Image processing and analysis 00460200

HW 2 Wet, due to 12/06/25

Lavie Lederman 319046504

Shahar Yankelovich 206159527

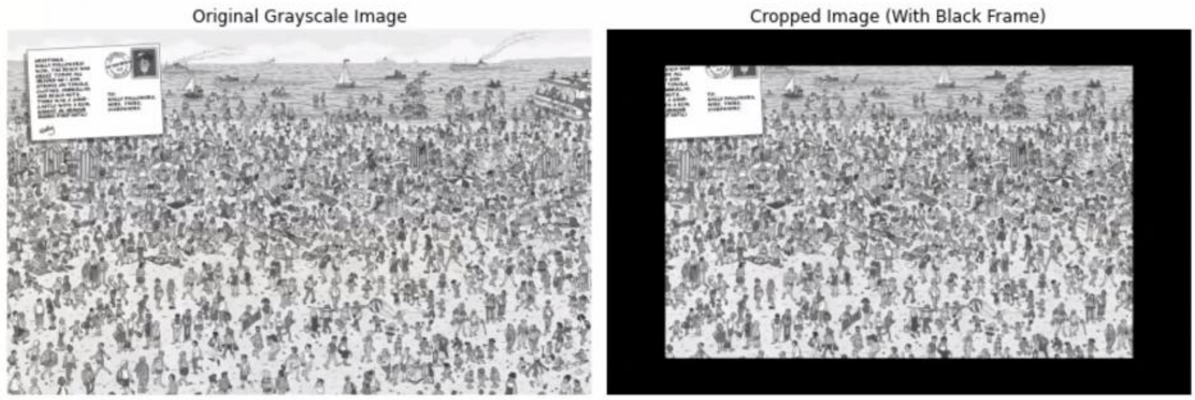
**Part 1**

**1.A**

After normalizing the image, we use filter2D to calculate the correlation between the given pattern and the image.

