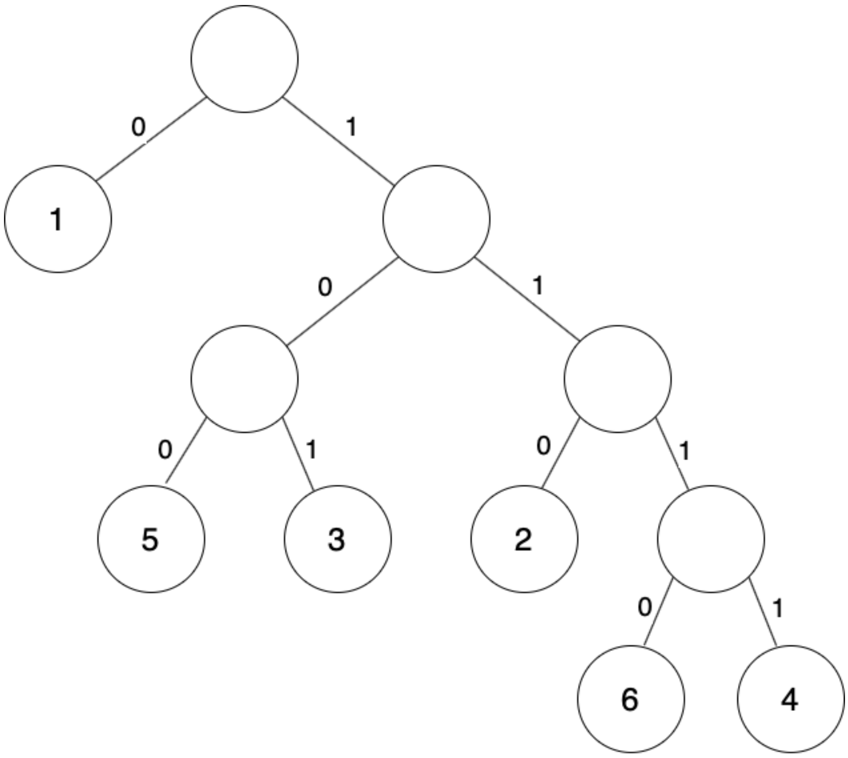

Figure 1. Huffman Coding Tree

1c. Calculate the average codeword length using your designed codewords. Does it agree with the Entropy of the events in bits/pel i.e. $-\sum_i p_i \log_2(p_i)$? Explain any discrepancy.

The average codeword length was equal to 1.8125 using the same line as above. This value was close to, but not equal, to the entropy of the image. The reason for this is that, according to the Source Coding Theorem (Shannon 1948), the entropy of an image is the minimum bound of the average codeword length that will preserve the information of that image. This theorem also states that the average codeword length must be within 1 bit per symbol of that codeword's minimum. Therefore, the average codeword length of this image must be within the range (17516, 1.8125), which means the average codeword length does agree with the answer above. If the average codeword length was below this minimum, then this would suggest that some information would have been lost.

2. Calculating Image Entropy

2a. Write a MATLAB function calcEntropy that calculates the Entropy of an image.
This function takes one 2D array, Y, that contains the image intensities of a picture as an input and returns the entropy of that picture. This function begins by validating that the array Y contains only numeric or logical values and that the array also contains only real, nonempty and nonsparse values. This function then calculates the probability of each intensity using the line `p = histcounts(Y, (0:255))` and by dividing the result of that line by the sum of the entire array. Following this, the entropy is calculated using the line `h = -p.*log2(p)` and any values that are equal to NaN are removed from the array using the line `h=(h(~isnan(h)))`. Finally, the remaining values are summed together to give the entropy of the image, and this value is returned.

2b. Estimate the entropy, H_o, for the unquantised image that has been given to you.
The entropy, H_o, of the unquantised image was equal to 7.0818.

2c Estimate the entropy (H_{qi}) of the quantised image using the given Q_step value.
The entropy, H_{qi}, of the quantised image was equal to 3.5215. The quantisation was performed using a Q_step value of 15.

