Görüntü İşlemede Mimari Detaylar

Batuhan Düzgün

Roadmap

1- Image Processing Introduction

2- History of Photography

3- Applications of Digital Image Processing

4- Industrial Applications

5- Image Processing Methods

6- Tesseract OCR Engine & OpenCV

7- Example Project: “sahibinden.com” Staff Identity Card OCR Project

8- Resources and Questions

Image Processing Introduction

Image Processing

1- Analog Image Processing

2- Digital Image Processing

Overlapping Fields

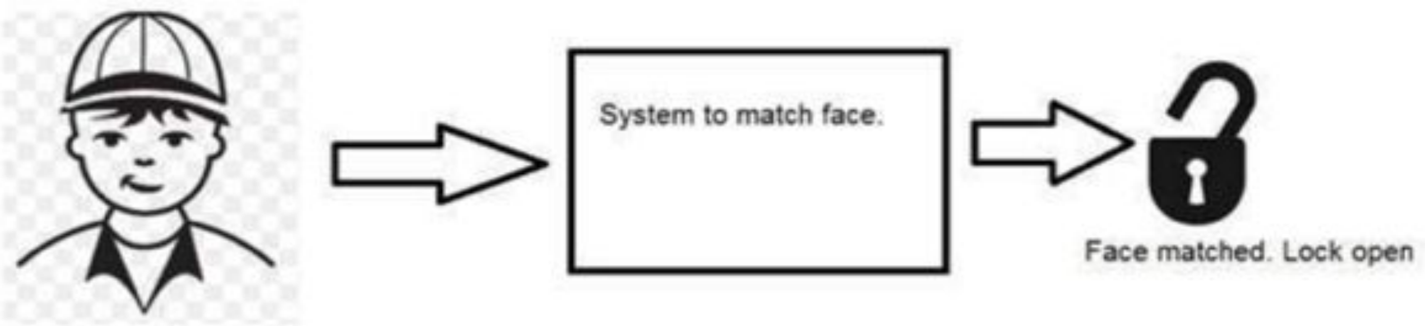
1- Computer Vision

2- Computer Graphics

3- Signal Processing

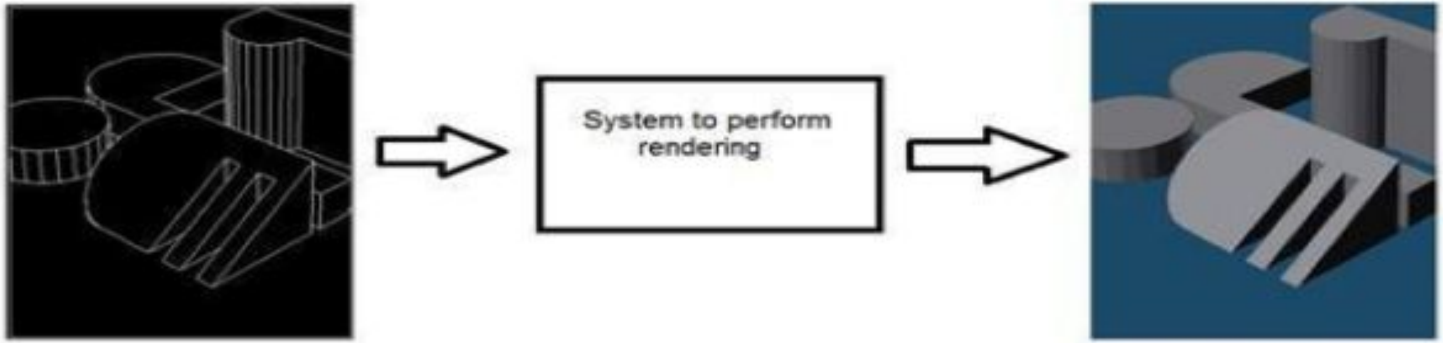
4- Artificial Intelligence

Overlapping Fields



System to match face.

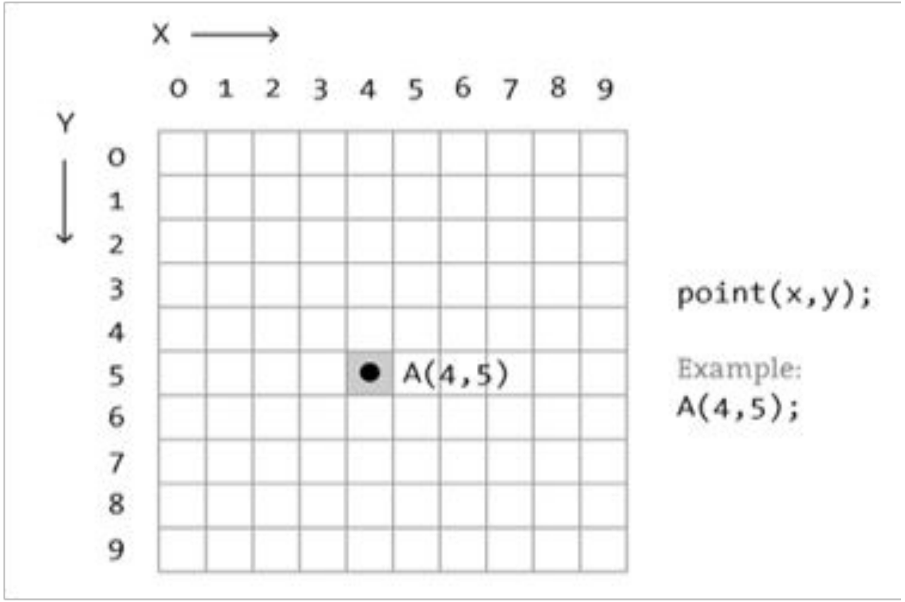
Face matched. Lock open



System to perform rendering

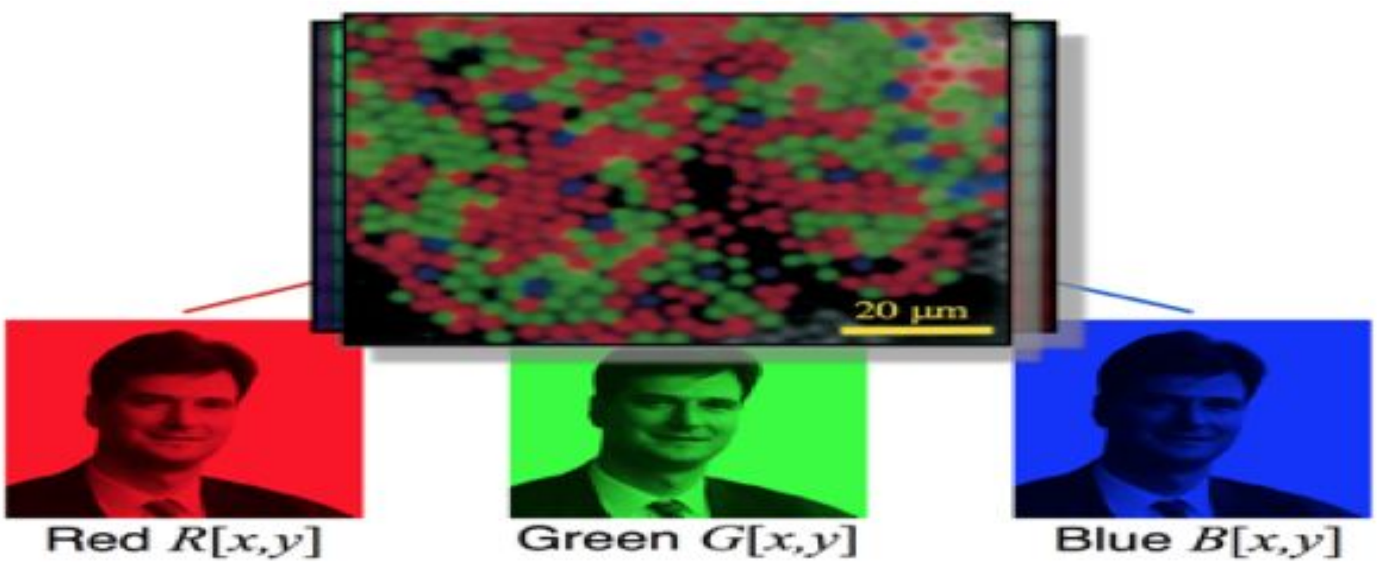
What is an Image ?

Mathematical function **f(x,y)** where x and y are the two coordinates horizontally and vertically.



Pixel ?

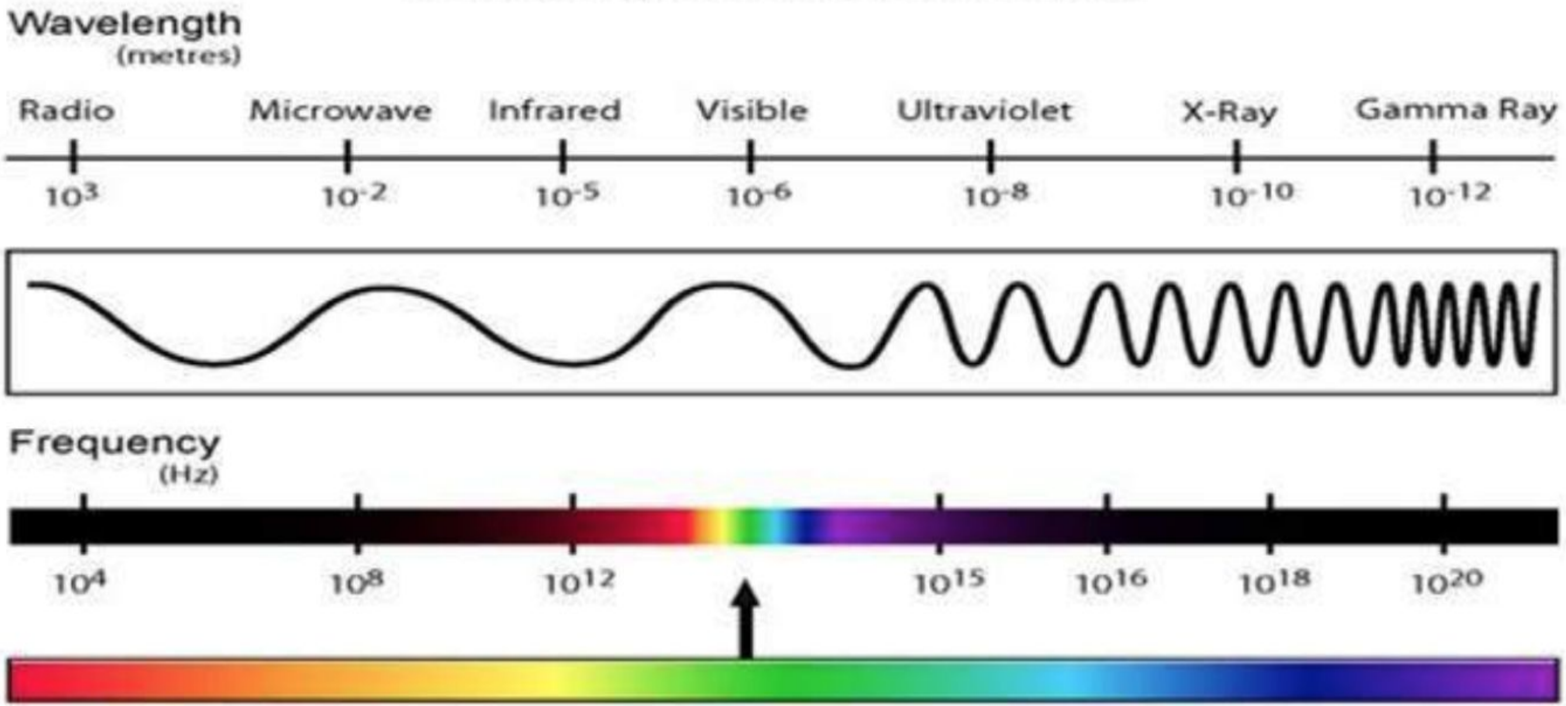
- Pixel is the smallest element of an image.
- In an 8-bit gray scale image, the value of the pixel between 0 and 255.



