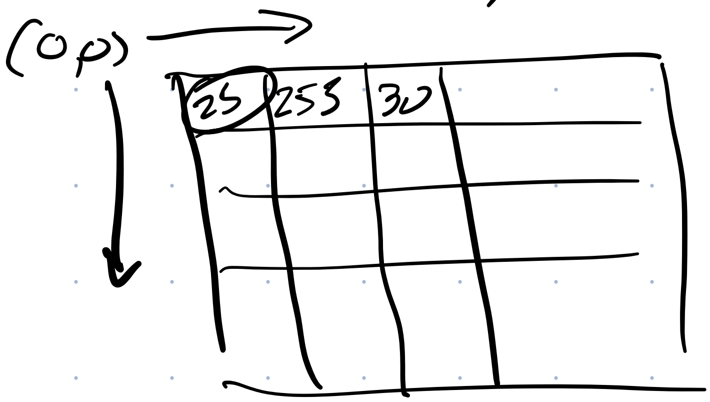
Image Processing - Chapter 3

- A <u>digital image</u> is composed of pixels each with finite, discrete numeric representation of its intensity (r, g, b) or grey level that is an <u>output</u> from its two-dimensional function fed as input by its spatial coordinates (x, y) or (i, j).



- <u>Binary Image / 1-Bit</u> - is an image that consises of pixels that have exactly two colors. (Black/white)
    - Stored as 0 or 1.

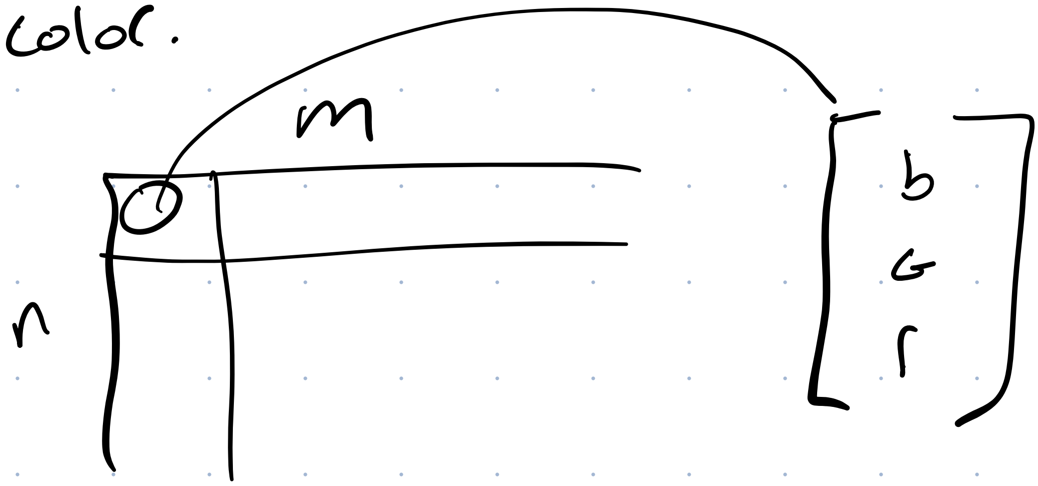
- Gray Scale - is an image where each pixel is a single sample representing only an amount of light / intensity information.
    - Each pixel ranges from 0 - 255
        - 0 → Black
        - 255 → white

- Color image represents multispectral data acquired through the visible domain.
    - For classical color images, this domain is reduced to three channels, one for each primary color red, green, blue.

    - Each pixel is composed of three numerical values
      range - 0-255

    - Each value represents the

brightness of each primary color.