2d. Explain any difference between H_{qi} and H_o.
This difference is a result of the image being quantised. Quantisation is the process of mapping a larger set of input values with small probabilities to a smaller set of output values with larger probabilities using rounding or truncation. This was performed in 2c using line `pic_qi = Q_step*round(pic/Q_step).` This has the effect of rounding each image intensity to the nearest factor of a given Q_step value. As a result, this decreases the entropy of that image. This is because the entropy of an image is minimised when a few symbols are much more probable than the others.

2e. Write a MATLAB function to calculate the MSE between 2 images.
This function takes two 2D arrays, Y1 and Y2, that contain the image intensities of two pictures as inputs and returns the mean square error between these two pictures. This function begins by validating that the two arrays only contain numeric or logical values and that both arrays only contain real, nonempty and nonsparse values. Following this, this function then verifies that both arrays are of the same class and that both arrays are equal in size. Finally, this function casts both of these arrays to doubles if they are integers and calculates the mean square error using the line `MSE = sum(power(Y1(:) - Y2(:), 2))/numel(Y1)`. The function then returns the value MSE.

2f. What is the MSE between the quantised and unquantised images?
The MSE between the quantised and unquantised image is 18.4985. The images intensities were cast to doubles using the function double() instead of using the function im2double(). This means that the image intensities ranged from (0, 255) instead of the range (0, 1). This convention was the convention used in the rest of the report.

2g. Comment on the visual quality of the quantised and unquantised images.
The unquantised image is shown in Figure 2, and the resulting image is shown in Figure 3. It is evident from Figure 3 that the image is quantised because of the staircase effect across the girl's forehead. This effect is a result of the larger set of input values in Figure 2 being mapped to the smaller set of output values in Figure 3. The image intensities in Figure 2 range from (0, 1, 2, … , 255).

However, the image intensities in Figure 3 range from (0, 15, 30, … 255). This is because the Q_step value was equal to 15.

Figure 2. Unquantised Image

Figure 3. Quantised Image

3. The 2D Haar Transform

3a. Write a MATLAB function that implements the 1-level Haar Transform and outputs an image of its subbands.
This function takes a 2D array, Y, containing the image intensities of a picture as an input and returns the level-1 Haar Transform of that picture. This function begins by validating that the array Y only contains numeric or logical values and that the array only contains real, nonempty and nonsparse values. This function then verifies that both the width and height of the image are divisible by 2. Following this, the image is split into four subarrays by taking every second image intensity along both dimensions of the picture. The four subbands are then calculated, and these four subbands are then concatenated back together to form a new array. This array is then returned to give the 1-level Haar Transform.

3b. Using the Matlab function that you have written to implement the 1-level Haar Transform, and using your assigned quantisation step size, calculate the resulting Entropy H_{qhaar} of the transformed image after quantisation has been applied.
The entropy H_{qhaar} was equal to 2.3545.

3c. Is $H_{qharr} < H_{qi}$? Why / Why not?
The entropy H_{qhaar} was less than the entropy H_{qi}. This is because the image associated with the entropy H_{qhaar} was quantised and then transformed using the Haar Transform, whereas the image associated with the entropy H_{qi} was just quantised. Similarly to quantisation, the Haar Transform has the desirable effect of decreasing the entropy of an image, but not by much. However, the Haar Transform in combination with quantisation is much more effective. This combination maximises the probability of a few symbols. As a result, this decreases the entropy. This is because the entropy of an image is minimised when a few symbols are much more probable than the others.

3d. To check the quality of the image after compression, the only thing to do is to reconstruct the picture using the calcInvHaar function. Compress and reconstruct your image using stepsizes of Qstep/2, Qstep and 2 × Qstep and comment on the differences you see in the pictures compared to the original image and the quantised image from Section 2.

