Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

4C8 – DIGITAL IMAGE AND VIDEO PROCESSING

Lab 3 – Haar Transform

Department of Electrical and Electronics

(e-Report submission)

Table of Content

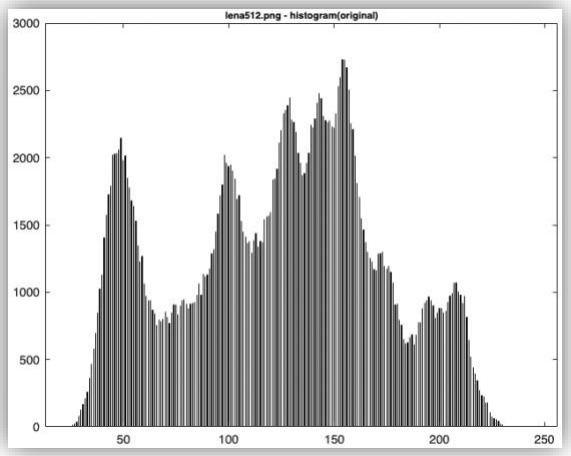
Submitted by:

Rohan Taneja
19323238

Submission: 20/03/2021

1. Image Entropy

1.1. Implement calcEntropy function

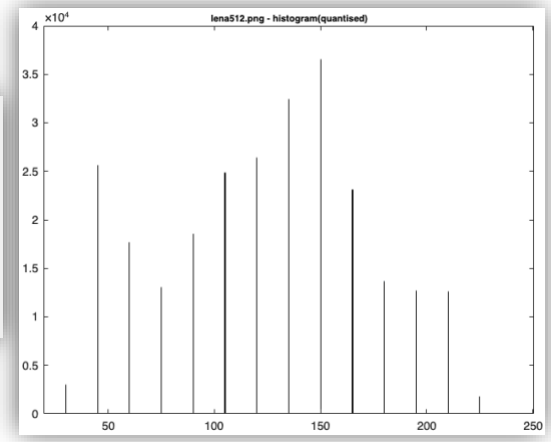



```
%1.1
function entropy = calcEntropy(Y)
% This function takes as input a 2D array Y containing
% the image intensities and returns the entropy.

% Z - stores normalized 2048 bins ranging in min-to-max value of input
% image. Take input as histcounts
    Z = histcounts(Y, 2048, 'Normalization', 'probability');
    H = Z .* log2(Z);
    H(isnan(H)) = 0;
    entropy = -sum(H);
end
```

Code snippet 1.1. - calcEntropy function

1.2. Quantise Image & Calculate Entropy




```
% 1.2
Q_step = 15;
Q_pic = Q_step .* round(I/Q_step);

Hqi = calcEntropy(Q_pic)

figure; imshow(Q_pic/256); axis image; title(file + ' - quantised',
figure; histogram(Q_pic,2048); title(file + ' - histogram(quantised)
```

1.3. Differentiate H_o and H_{qi}

```
%1.3 - entropy for original picture
Ho = calcEntropy(I)
```

```
Ho =

    7.4474
```

```
Hqi =

    3.5798
```

As we can see in the entropy values. After quantisation, the definition of pixels in the image reduces. Therefore, it results in decrease entropy.

1.4. Compute PSNR between images

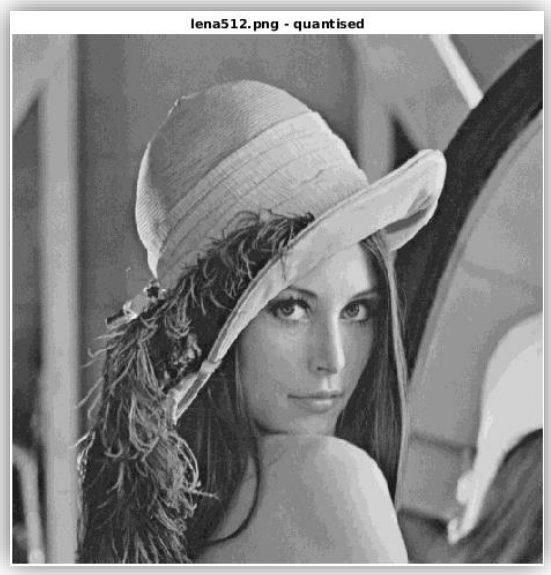
The peak signal to noise ratio is calculated as peak_snr.

```
%1.4
peak_snr = psnr(Q_pic,I,256)
```

```
peak_snr =

   35.3810
```

1.5. Visual Comparison



When an image is quantised, a reduction in the number of colours or grayscales in the resulting image is observed. Which in turn results in loss of definition of pixels and the banding colours.