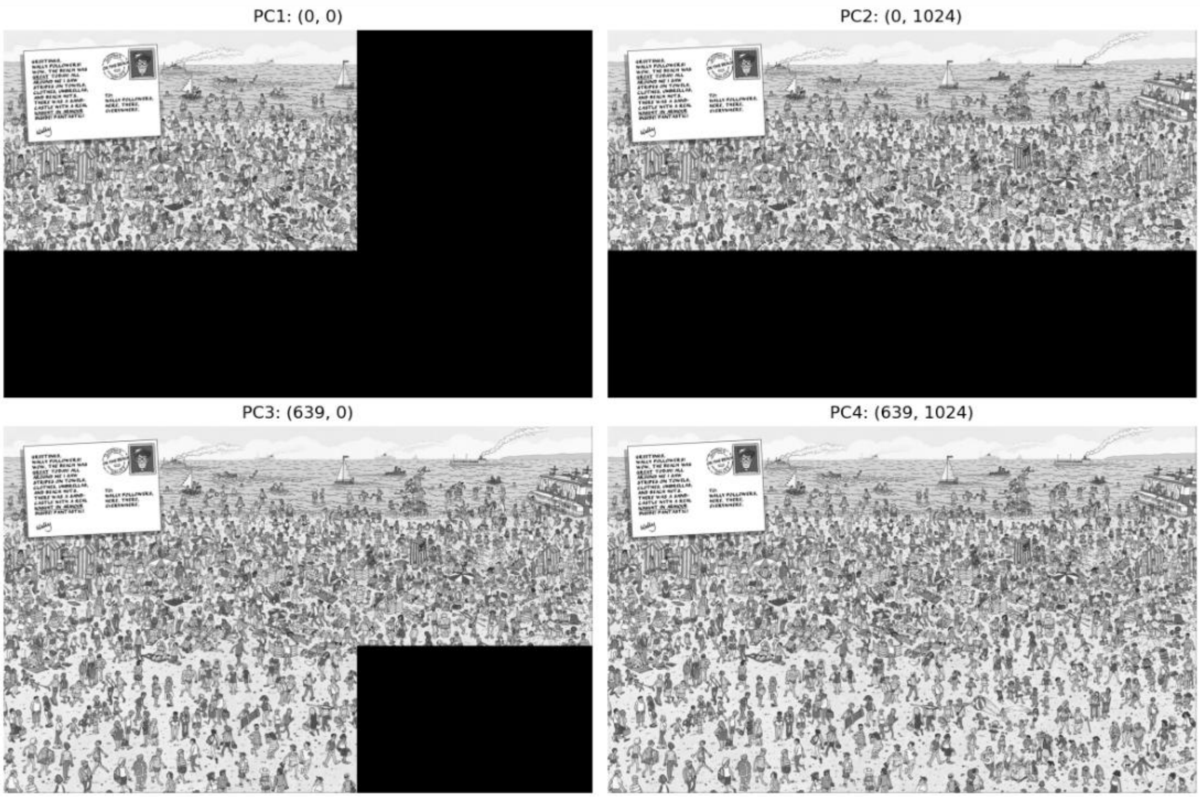
**1.B**

Here are the gray pictures – with frame and without frame –



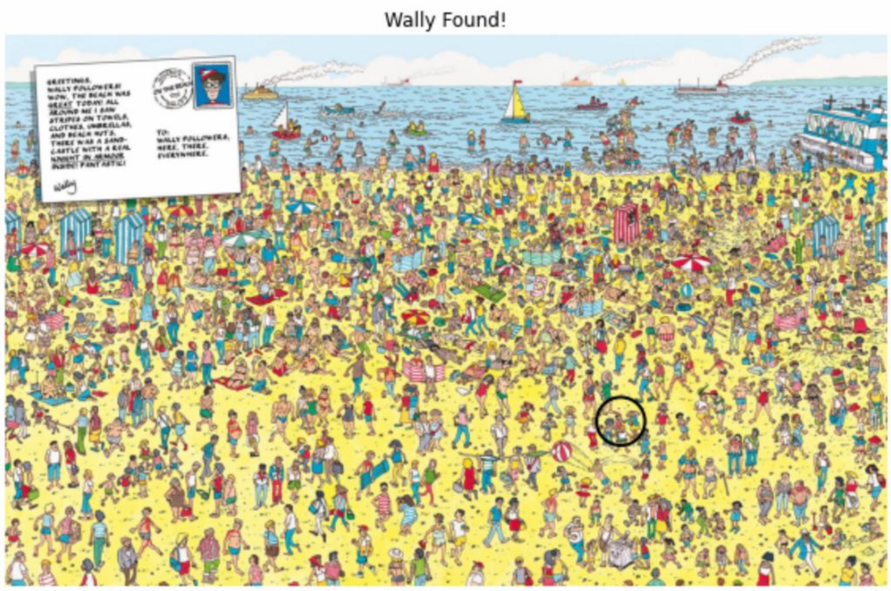

**1.C**

Here are the pictures with the locations –




**1.D**

Here is Wally marked with a circle –



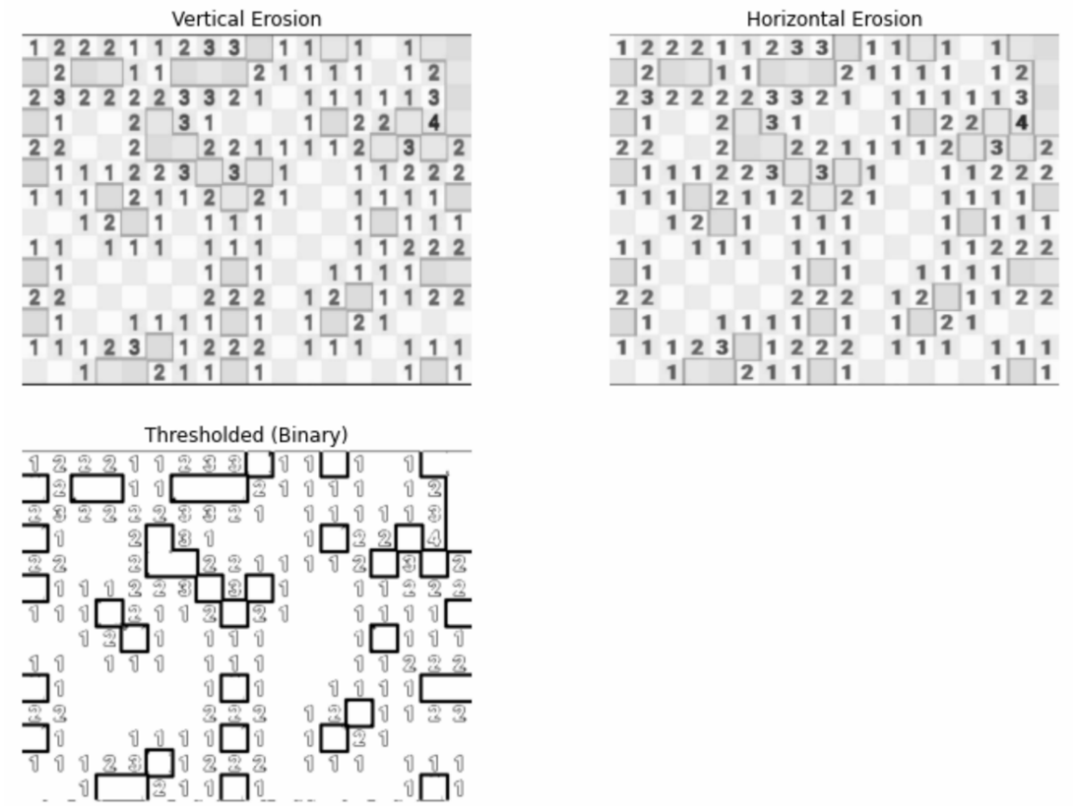

Zoomed in –

The circle was done using “patches.circle” with a radius the size of the larger dimension of the “Wally” image.