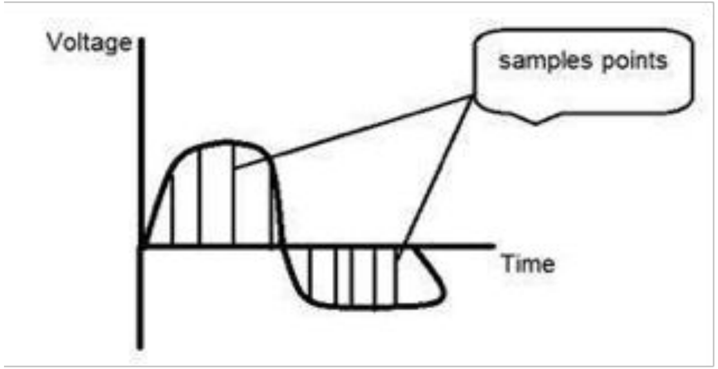
Pixel ?

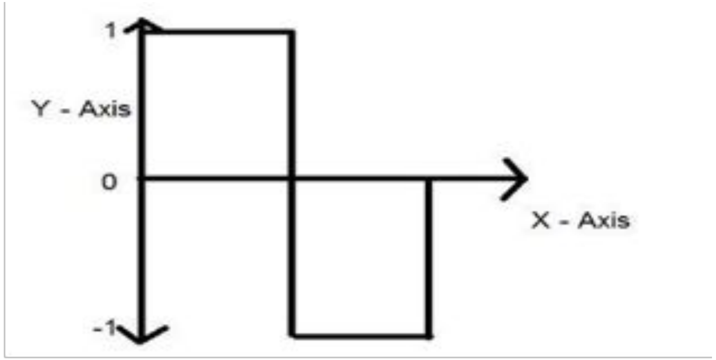
- Red Green Blue (RGB)
- Decimal Format: (0,255,0)
- Hex Format: #FFFFFF

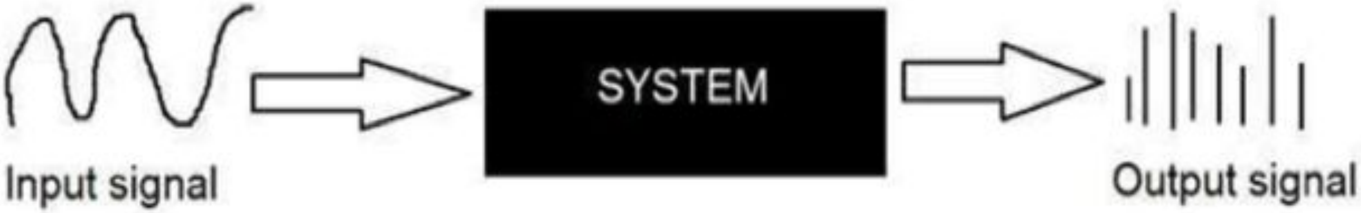
Electro Magnetic Spectrum



Signal Processing (Analog to Digital)







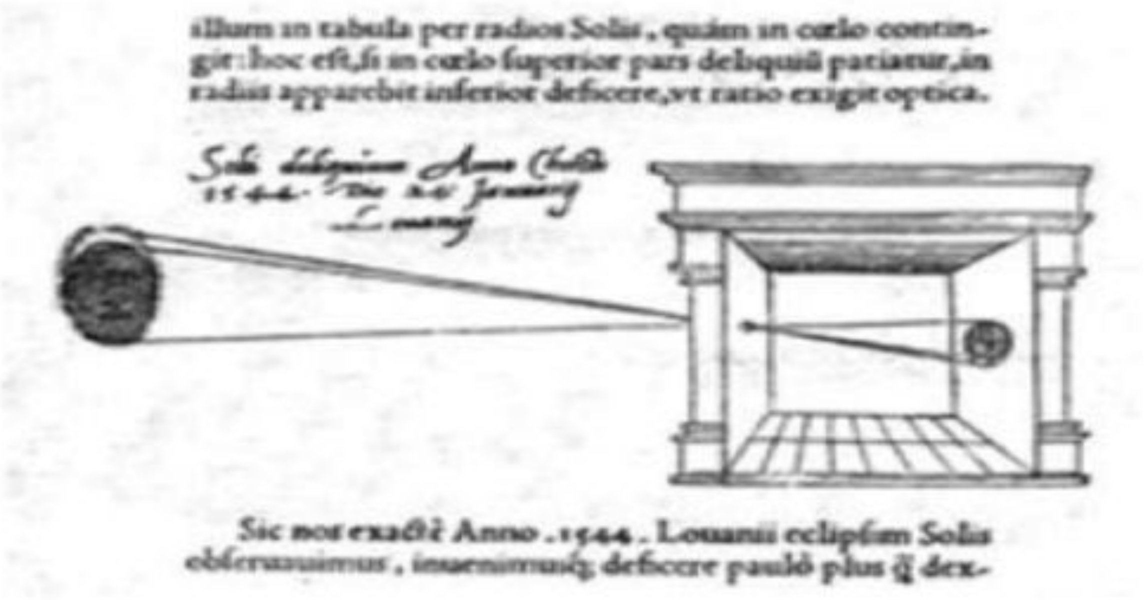
History of Photography

Camera Obscura

- The principles of the camera were first introduced by a Chinese philosopher **MOZI**.
- Ibn al-Haitham built the first camera obscura.

Da Vinci

- Many principles of camera obscura has been defined in **Codex Atlanticus**.



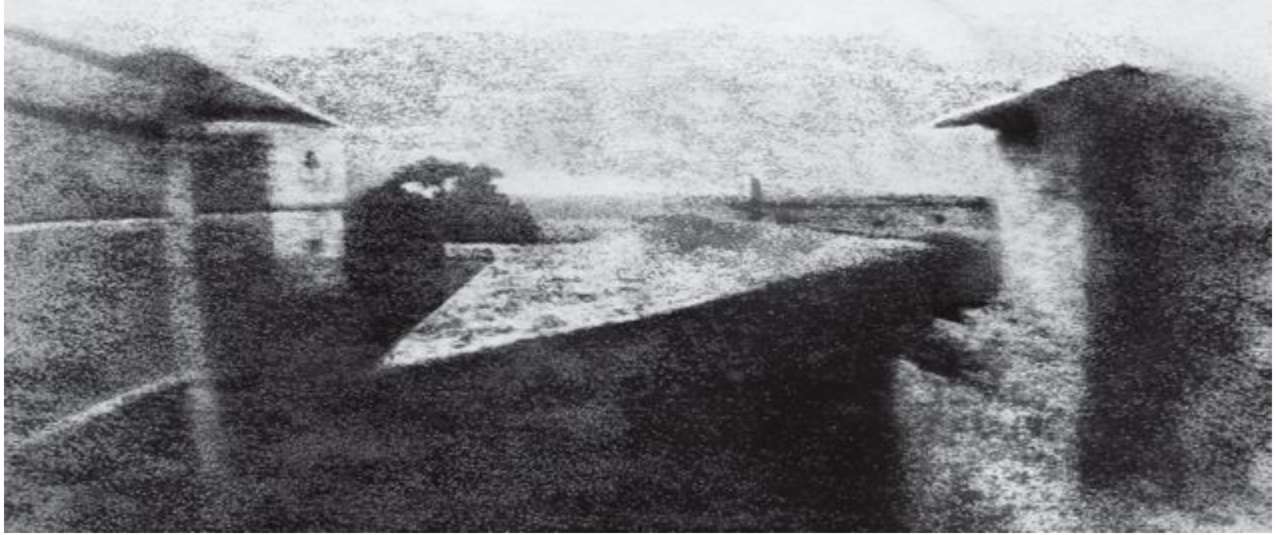
illum in tabula per radios Solis, quàm in cœlo contingit: hoc eſt, ſi in cœlo ſuperior pars deliquiũ patiatur, in radiis apparebit inferior deficere, vt ratio exigit optica.

Solis deliquium Anno Christi
1544. Die 24 Januarij
Louanij

Sic nos exactè Anno .1544. Louanii eclipſim Solis obſeruauimus, inuenimusq; deficere paulò plus q̃ dex-

First Photograph

- **The first photograph was taken in 1814** by a French inventor Joseph Nicephore Niepce. (Le Gras, France)



Analog and Digital Cameras

- Leica and Argus

Leica and argus are the two analog cameras developed in 1925 and in 1939 respectively.