The comparison is clearly visible in context of contrasting lena512.png for original on the left side and quantised image on the right.

2. The 2D Haar Transform

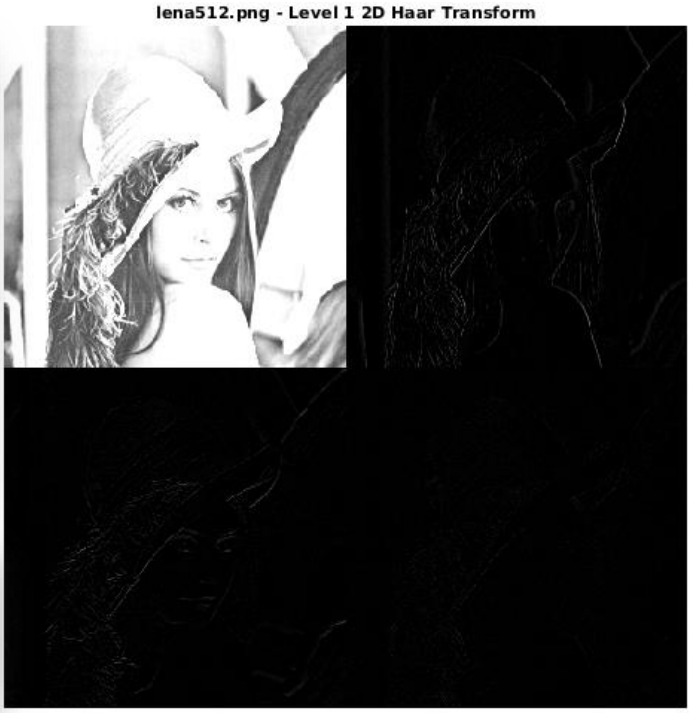
2.1. Implement 1-Level Haar Transform Function

```
function H =  calcHaarLevel1 (Y)
    % check dimensions
    if mod(size(Y,1), 2) ~= 0
      error('height must be multiple of 2');
    end
    if mod(size(Y,2), 2) ~= 0
      error('width must be multiple of 2');
    end

    % calculate respective a,b,c,d
    a = Y(1:2:end, 1:2:end); b = Y(1:2:end, 2:2:end);
    c = Y(2:2:end, 1:2:end); d = Y(2:2:end, 2:2:end);

    % H = [lo-lo hi-lo]
    %     [lo-hi hi-hi]
    H = [(a + b + c + d) (a - b + c - d);
        (a - c + b - d) (a - b - (c - d))]/2;
end
```

```
%2.1
pic = double(imread(file));
HaarT = calcHaarLevel1(pic);
figure; imshow(HaarT/256); axis image;
```



Output image of Haar Transform Level 1 sub bands generated. The visible of other bands can be contrasted or visible when seen from different viewing angle.

2.2. Quantising Haar Transformed Image & Calculate Entropy

```
%2.2
Q_step = 15;
Qhaar_pic = Q_step .* round(HaarT/Q_step);

Hqhaar = calcEntropyHaar(Qhaar_pic,1)
```

```
Hqhaar =

    1.7500
```