**Part 2**

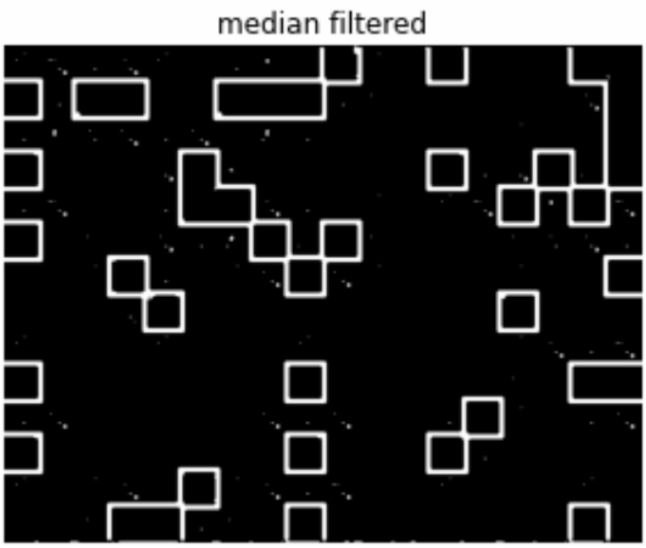
**2.A**

Here are the results of erosion with both kernels separately –



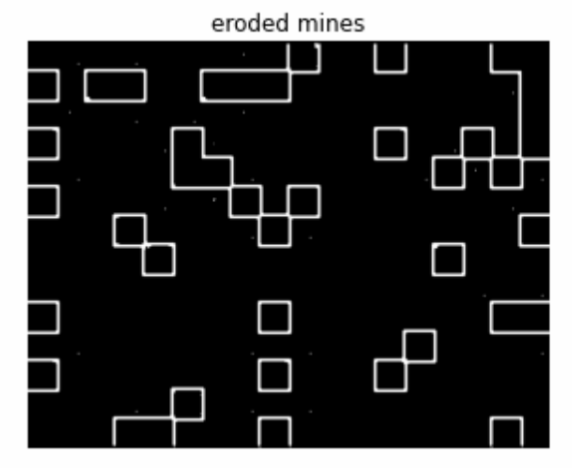

We can see that these operations preserve the general structures in the images – the numbers, the blocks etc. yet, we can notice a slight smear of the numbers –for example, the “1” with the horizontal erosion seems slightly wider than the “1” with the vertical erosion.