- Digital Cameras

Mavica was launched by Sony in 1981

Applications of Digital Image Processing

Application Fields

- Medical field
- Machine/Robot vision
- Color processing
- Pattern recognition
- Video processing

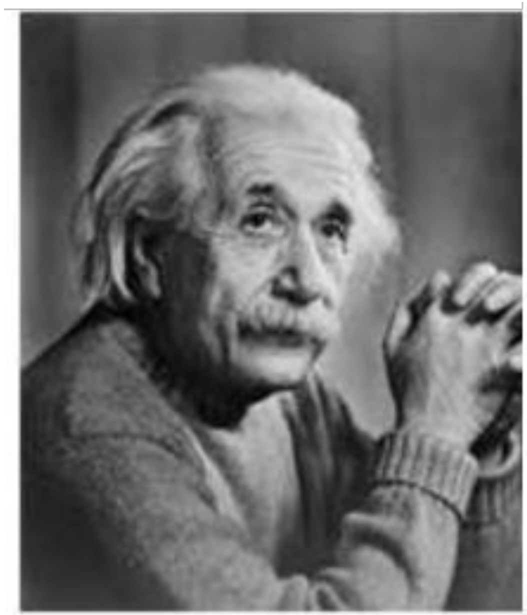
Industrial Image Processing and OCR Applications

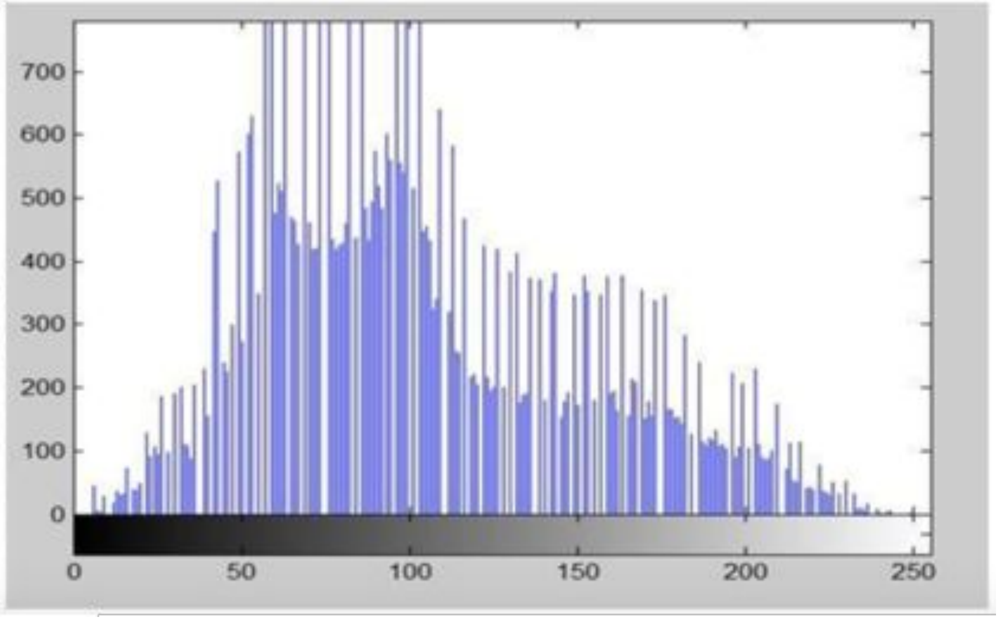
Application Fields

Image Processing Methods

Histogram of an image

- Histograms shows **frequency**. The frequency of pixels intensity values.





Histogram Equalization

- Histogram equalization is used to enhance contrast.

  Steps:

  1- **Find histogram of an image**

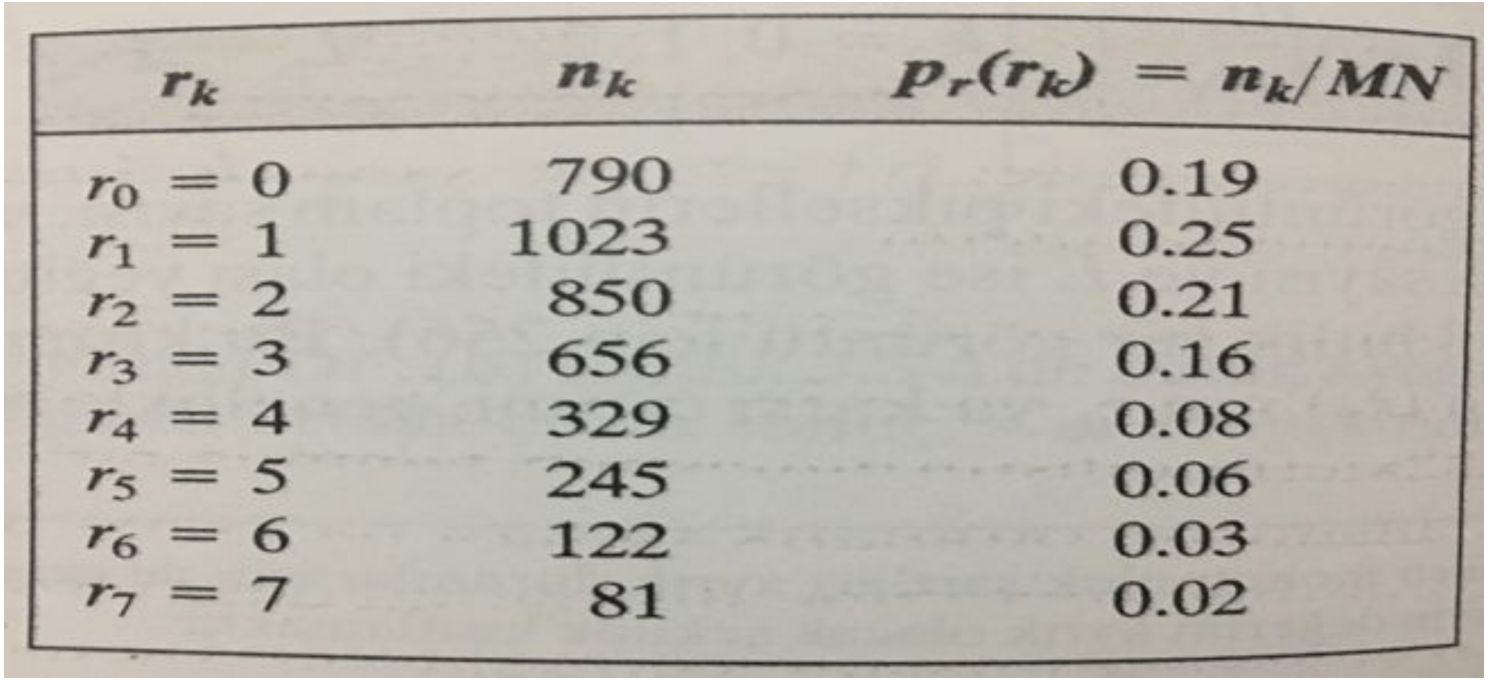
  2- **Calculate Cumulative Distributive Function**

  3- **Map new values in the histogram**

- Let’s assume that we have **3-bit** image. It will have **(0-7) values**.