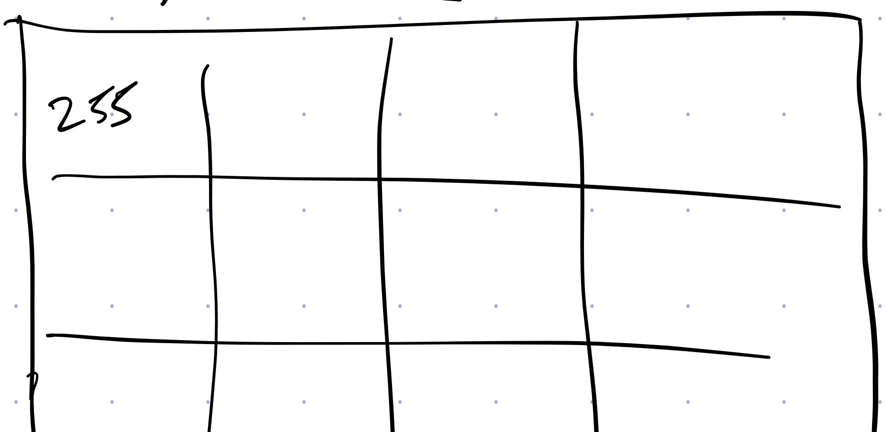


$$\begin{bmatrix}255\\0\\0\end{bmatrix} \longrightarrow \text{blue}$$

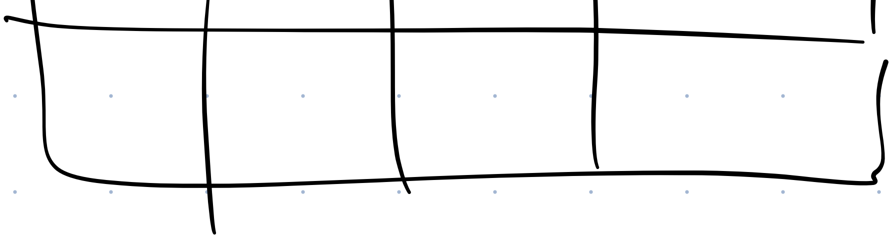
$$\begin{bmatrix}0\\255\\0\end{bmatrix} \longrightarrow \text{Green}$$

$$\begin{bmatrix}255\\255\\0\end{bmatrix} \longrightarrow \text{Yellow}$$