2.b



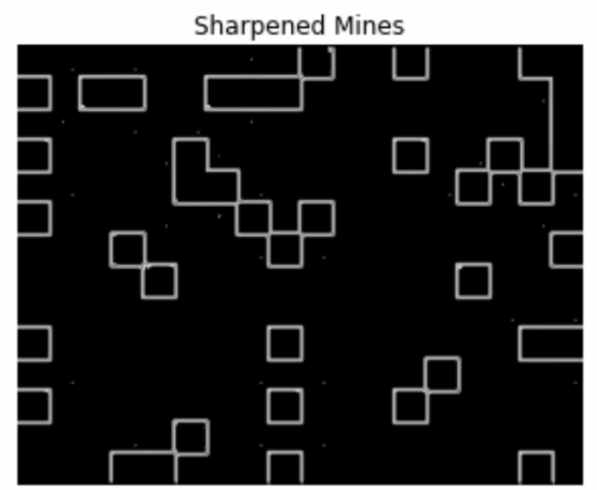
Median filter cleaned most of the numbers, indicating how many mines are nearby, leaving us with the mines themselves and some noise. Using a mean filter might have given us values that are between 0 and 1 – we have a binary image, so we want only 0 and 1.

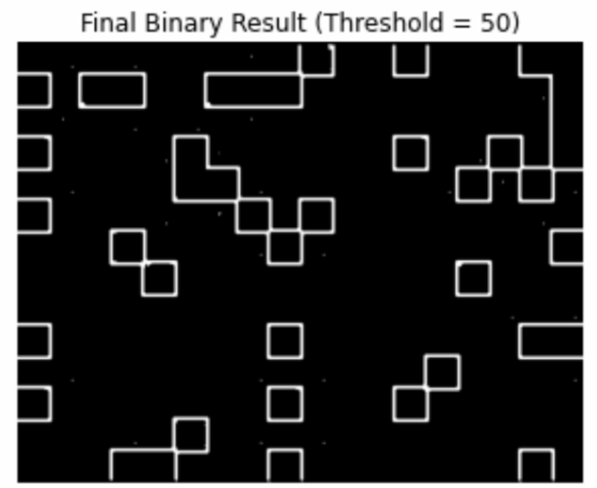
2.c




By using 2x2 filter and erosion we cleaned most (but not all) of the noise, and we are left only with the mines.

2.d



This kernel uses the image's properties and emphasizes the edges by enhancing the pixels that have neighboring opposite pixels. If a pixel has a neighboring pixel with the same value – the kernel will weaken its value, and if it has opposite value neighboring pixel, it will enhance its value. Since the first image in the section already has sharp edges, performing the correlation didn't do much (visually).

**Part 3**

**3.A**

1. Given a set of grayscale image distributions $p^\theta(x)$, each has an associated prior probability $\pi_\theta$, the mixture distribution is defined as –

$$\bar{p}(x) = \sum_{\theta=1}^{K} \pi_\theta p^\theta(x)$$

This is a valid probability distribution since for each x –

$$p^\theta(x) \geq 0, \pi_\theta \geq 0 \rightarrow \bar{p}(x) = \sum_{\theta=1}^{K} \pi_\theta p^\theta(x) \geq 0, for\ all\ x$$

