Artificial Intelligence
Exercises week 7 Solution- Computer Vision

Maryam Hashemi

COMP3411/9814

Question 1: Histogram Equalization

Consider an image **X** with $N = 8$ pixels and $L = 4$ intensity levels (0 to 3). The intensity values and corresponding pixel counts are provided in the table below. Calculate the equalized histogram and show all of your calculations.

Intensity Level (i)	**Pixel Count** ($f(i)$)
0	1
1	2
2	3
3	2

Answer: First we need to calculate the histogram probabilities ($p_x(i)$):

Intensity Level (i)	**Pixel Count** ($f(i)$)	**Probability** ($p_X(i) = f(i)/8$)
0	1	1/8
1	2	2/8
2	3	3/8
3	2	2/8

With having $p_x(i)$, we can find Cumulative Distribution Function (CDF). The CDF is $cdf_X(i) = \sum_{j=0}^{i} p_X(j)$.

Intensity Level (i)	$p_X(i)$	$\mathbf{cdf_X(i)}$
0	1/8	1/8
1	2/8	$1/8 + 2/8 = 3/8$
2	3/8	$3/8 + 3/8 = 6/8$
3	2/8	$6/8 + 2/8 = 8/8 = 1$

The transfer function maps the input levels i to the output levels k using the scaled CDF. The formula is $T(i) = \text{round}\left(\frac{cdf_X(i) - cdf_{min}}{1 - cdf_{min}} \times (L-1)\right)$, where $L - 1 = 3$ and $cdf_{min} = 1/8 = 0.125$.

Input Level (i)	$\mathbf{cdf_X(i)}$	**Output Level** $\mathbf{T}(i)$ (k)
0	1/8	round(0) = **0**
1	3/8	round(0.875) = **1**
2	6/8	round(2.143) = **2**
3	8/8	round(3.0) = **3**

Equalized Output Histogram Y: The pixel counts are transferred to the new levels, resulting in the equalized histogram.

Output Level (k)	**Pixel Count** ($f_Y(k)$)
0	1
1	2
2	3
3	2

Interpretation: The original counts (1, 2, 3, 2) were clustered, meaning the image had low contrast. By using the CDF as the transfer function, the output histogram is significantly more uniform (closer to the ideal uniform count of $N/L = 2$ per level), achieving **histogram equalization** and improving contrast.

Question 2: Averaging

Consider the binary image with dimension 7×16 shown in Figure 1. Use the averaging method with a threshold $\epsilon = 3$ and a 3×3 sliding windows. Show the resulting image.

Answer: White boxes represent a value 0, black boxes represent a value 1. We need to slide a window through the image and replace the middle pixel

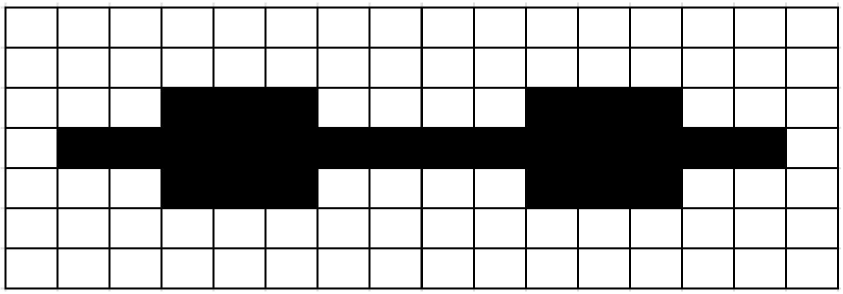

Figure 1: Original image.

with a 0 or a 1 depending on if the number of pixels with value 1 are greater or not than the threshold $\epsilon = 3$ (if sum $> \epsilon \to 1$, otherwise 0). If we repeat the process, the result should look like the one shown in Figure 2.

Question 3: Convolution

A digital image (I) is given to you as follows. We apply a convolution operation with a kernel presented as K with **1 layer of zero padding**, stride equals to 2 for width and stride 1 for height, and max pooling with 2×2 filters. Finally, pass it through a flattening layer. Calculate the final output.

$$I = \begin{pmatrix} 255 & 100 & 100 & 0 \\ 10 & 100 & 200 & 0 \\ 200 & 255 & 0 & 10 \\ 10 & 0 & 30 & 50 \end{pmatrix}$$

$$K = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 10 & 2 \\ 0 & 2 & 0 \end{pmatrix}$$

Answer:

We apply convolution with zero padding of 1, stride $s_h = 1$ (height) and $s_w = 2$ (width), followed by max pooling with 2×2 filters and finally flattening.

Step 1: Zero Padding: Adding 1 layer of zeros around I gives:

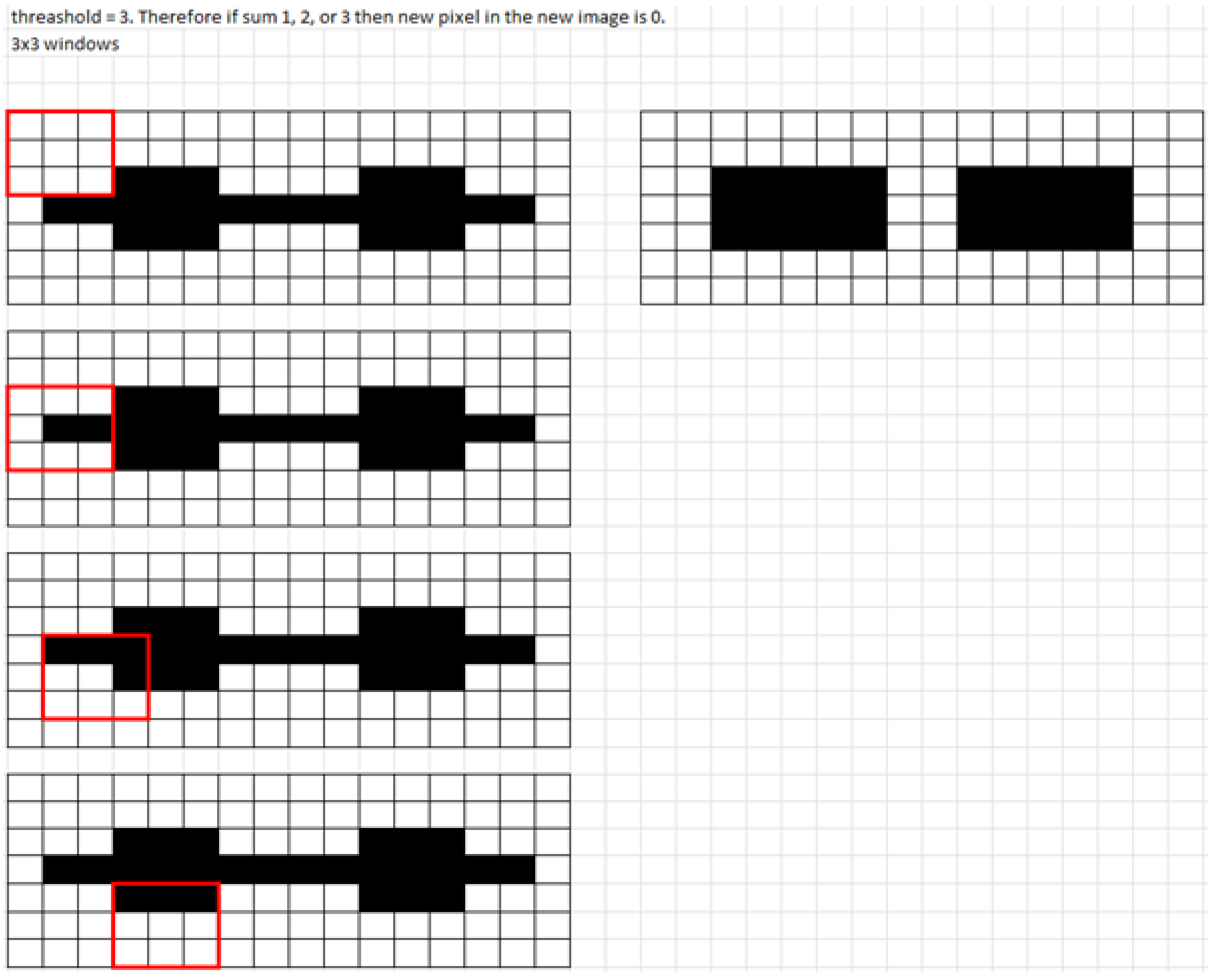
threashold = 3. Therefore if sum 1, 2, or 3 then new pixel in the new image is 0.
3x3 windows

Figure 2: Resulting image.

$$I_p = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 100 & 100 & 0 & 0 \\ 0 & 10 & 100 & 200 & 0 & 0 \\ 0 & 200 & 255 & 0 & 10 & 0 \\ 0 & 10 & 0 & 30 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Step 2: Convolution

The output of convolution at position (i, j) is:

$$O[i,j] = \sum_{m=0}^{2} \sum_{n=0}^{2} K[m,n] \cdot I_p[i+m, j+n]$$

$$O[0,0] = 0*0+2*0+0*0+2*0+10*255+2*100+0*0+2*10+0*100 = 2770$$
$$O[0,1] = 0*0+2*0+0*0+2*100+10*100+2*0+0*100+2*200+0*0 = 1600$$
$$O[1,0] = 0*0+2*255+0*100+2*0+10*10+2*100+0*0+2*200+0*255 = 1210$$
$$O[1,1] = 0*100+2*100+0*0+2*100+10*200+2*0+0*255+2*0+0*10 = 2400$$
$$O[2,0] = 0*0+2*10+0*100+2*0+10*200+2*255+0*0+2*10+0*0 = 2550$$
$$O[2,1] = 0*100+2*200+0*0+2*255+10*0+2*10+0*0+2*30+0*50 = 990$$
$$O[3,0] = 0*0+2*200+0*255+2*0+10*10+2*0+0*0+2*0+0*0 = 500$$
$$O[3,1] = 0*255+2*0+0*10+2*0+10*30+2*50+0*0+2*0+0*0 = 400$$

Thus, the convolution output is:

$$O = \begin{pmatrix} 2770 & 1600 \\ 1210 & 2400 \\ 2550 & 990 \\ 500 & 400 \end{pmatrix}$$

Step 3: Max Pooling (2×2, stride 2):

1. Window 1 (rows 0–1, cols 0–1): $\begin{pmatrix} 2770 & 1600 \\ 1210 & 2400 \end{pmatrix} \Rightarrow \max = 2770$
2. Window 2 (rows 2–3, cols 0–1): $\begin{pmatrix} 2550 & 990 \\ 500 & 400 \end{pmatrix} \Rightarrow \max = 2550$

$$P = \begin{pmatrix} 2770 \\ 2550 \end{pmatrix}$$

Step 4: Flattening
Flattening the pooling output gives the final output:

$$\text{Output} = \begin{bmatrix} 2770 & 2550 \end{bmatrix}$$