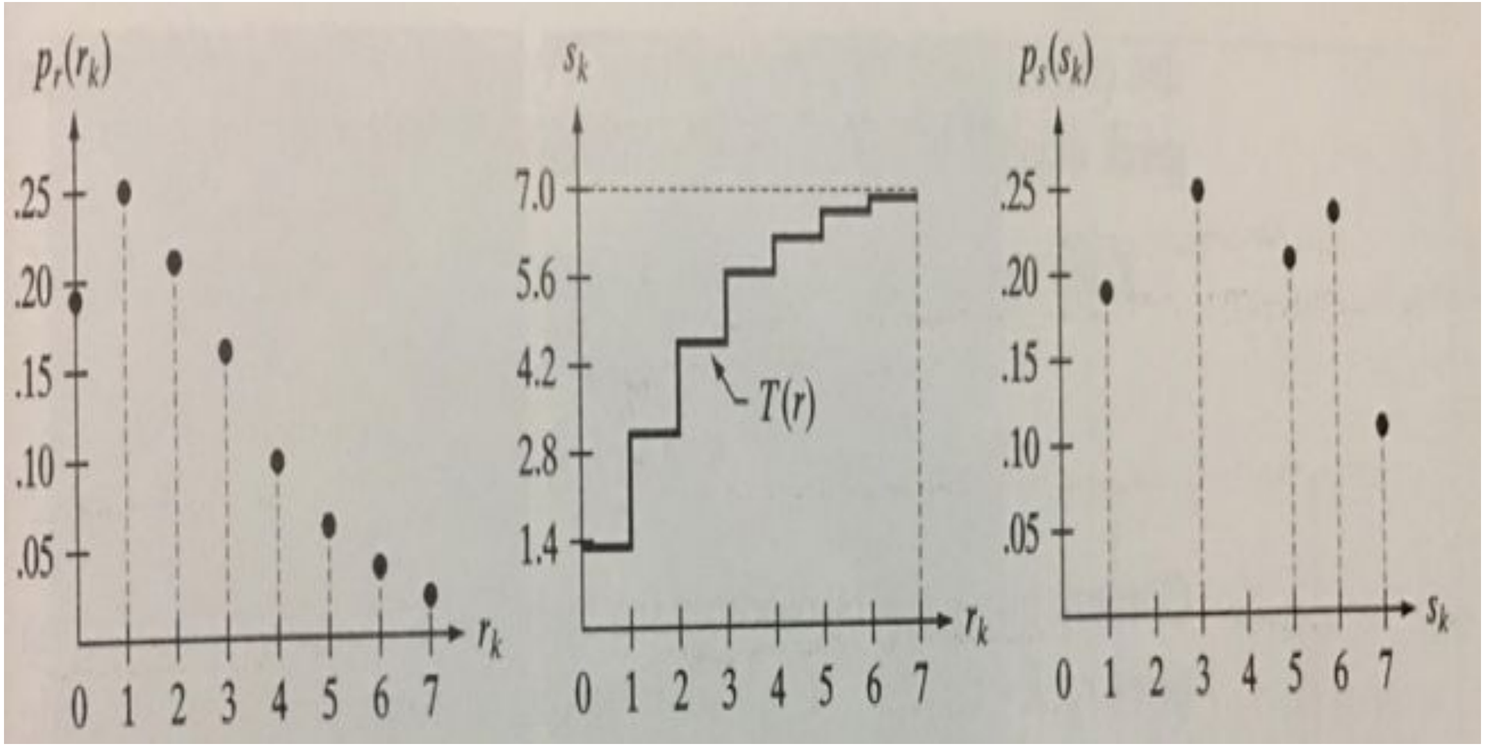
Histogram Equalization

- M=64 and N=64 ==> MxN=4096

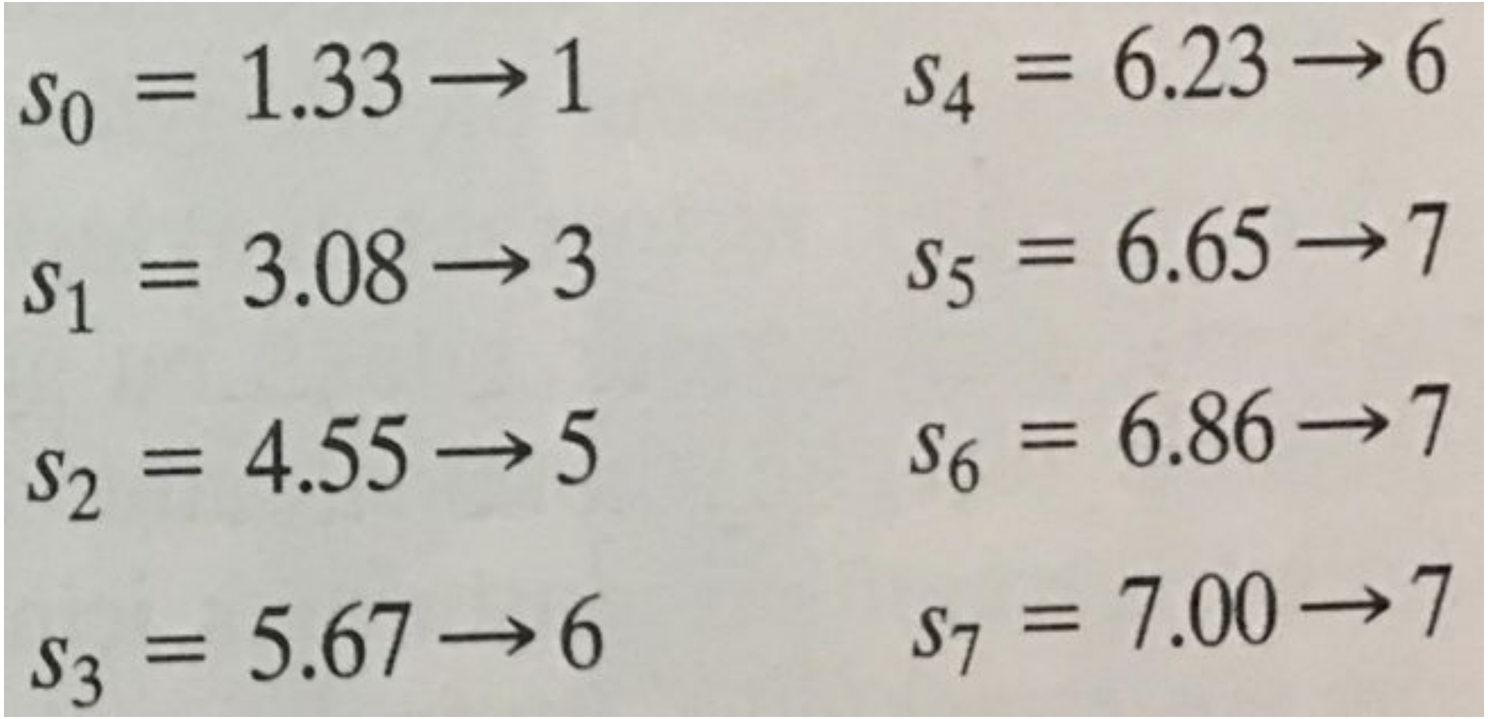


r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Histogram Equalization



Histogram Equalization



$$s_0 = 1.33 \rightarrow 1$$

$$s_1 = 3.08 \rightarrow 3$$

$$s_2 = 4.55 \rightarrow 5$$

$$s_3 = 5.67 \rightarrow 6$$

$$s_4 = 6.23 \rightarrow 6$$

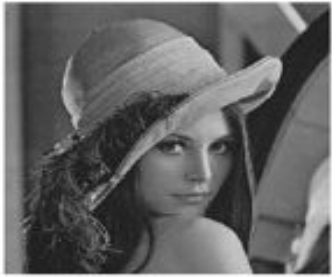
$$s_5 = 6.65 \rightarrow 7$$

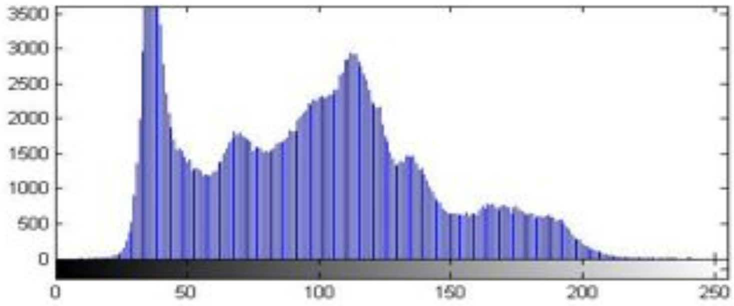
$$s_6 = 6.86 \rightarrow 7$$

$$s_7 = 7.00 \rightarrow 7$$

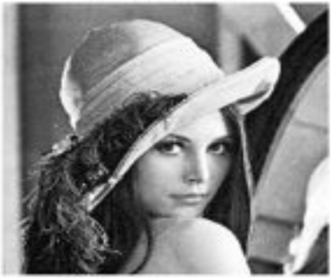
Histogram Equalization