Plus –

$$\sum p^\theta(x) = \sum \pi_\theta = 1$$

and if we sum all the options, we get 1 –

$$\sum \bar{p}(x) = \sum \sum_{\theta=1}^{K} \pi_\theta p^\theta(x) = \sum_{\theta=1}^{K} \pi_\theta \sum p^\theta(x) = \sum_{\theta=1}^{K} \pi_\theta \cdot 1 = 1$$

2. The minimal expected length of a symbol is calculated using the entropy –

$$L_{min} = \sum\nolimits_x p^\theta(X) \cdot \log_2 p^\theta(x)$$

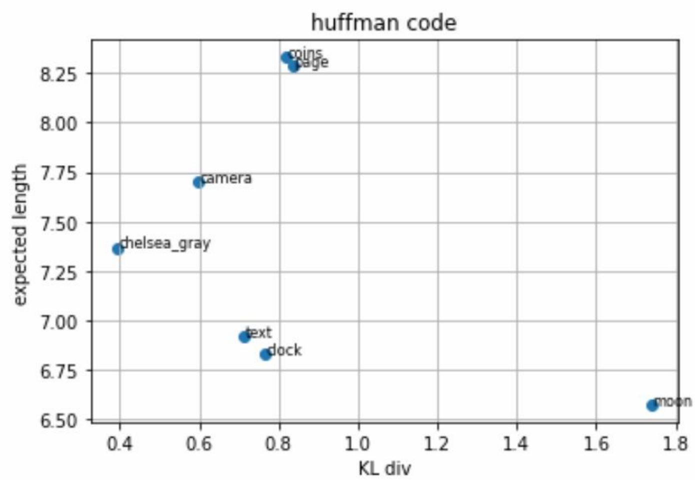
To find the overall expected code length across all images we will calculate the weighted sum of code lengths, which is weighted by the mixture distribution –

$$\mathbb{E}_\theta = \sum\nolimits_\theta \pi_\theta \cdot \left(\sum\nolimits_x p^\theta(x) \cdot l(x)\right) = \sum\nolimits_x l(x) \cdot \left(\sum\nolimits_\theta \pi_\theta \cdot p^\theta(x)\right) =$$
$$= \sum\nolimits_x l(x) \cdot \bar{p}(x)$$

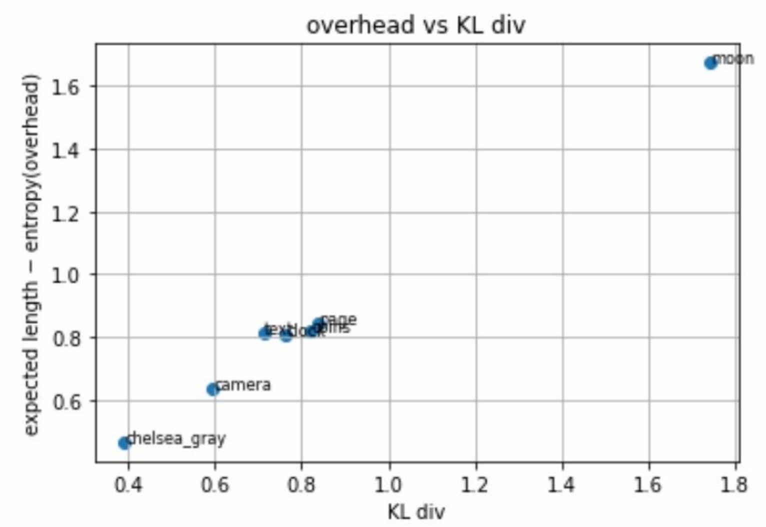
We used the mixture distribution definition from the previous section.

**3.B**

After applying the Huffman code on the dataset and doing the required calculations, at the beginning we got a non-linear result. We believe it stems from the images' varying properties (as we explained in the code).




After performing some manipulations, we can see in the following graph that the trend is indeed linear as expected – as KL increases, the code length increases too –


overhead vs KL div
expected length − entropy(overhead)
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
KL div
moon
page
text
camera
chelsea_gray