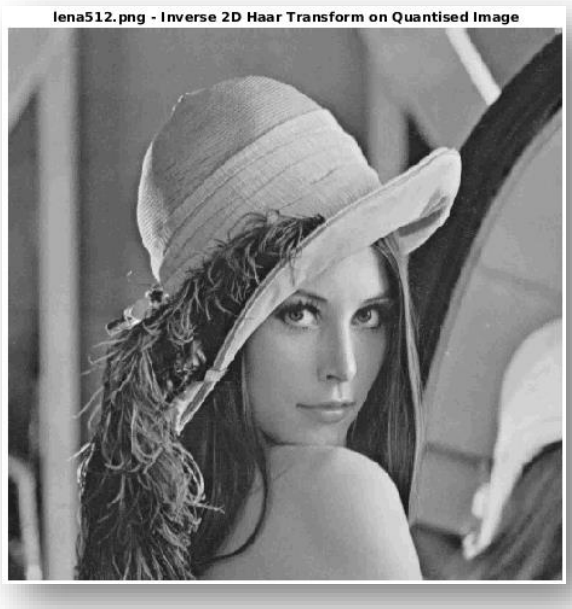
In the code snippet 2.2 we apply quantisation of step 15 to Haar Transform of the original image and calculate the entropy of generated image.

2.3. $H_{qhaar} < H_{qi}$ – Reasoning

H_{qi} is the entropy for quantisation with step equal to 15 of the original image which is turn is the same step used for the Haar transform image. The brighter pixels in Haar Transformed image as shown in section 2.1 lies in Lo-Lo sub band. Hence, the entropy is highly effected from a quarter size of the image resulting in lower numerical value.

2.4. Decoded Haar Image – Comparison

```
%2.4
invHaar_pic = calcInvHaar(Qhaar_pic,1);
figure; imshow(invHaar_pic/256); axis image;
```



As we can see the decoded image, there is a slight loss in pixels resulting in increasing sharpness of the overall image. The dark zones are shifted slightly to more darker colour which is noticeable in chin and shoulder area of foreground of lena512.png. In original image, the skin tone transition is smoother and less sharper.

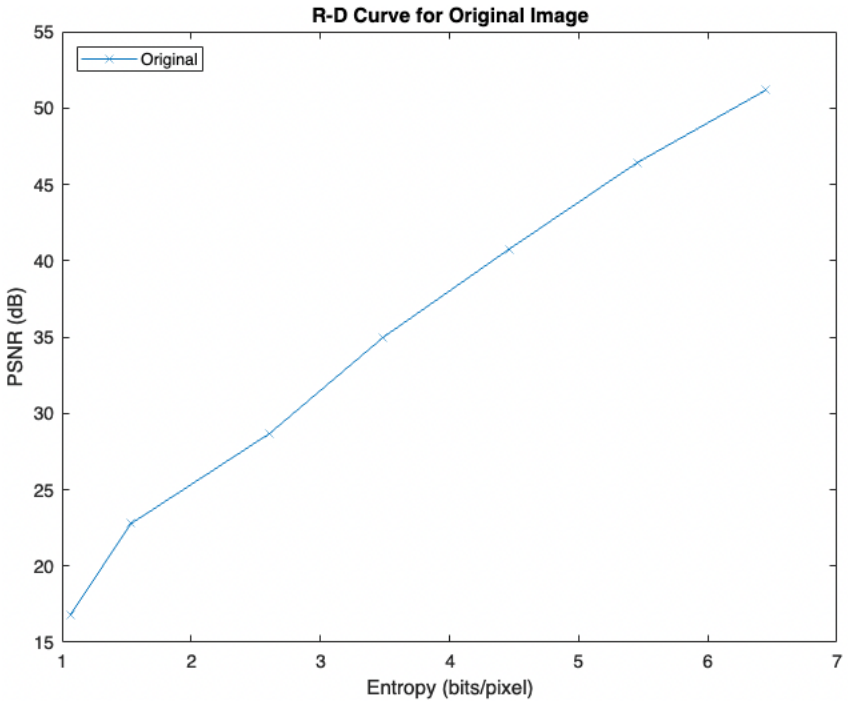
3. Rate-Distortion Curve

3.1. Plot R-D curve for the original image

```
%3.1
e = [];
p = [];

for Q_step = [2 4 8 16 32 64 128]
    Q_pic = Q_step .* round(I/Q_step);
    e = [e;calcEntropy(Q_pic)];
    p = [p;psnr(Q_pic,I,256)];
end

figure; plot(e, p, '-x'); xlabel('Entropy (bits/pixel)');
ylabel('PSNR (dB)');title('R-D Curve for Original Image');
legend('Original', 'Location','northwest', 'Orientation','horizontal');
```



The figure above project R-D Curve for the original image with no transformation applied. The plot is for 7 different quantisation steps applied on the image v/s entropy.