Original Image

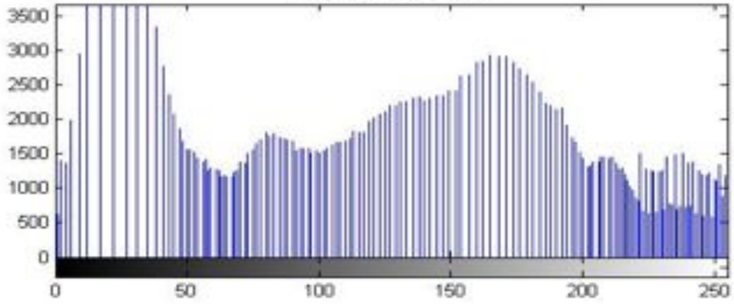




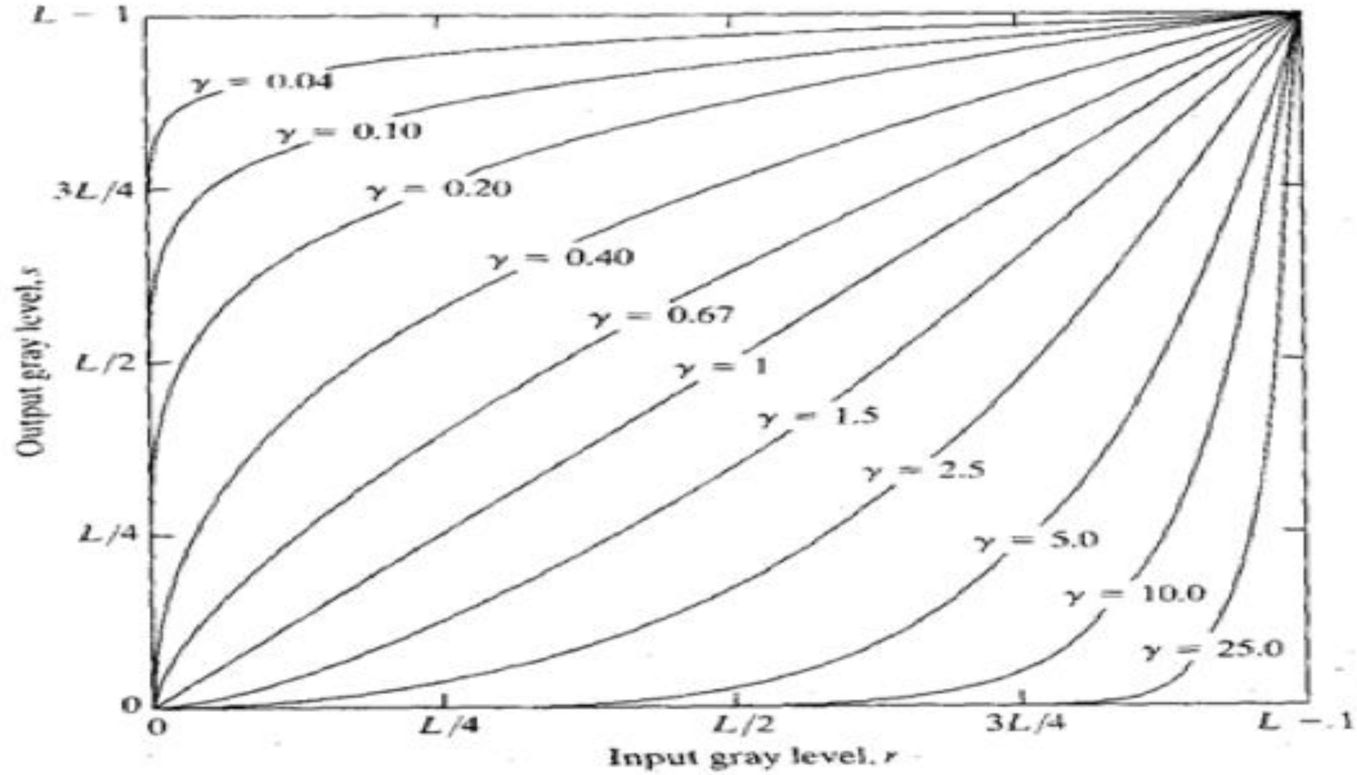
Enhanced Image



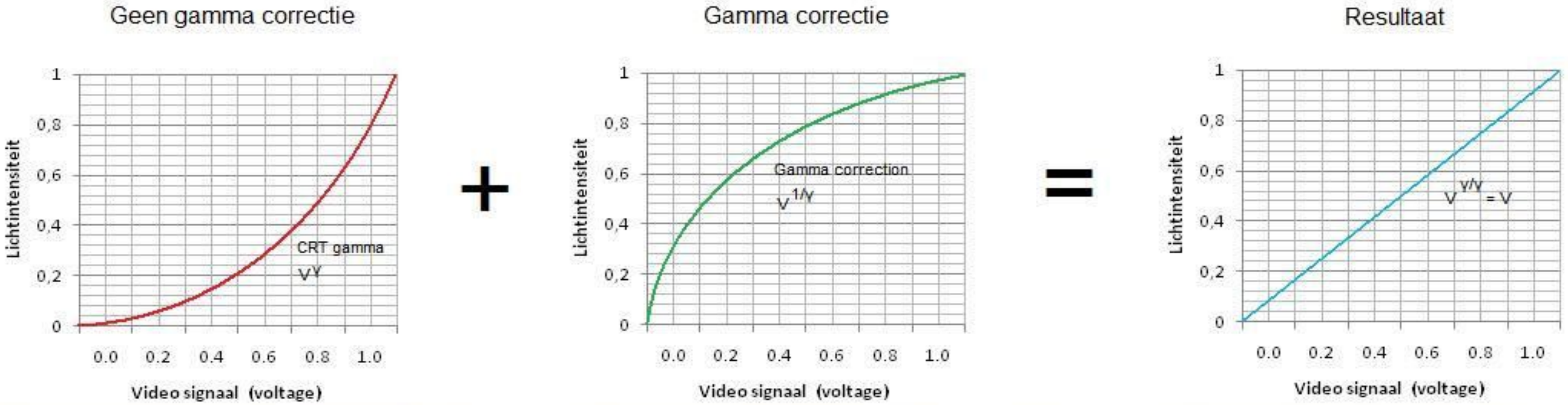
Histogram equalization

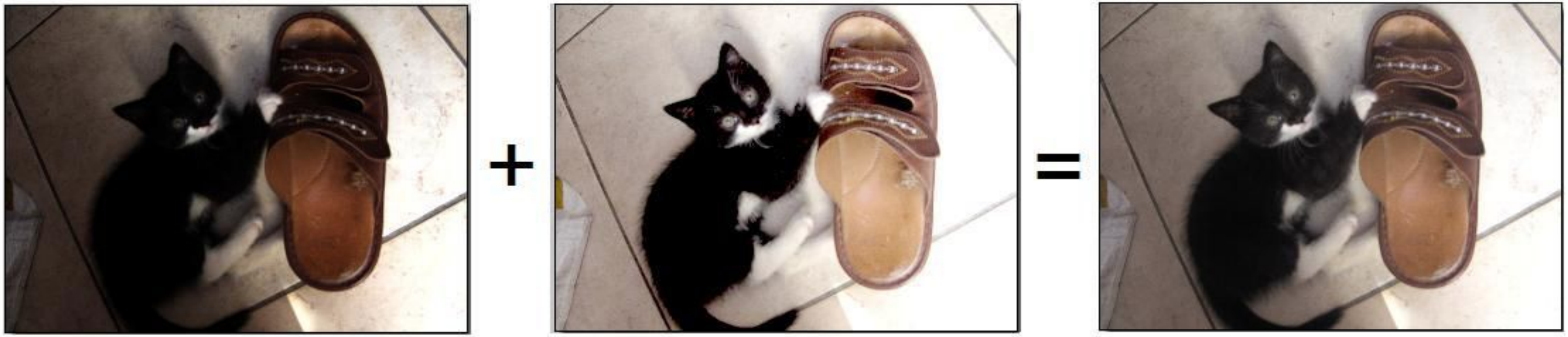


Gamma Correction

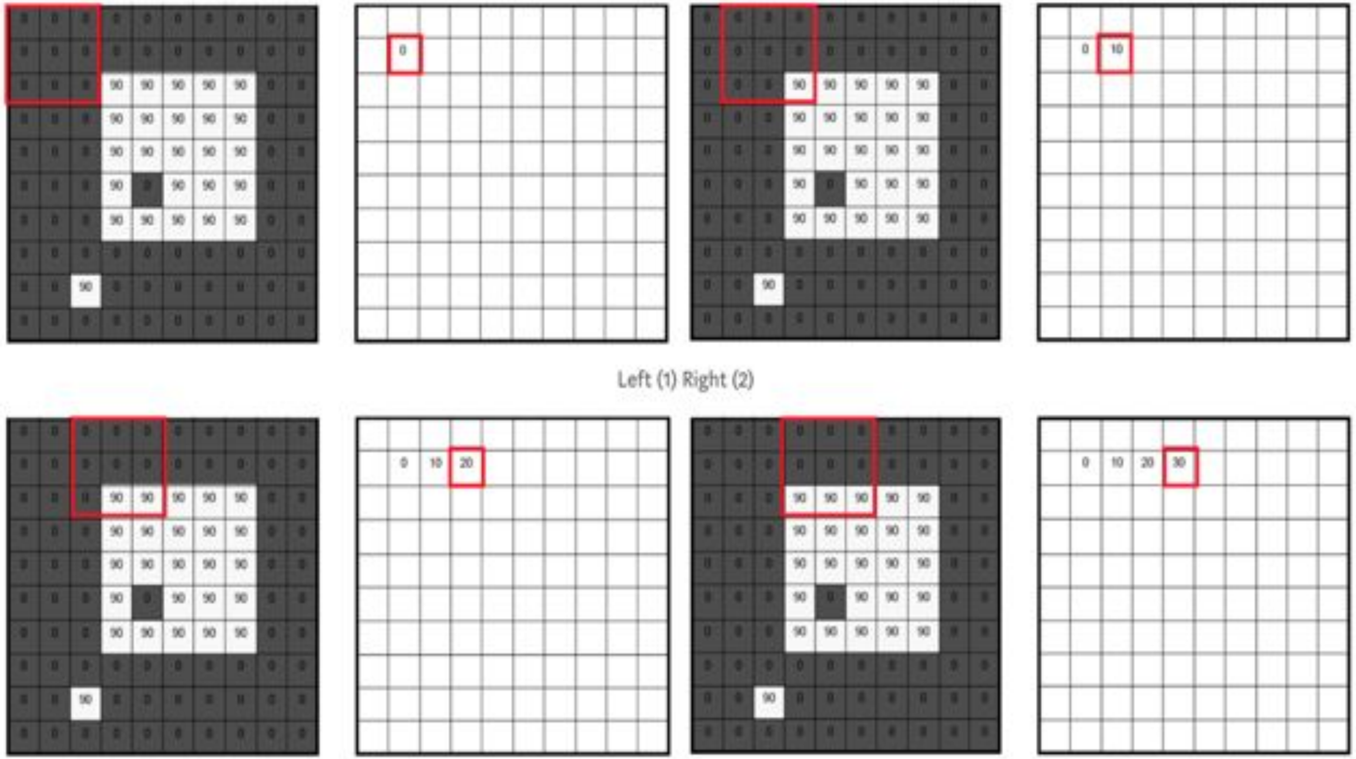


Gamma Correction



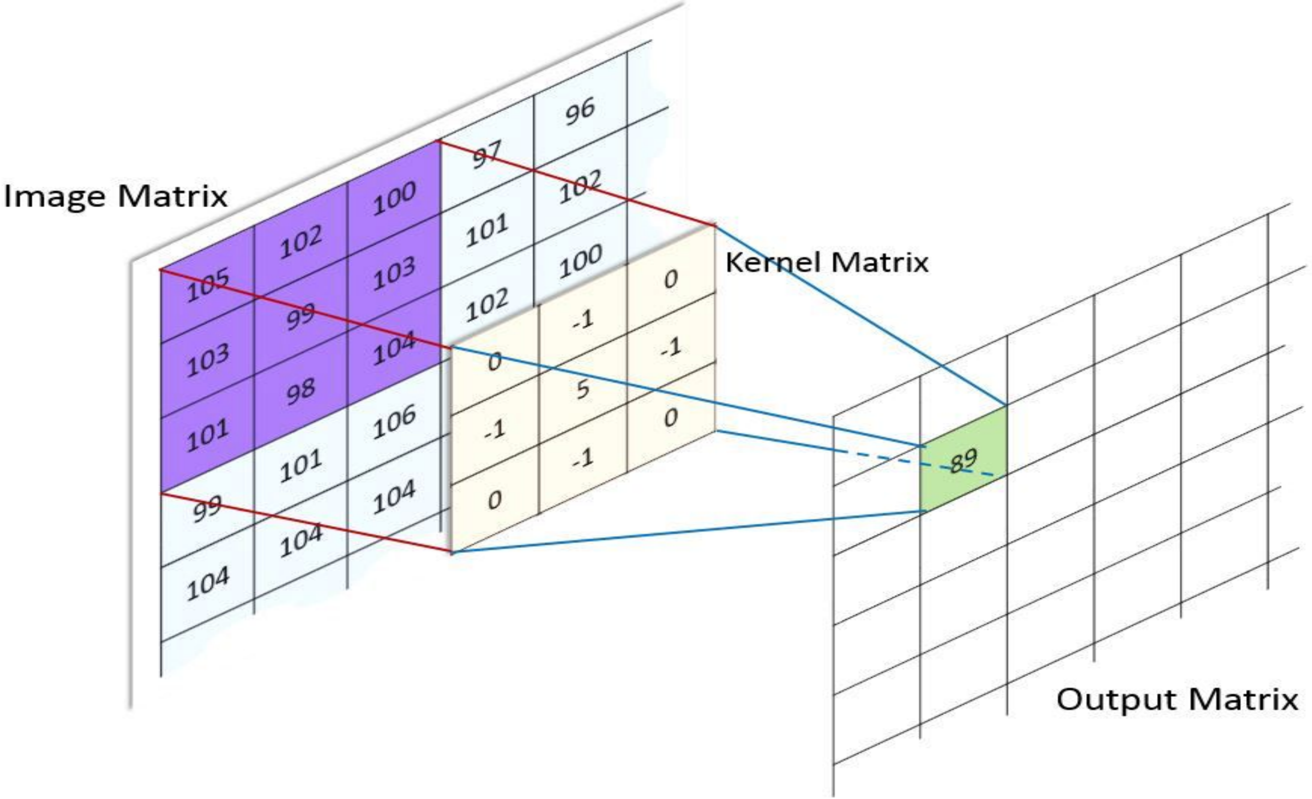


Convolution



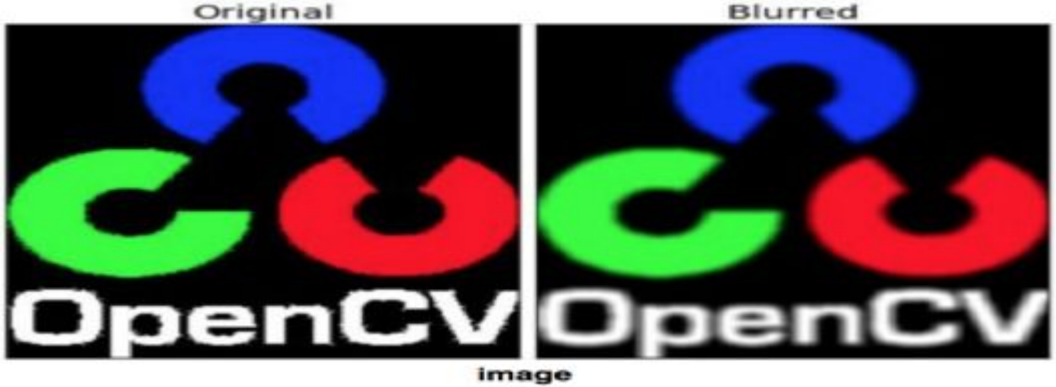
Left (1) Right (2)

Convolution



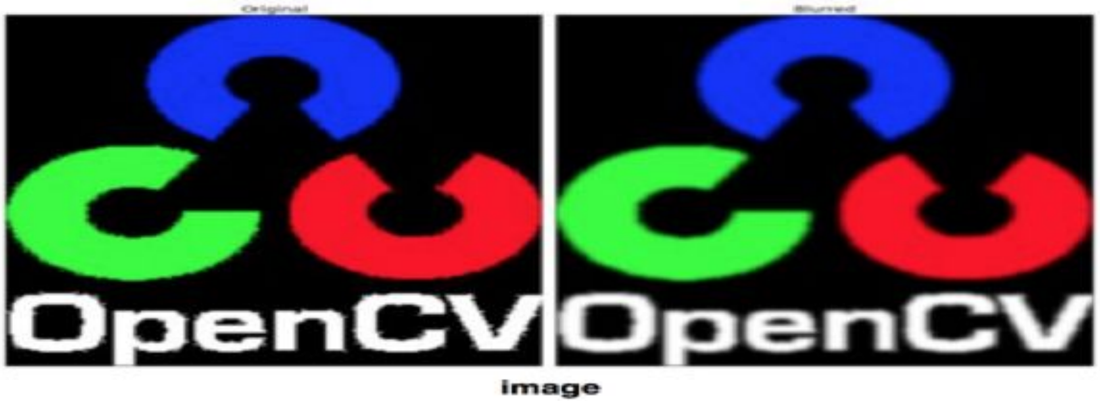
Smoothing Images

- **Averaging** : It simply takes the “**average of all the pixels**” under kernel area and replace the central element.