Grey Scale

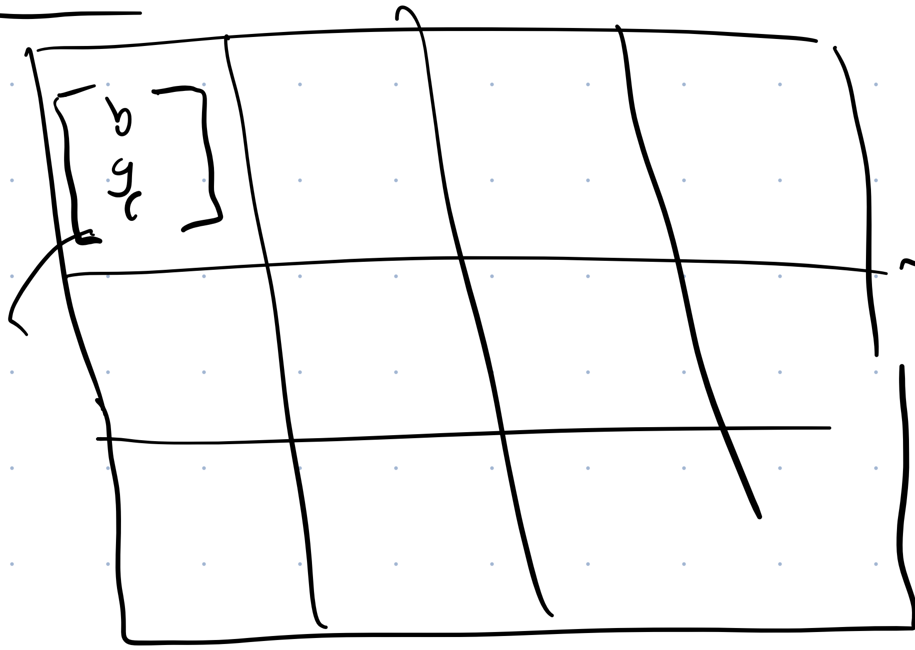


$\longrightarrow$ 2D array



(1-channel)

Color



→ 3D array

(3-channel)

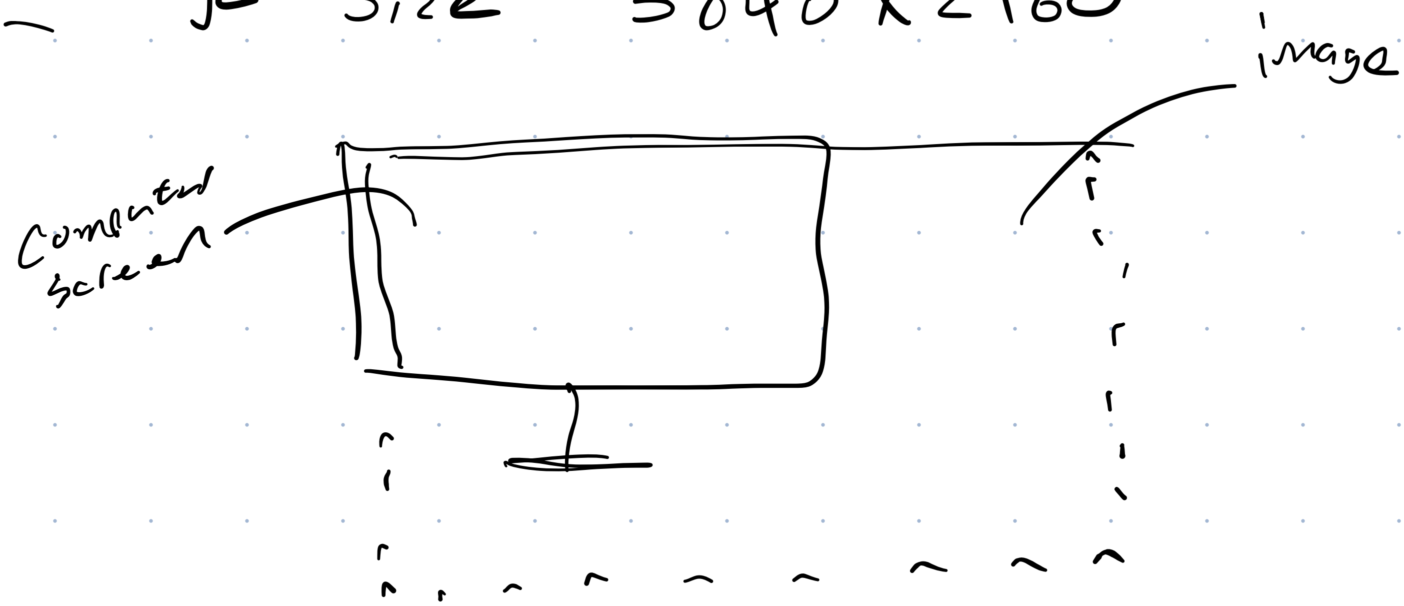
- **Image resolution** – is the count of pixels displayed horizontally and vertically