3.2. Plot R-D curve for Haar transformed

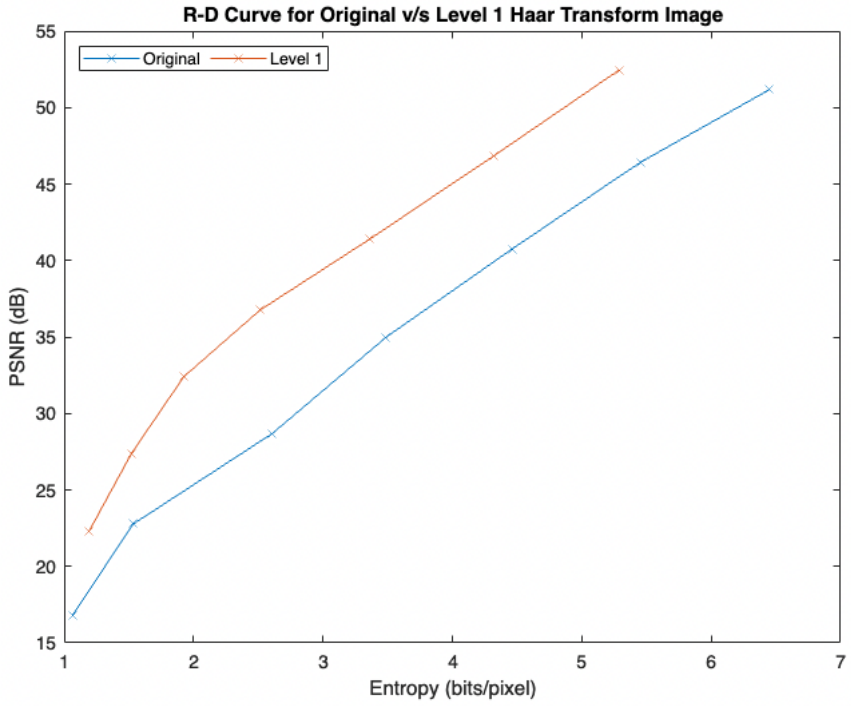
```
%3.2
e_haar = [];
p_haar = [];

for Q_step = [2 4 8 16 32 64 128]
    HaarT = calcHaarLevel1(I);
    Qhaar_pic = Q_step .* round(HaarT/Q_step);
    InvHaarT = calcInvHaar(Qhaar_pic,1);

    e_haar = [e_haar;calcEntropy(Qhaar_pic)];
    p_haar = [p_haar;psnr(InvHaarT,I,256)];
end

figure;
plot(e, p,'-x', e_haar, p_haar, '-x');
xlabel('Entropy (bits/pixel)'); ylabel('PSNR (dB)');
title('R-D Curve for Original v/s Level 1 Haar Transform Image');
```

In the code snippet 3.2 we apply quantisation on Haar Transformed image and regenerate the image with inverted Haar transform applied on quantised Haar Transformed images. The resulting plot is shown as follows specified in the legend:



3.3. Comment on the Plot

The quantised images without application of Haar transform corresponds to lower distortion with increasing entropy. The image generated with quantisation of original image is of better quality than with Haar Transform. Due to loss of details in Haar transformed image where the sub bands coefficients are lying close numerically. The only effect in terms of quality of quantised original image is loss of grayscale pixels.

For **lossy compression** – Haar Transformed Image is better as higher PSNR tends to better results for media delivery.

For **quality** – Quantised Original Image as elaborated in detail above.