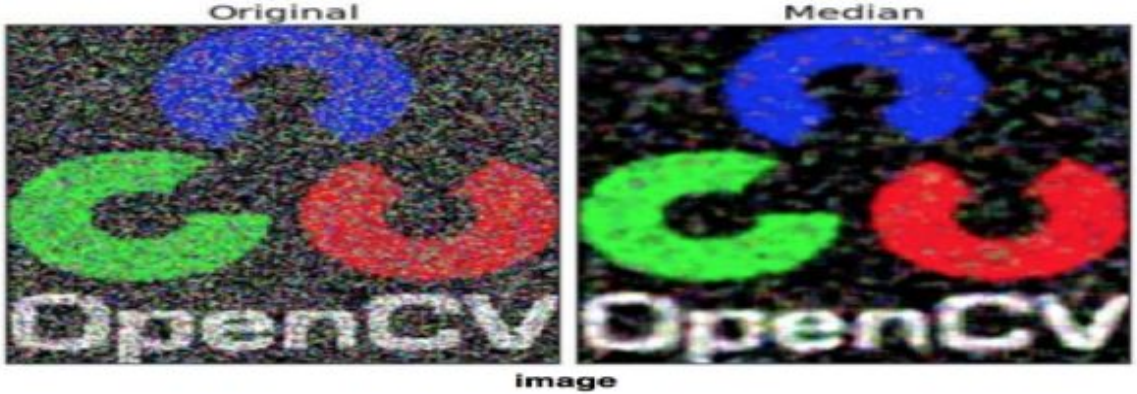
Smoothing Images

- **Gaussian Blurring** : It blurs an image by a Gaussian function to “**reduce image noise and reduce detail**”.



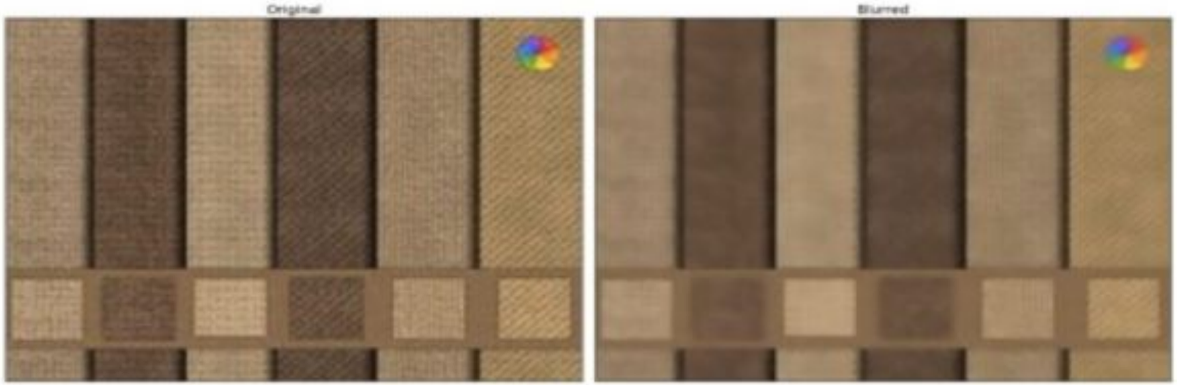
Smoothing Images

- **Median Blurring** : It takes “**median of all the pixels**” under kernel area and central element is replaced with this median value. This is highly effective against “**salt-and-pepper noise**” in the images.



Smoothing Images

- **Bilateral Filtering** : It is highly effective in noise removal while “**keeping edges sharp**”.



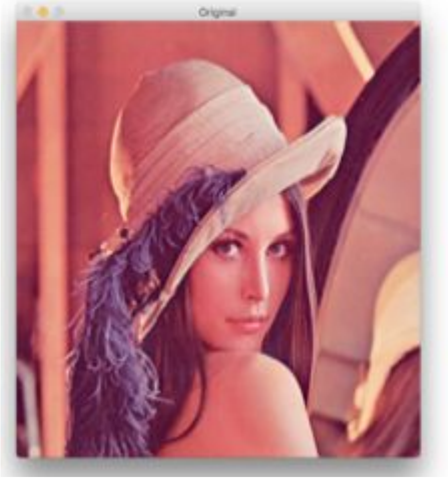
image

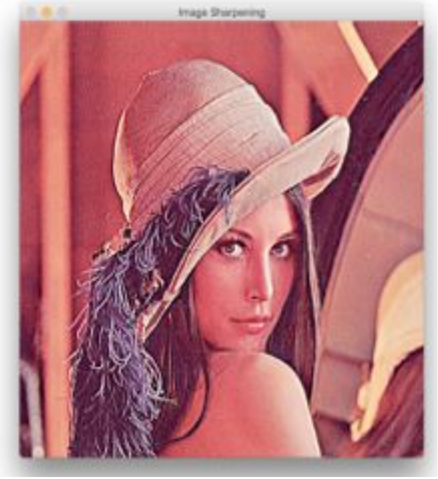
Sharpening Images

- We use OpenCV's filter2D function.

[-1,-1,-1]
[-1, 9,-1]
[-1,-1,-1]

Sharpening Kernel

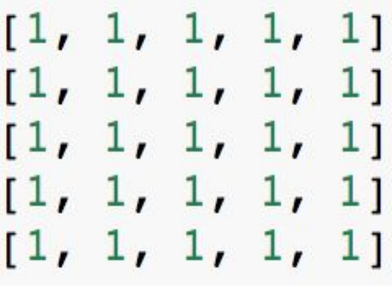




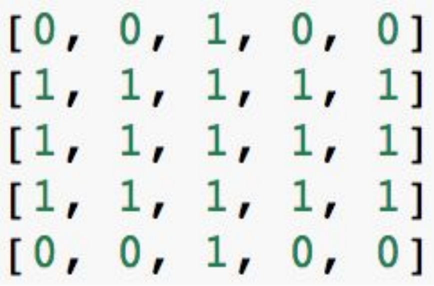
Morphological Transformations

- Morphological transformations are some simple operations based on the image shape.

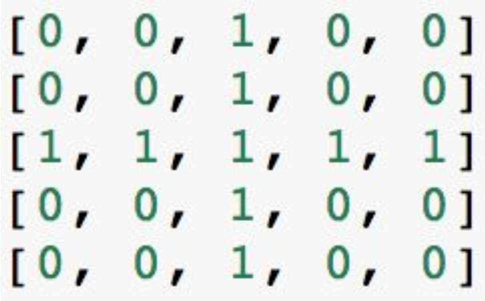
- Structuring Element:
  1. Rectangular Kernel
  2. Elliptical Kernel
  3. Circular Kernel



```
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
```



```
[0, 0, 1, 0, 0]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[1, 1, 1, 1, 1]
[0, 0, 1, 0, 0]
```



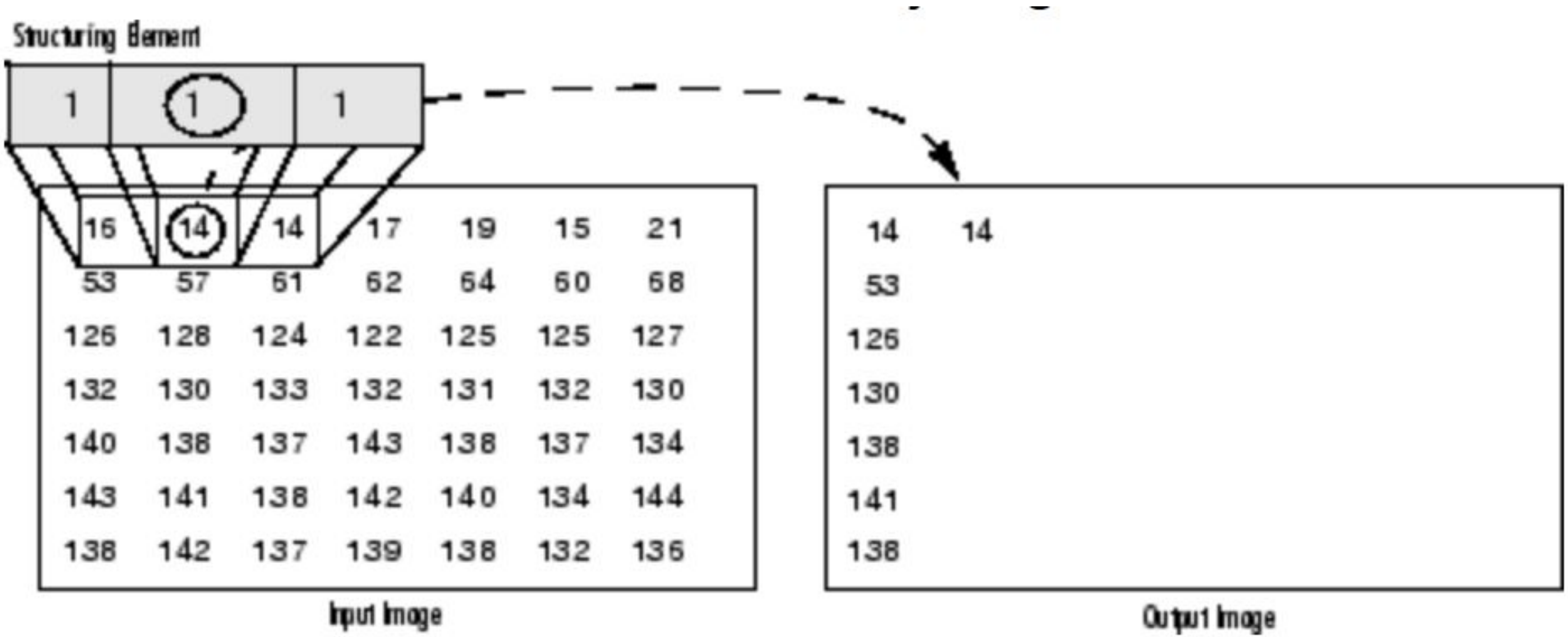
```
[0, 0, 1, 0, 0]
[0, 0, 1, 0, 0]
[1, 1, 1, 1, 1]
[0, 0, 1, 0, 0]
[0, 0, 1, 0, 0]
```