(1920 x 1080)

1920

1080

- Screen size 1920 x1080
- image size 3840 x 2160



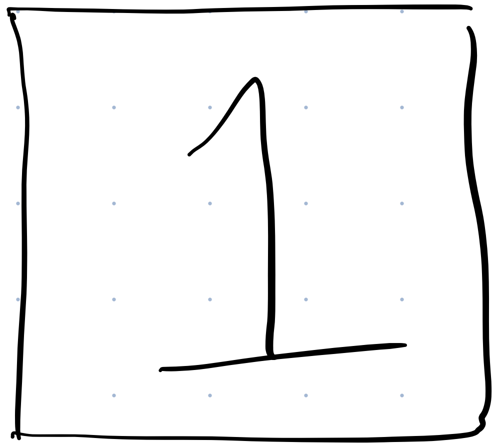
- **aspect ratio** - describes the proportional relationship between h x w.

i.e $1920:1080 \rightarrow 16:9$ aspect ratio

$640:480 \rightarrow 4:3$

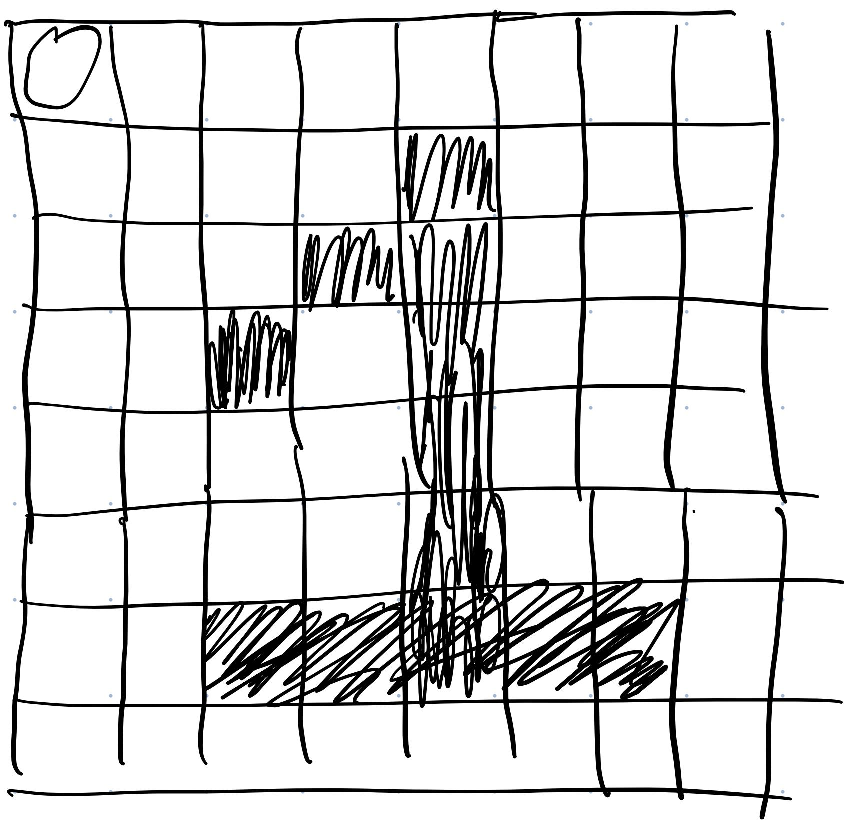
$800:600 \rightarrow 4:3$

$960:720 \rightarrow 4:3$



- Binary image → 0/1

8 x 8 image



↓

1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	1
1	1	1	0	0	1	1	1

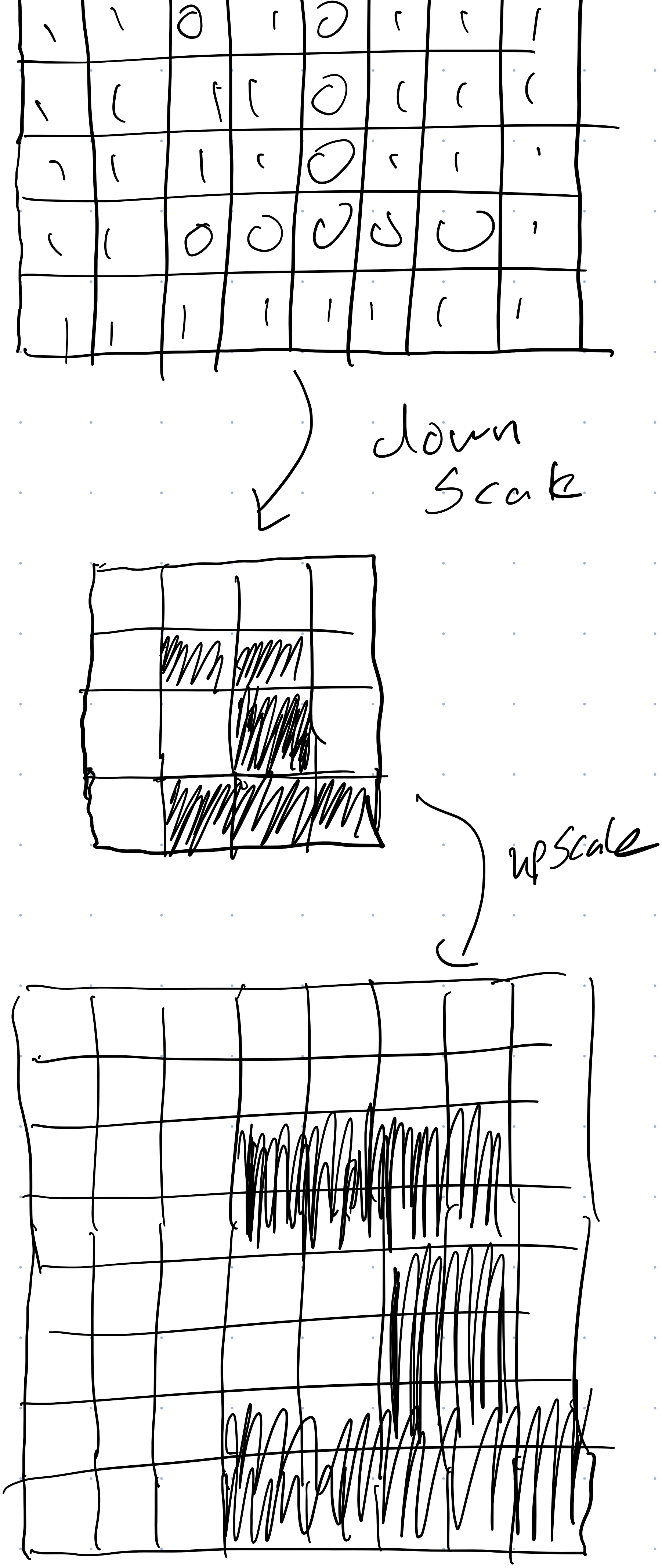

down
Scale
upScale

Image Processing Operators

- is a function that takes one or more input images and produces an output image

$$g(x) = h(f(x))$$

$$\downarrow$$

$$g(i,j) = f(i,j)$$

- x original image
- i and j are pixel location

- add and subtract constants

$$g(i,j) = f(i,j) + b$$

Matrix size i x j of constant numbers

- Must be the same size!

• example – getting a negative of an grey scale image.