4. The Multi-Level 2D Haar Transform

4.1. Implement calcHaar for n-levels

```
function H = calcHaar(Y, n)
% This function takes as input a 2D array Y containing
% the image intensities of a picture and returns the 1-level
% Haar Transform
% n is the number of levels used.

% calculate level 1 Haar transform
H = calcHaarLevel1(Y);
if n == 1
  return
end

% for loop - call HT on lolo band till end of image arrays
for i = 1 : n-1
  % split lolo sub-band
  lolo = H(1 : (2 .^ -i) * end, 1 : (2 .^ -i) * end);
  % apply Haar L1 on lolo sub-band
  H_lolo = calcHaarLevel1(lolo);
  % swap results in original array
  H(1 : (2 .^ -i) * end, 1 : (2 .^ -i) * end) = H_lolo;
end

end
```

In the code snippet 4.1 – implemented the calcHaar function to apply n-levels of Haar Transform on an input Y image. The code is written as instructed to used Level 1 implementation and apply recursive transformation on Lo-Lol sub-band.

After testing the implementation. Results obtained are as follows -

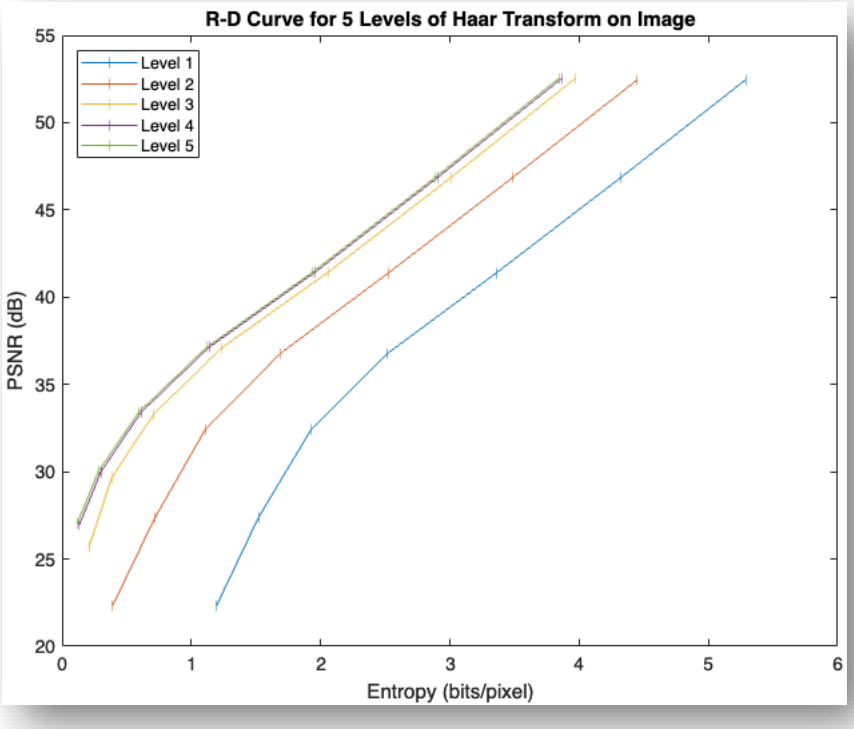
```
>> testHaar
calcHaarLevel1: OK
calcHaar: OK
```

4.2. R-D Curve for Haar Transform on 1-5 Levels

```
%4.2
figure; x=1; y=7;
for i = (1:5)
    for Q_step = [2 4 8 16 32 64 128]
        HaarT = calcHaar(I,i);
        Qhaar_pic = Q_step .* round(HaarT/Q_step);
        InvHaarT = calcInvHaar(Qhaar_pic,i);

        e_haar = [e_haar;calcEntropyHaar(Qhaar_pic,i)];
        p_haar = [p_haar;psnr(InvHaarT,I,256)];
    end
    plot(e_haar(x:y*i), p_haar(x:y*i), '-|'); hold on; x=x+y;
end
```

In code snippet 4.2 – we introduce a for-loop to calculate rate and distortion values up to 5 levels storing them in their respective and plotting a curve onto a single graph.



As the level of Haar Transform increases, the quality of the images also becomes crisp, details are more visible and comparable to the original image. As the entropy increases, height of range of PSNR is higher with increasing level. Which makes it comparably more compressible and there is overall less distortion. As soon as Level 5 is reached it somewhat overlaps with the previous level. With increasing level the picture quality is impacted positively despite lowering entropy and hence, it is beneficial overall..