Morphological Transformations

- **Erosion** : It erodes away the boundaries of foreground object.

image

Morphological Transformations



16	14	14	17	19	15	21
53	57	61	62	64	60	68
126	128	124	122	125	125	127
132	130	133	132	131	132	130
140	138	137	143	138	137	134
143	141	138	142	140	134	144
138	142	137	139	138	132	136

14	14
53	
126	
130	
138	
141	
138	

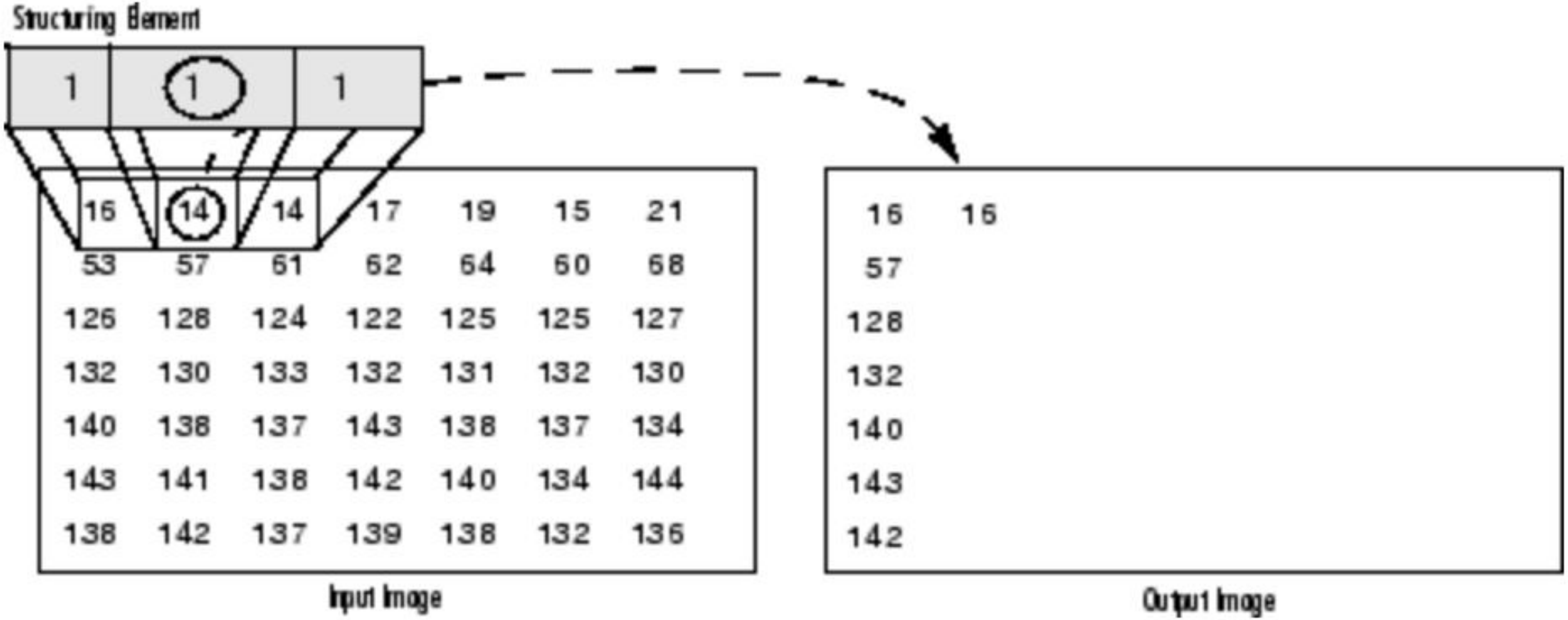
Erosion on a Grayscale Image

Morphological Transformations

- **Dilation** : It increases the white region in the image or size of foreground object increases.

image

Morphological Transformations



Structuring Element

1	1	1

Input Image

16	14	14	17	19	15	21
53	57	61	62	64	60	68
126	128	124	122	125	125	127
132	130	133	132	131	132	130
140	138	137	143	138	137	134
143	141	138	142	140	134	144
138	142	137	139	138	132	136

Output Image

16	16
57	
128	
132	
140	
143	
142	

Dilation on a Grayscale Image

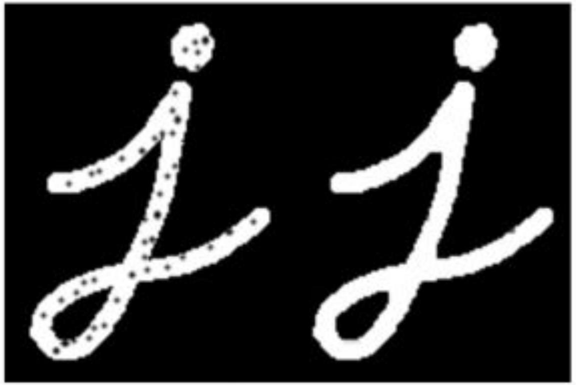
Morphological Transformations

- **Opening** : Erosion followed by Dilation

image

Morphological Transformations

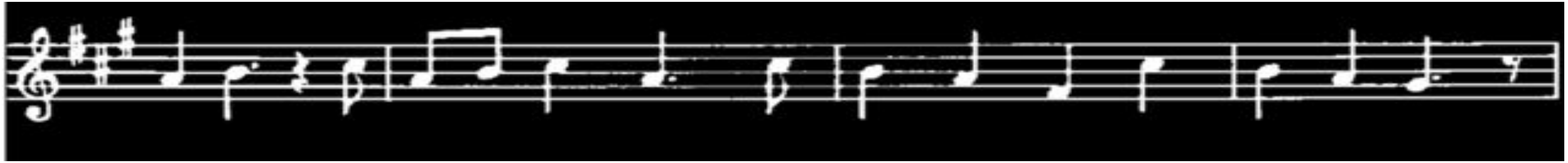
- **Closing** : Dilation followed by Erosion



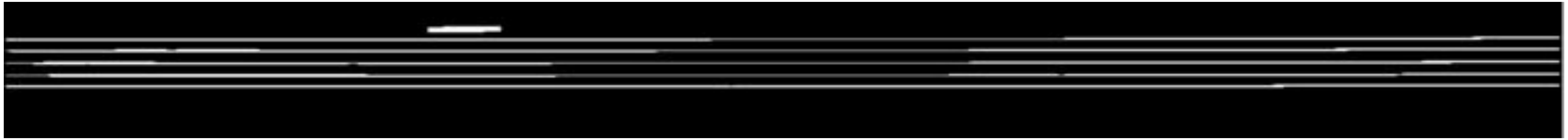
image

Morphological Transformations

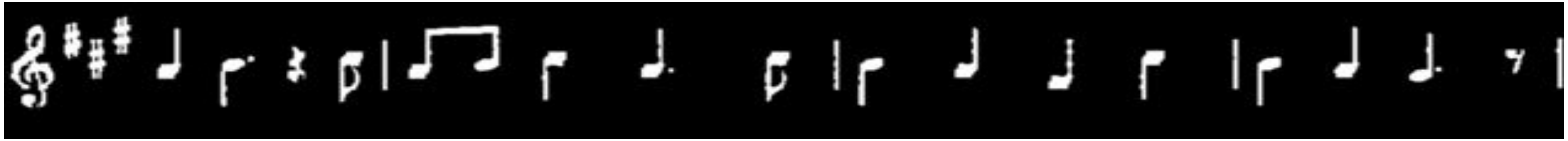
- Original



- Horizontal Opening



- Vertical Opening



Perspective Transformations

- Perspective transformation deals with the conversion of 3D world into 2D image.
- We **need 4 points** on the input image and corresponding points on the output image.

Perspective Transformations

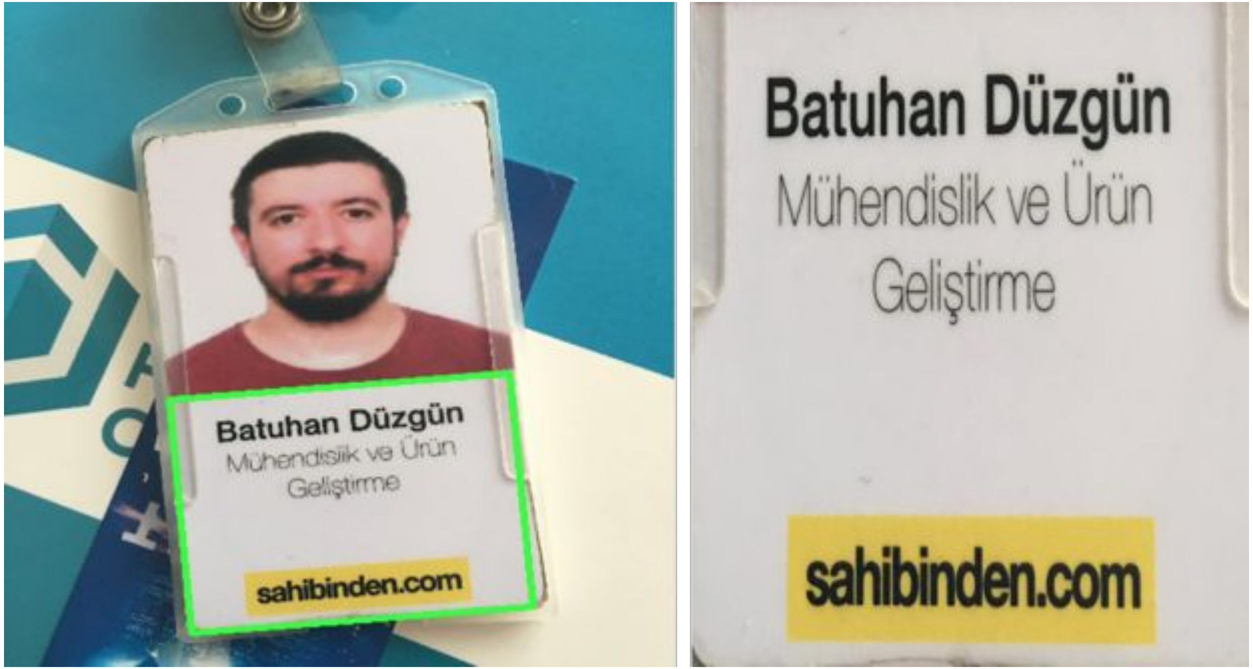


Image Thresholding

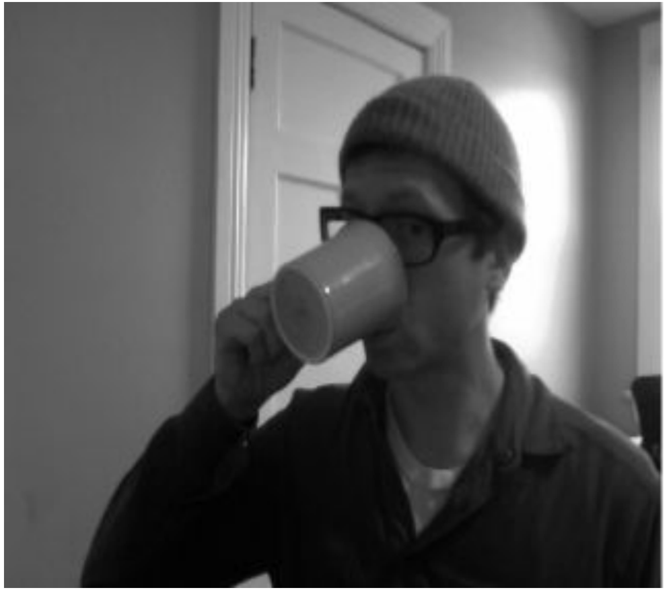
- From a grayscale image, thresholding can be used to **create binary images**.