$$B'(x) = 255 - B(x)$$

original image

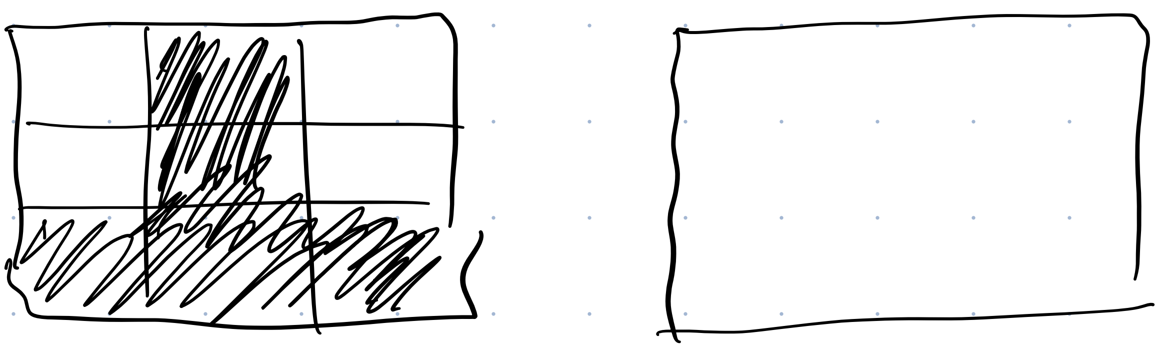
negative image

Matrix size i, j with all values set to 255

input image

255

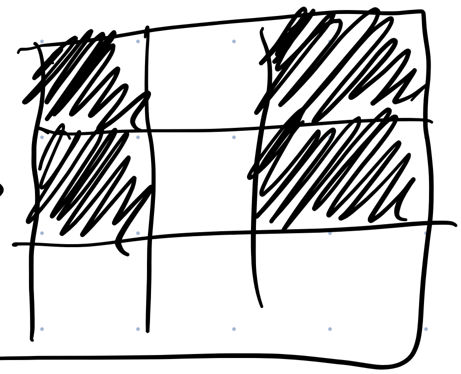
$$\begin{bmatrix} 255 & 0 & 255 \\ 255 & 0 & 255 \\ 0 & 0 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 255 & 255 & 255 \\ 255 & 255 & 255 \\ 255 & 255 & 255 \end{bmatrix}$$



$$255 - B(x)$$

$$\begin{bmatrix} 255 & 255 & 255 \\ 255 & 255 & 255 \\ 255 & 255 & 255 \end{bmatrix} - \begin{bmatrix} 255 & 0 & \\ 255 & 0 & 255 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0 & 255 & 0 \\ 0 & 255 & 0 \\ 255 & 255 & 255 \end{bmatrix} \rightarrow$$



How to Gray Scale a Colored image

Color image = $\begin{bmatrix} 0 & & \\ & & \\ & & \end{bmatrix}$

$\begin{bmatrix} b \\ g \\ r \end{bmatrix}$

- Average Method = $(r + g + b)/3$

$$\frac{255 + 255 + 255}{3} = 255$$

- NTSC formula

$$\underline{0.299} \cdot R + \underline{0.587} \cdot G + \underline{0.114} \cdot B$$