The reconstructed images with stepsizes of $Q_{step}/2$, Q_{step} and 2 x Q_{step}, are shown in Figure 4 – 6 respectively. It is evident from these figures that the images have been reconstructed perfectly. Figure 5 is identical to Figure 3 in Section 2. This is because both images are quantised. In fact, it is evident from each of these figures that each of the images have been quantised. This can be deduced from the staircase effect that is visible across the girl's forehead. This effect is a result of the larger set of input values in the original image being mapped to the smaller set of output values in Figures 4 - 6. This effect is most evident in the images with a larger stepsize because the image intensities in the original image are being mapped to a smaller set of image intensities than the images with smaller stepsizes. The image intensities in the original image range from (0, 1, 2, … , 255) whereas the set of image intensities in Figure 4-6 range from (0, 7.5, 15, … , 255), (0, 15, 30, … , 255) and (0, 30, 60, 240) respectively.

Figure 4. Q_step/2 = 7.5

Figure 5. Q_step = 15

Figure 6. Q_step x 2 = 30

3e. Calculate the entropy of the transformed and quantised images and the mean squared error between each reconstructed image and the original image. Comment on the relationship between the entropy and the objective quality metric for the 3 quantisation step sizes.

The entropy of the transformed and quantised images as well as the mean squared error between each of the reconstructed images and the original image are shown in table 1. It is evident from this table that the entropy is inversely proportional to the mean squared error. The reason for this is the larger the stepsize value, the larger the rounding error will be as a result of quantisation. Thus, the larger the mean squared error will be.

	Entropy	**Mean Squared Error**
Q_step/2	3.1761	4.5988
Q_step	2.3545	18.4985
2 x Q_step	1.6835	70.0834

Table 1. With Haar Transform

3f. Using the same quantisation step sizes, calculate the MSE for the case where quantisation is performed on the image directly (ie. no haar transform is applied). Does the MSE metric correctly rank the perceived quality of the 6 compressed images (3 quantisation levels with/without transformation into the Haar domain)? Explain your answer.

The mean squared error between each of the compressed images and the original image are shown in table 2. It is evident from this table that the MSE ranks the perceived quality of the six images correctly. This is because images with a smaller stepsize would be expected to rank higher than the images with a larger stepsize. The reason for this is images with a smaller stepsize are closer to the original image than the images with a larger stepsize. Furthermore, the images that were transformed and reconstructed would rank equally to the images that had not been transformed and reconstructed. This is because the Haar Transform is a completely reversible transform.

	With Haar Transform	**Mean Squared Error**
Q_step/2	No	4.5988
Q_step/2	Yes	4.5988
Q_step	No	18.4985
Q_step	Yes	18.4985
2 x Q_step	No	70.0834
2 x Q_step	Yes	70.0834

Table 2. Without Haar Transform

4. The multi-level 2D Haar Transform

4a. Write a MATLAB function that implements the n-level Haar Transform and outputs an image of its subbands.

This function was implemented by modifying the level-1 Haar Transform function. There are only three modifications that were made to this function. Firstly, the level-n Haar Transform function takes an integer, n, which represents the number of levels of the Haar Transform as an input. Secondly, this function then verifies that n is an integer that is greater than or equal to 0. Thirdly, this function then recursively calls the Haar Transform on the Lolo subband. This continues until n = 1, or the width or height of the image cannot be evenly divided any further.

4b. Calculate the multi-level Haar Transform of your image and the reconstructed image for level numbers from 1 to 5 using your quantisation step size. Calculate the entropy of the quantised transformed image. Comment on the how the number of levels chosen affects the entropy and image quality.

The entropy of the quantised and transformed images are shown in table 3. It is evident from this table that the entropy is inversely proportional to the number of the level of the Haar Transform. That is, up until a certain point. In this example, that point is level-5. It is also evident from Table 3 that the entropy increases slightly after level-5 and remains constant after that point. Further experimentation from levels 6-8 can be conducted to confirm this. This suggests that indefinitely increasing the level of the transform will not indefinitely decrease the entropy of the image; only up until a certain point.

Level	**Entropy (Transformed and Quantised)**	**Entropy (Reconstructed)**
1	2.3545	3.5215
2	1.5606	3.5215
3	1.6048	3.5215
4	1.6352	3.5215
5	1.6418	3.5215

Table 3. Entropy of Quantised and Transformed Images and Reconstructed Images

References:

Shannon C.E. (1948) A Mathematical Theory of Communication. *Bell System Technical Journal*. **27**(3), 379–423, 623–656