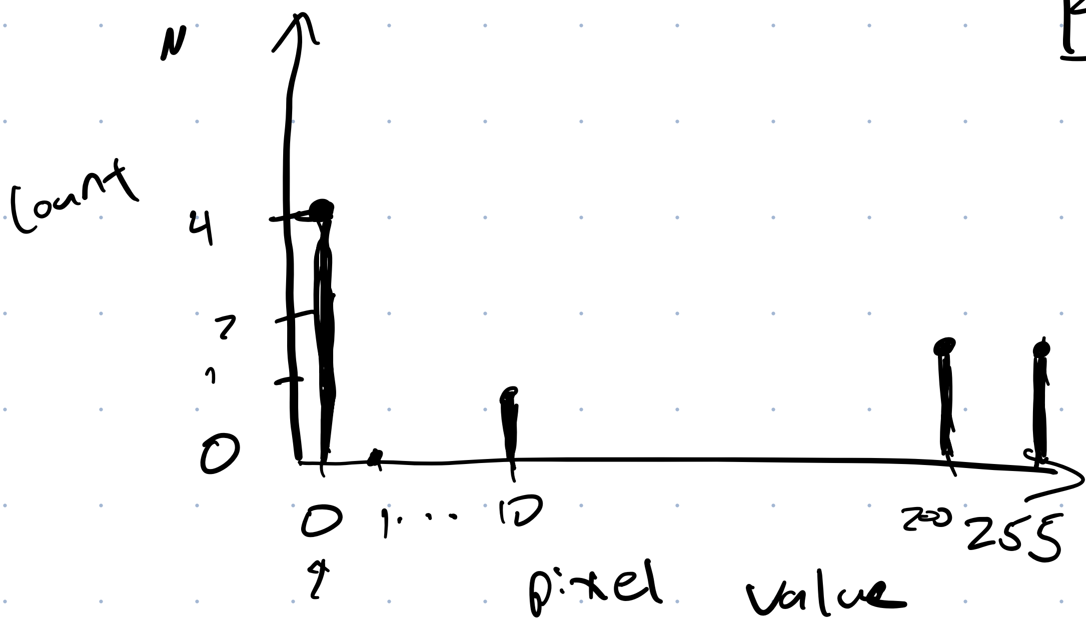
Histograms / Equalization

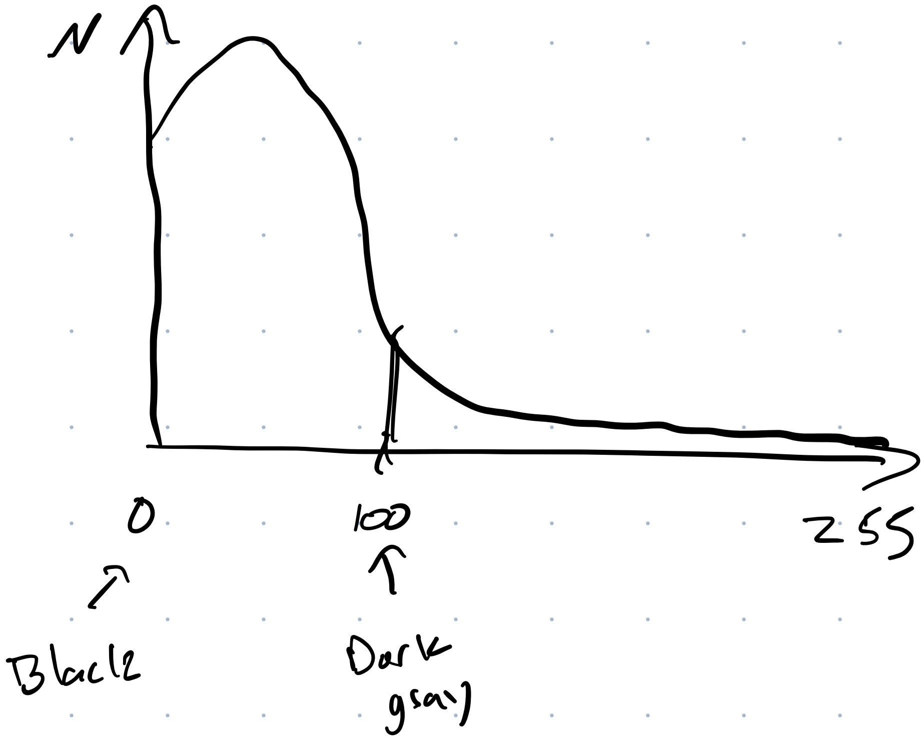
- recall that a Gray Scale image represents the amount of light/intensity of an image.

0 → black

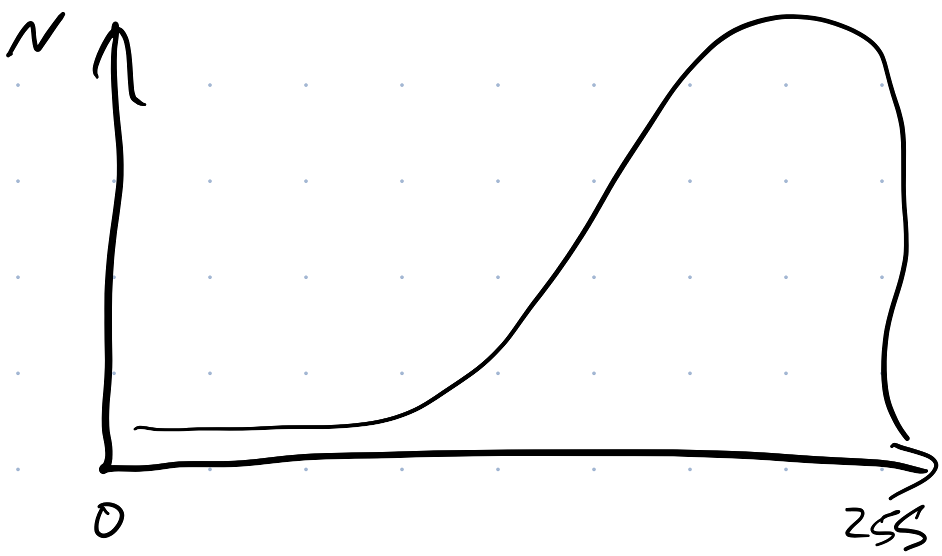
255 → white

0	0	0
10	0	200
255	255	200

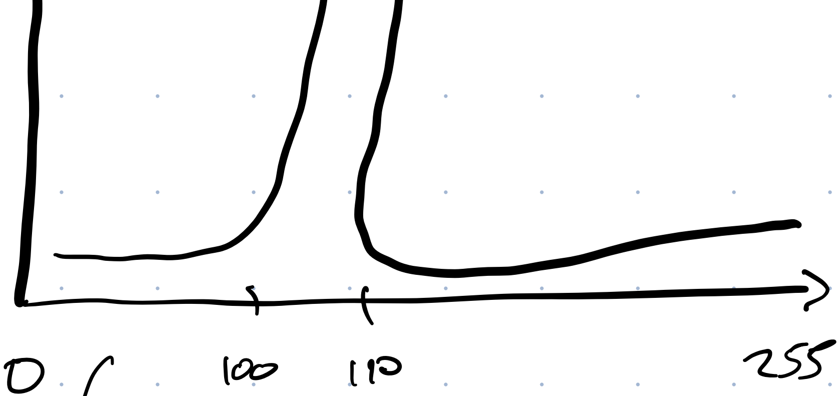




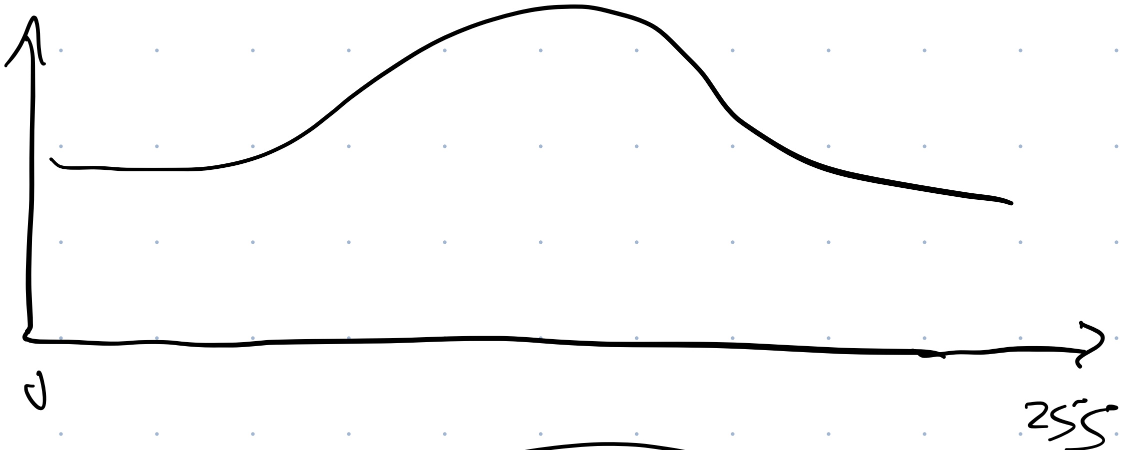
- Our image is going to be Dark!

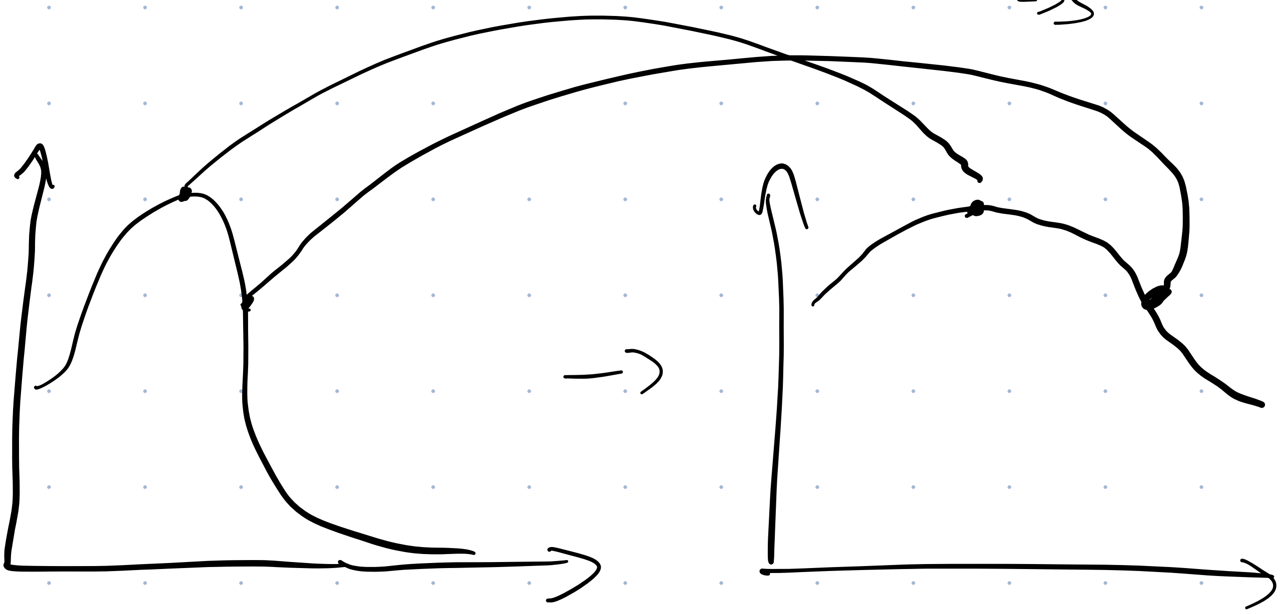


- Our image is going to be light!

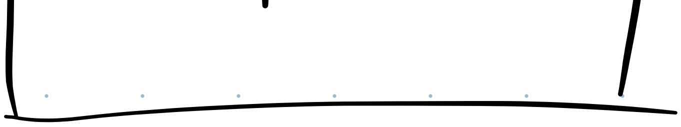


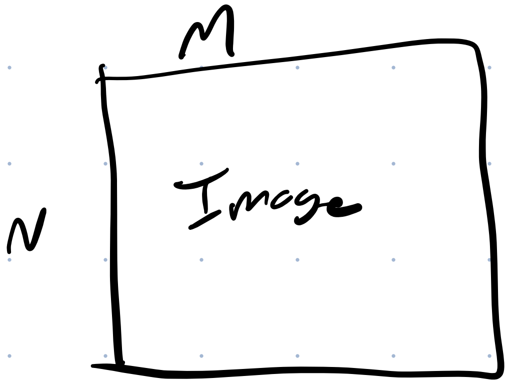
• really hard to see.





old Value	New value
10	20
100	150
110	170





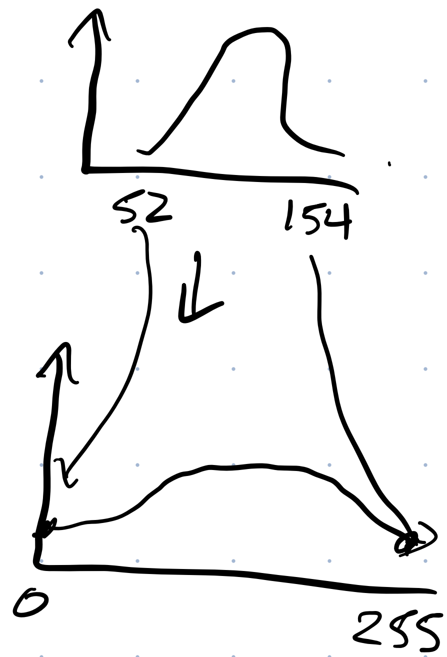
Gray Scale

Step 1 - get counts of each pixle (histogram)

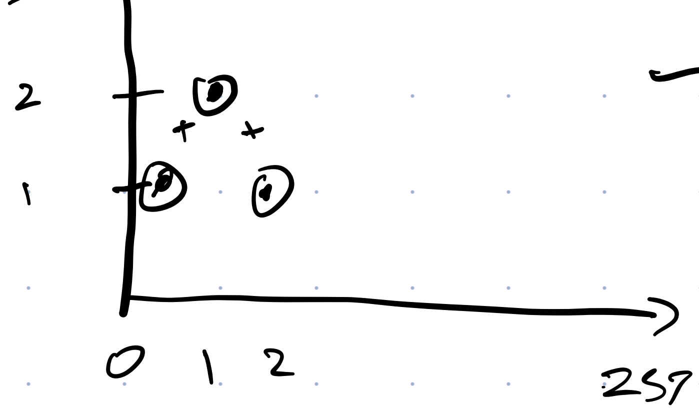
all pixel values in the image

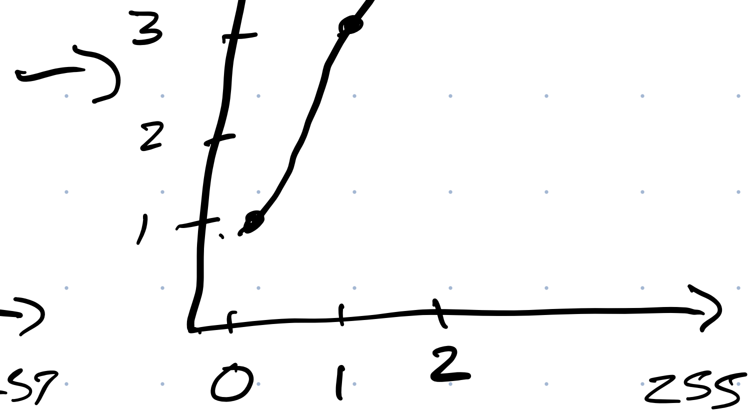
Value	Count	CDF(v)	Map(v)
52	1	1	0
55	3	4	12
58	2	6	
59	3	9	
⋮	⋮	⋮	
144	1	63	
154	1	64	255

total number of pixels.



Step 2 Calculate the cumulative distribution function (cdf)





Step 3 equalization formula

old value

$$map(v) = \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times \underline{255}$$

$M \times N$: # of pixels

cdf_{min} is the smallest non-zero value of the CDF

$$map(\underline{52}) = \frac{1-1}{64-1} \times 255 = \frac{0}{63} \times 255 = \boxed{0}$$

$$Map(55) = \frac{4-1}{63} \times 255 = \frac{3}{63} \times 255 = 12.1 \rightarrow \boxed{12}$$

$$Map(154) = \frac{64-1}{} \times 255 = \frac{63}{63} \times 255 = \boxed{255}$$

63 63