Simple Thresholding : give a “**constant threshold**” value

Adaptive Thresholding : calculate “**threshold for a small regions**” of the image

Otsu’s Binarization : calculates “**threshold value from image histogram**” for a bimodal image

Image Thresholding



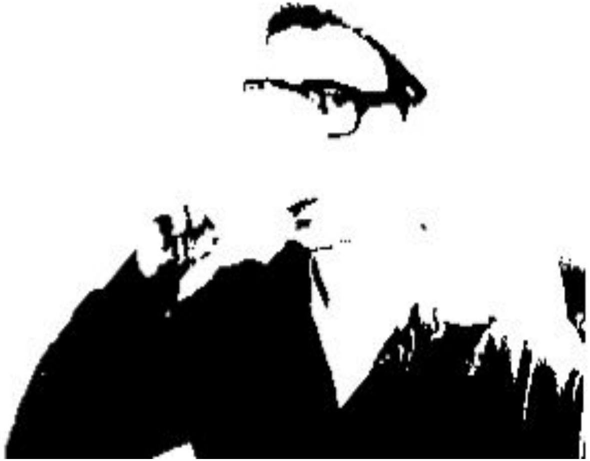
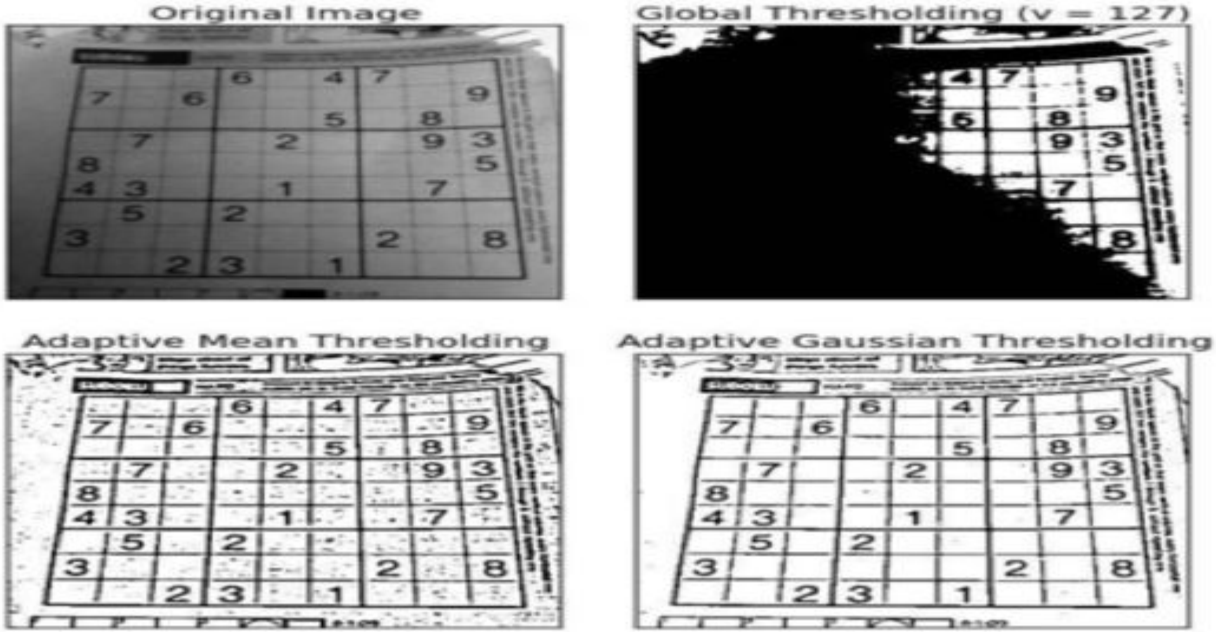
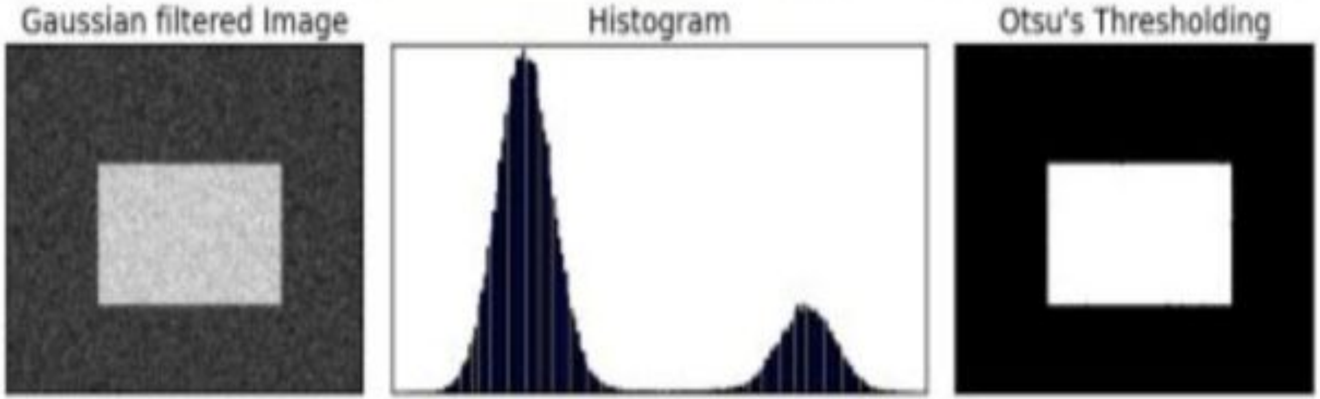


Image Thresholding


Original Image
Global Thresholding (v = 127)
Adaptive Mean Thresholding
Adaptive Gaussian Thresholding

image

Image Thresholding



image

Edge Detection

What are edges?

We can also say that **sudden changes of discontinuities** in an image are called as edges.

- Canny Edge Detection
- Sobel Edge Detection
- Laplacian Edge Detection

Edge Detection

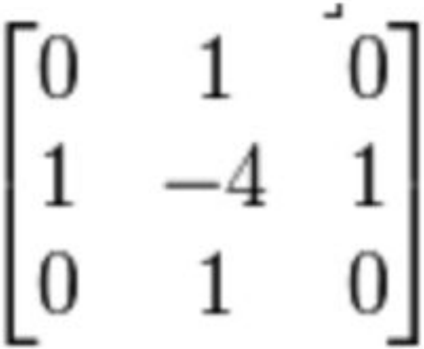
- **Sobel Edge Detection** : First derivatives of the image separately for the X and Y axes.

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

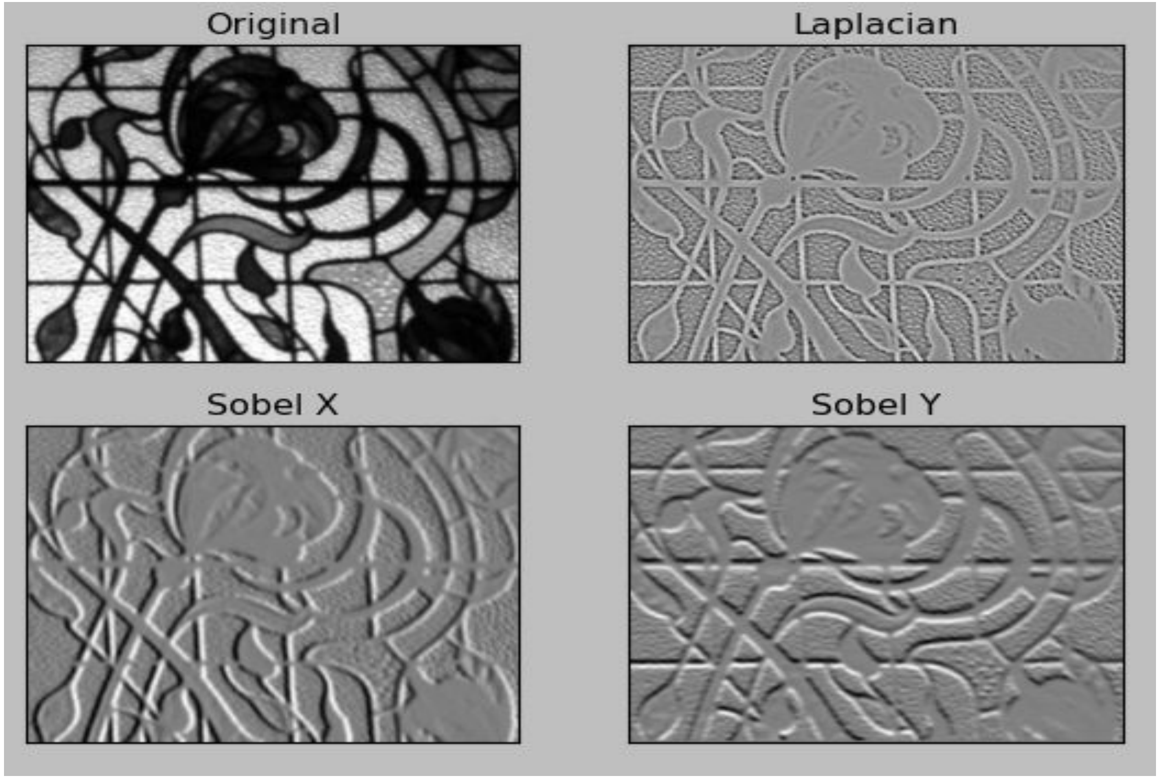
Edge Detection

- **Laplacian Edge Detection** : Laplacian edge detector uses only one kernel.



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Edge Detection

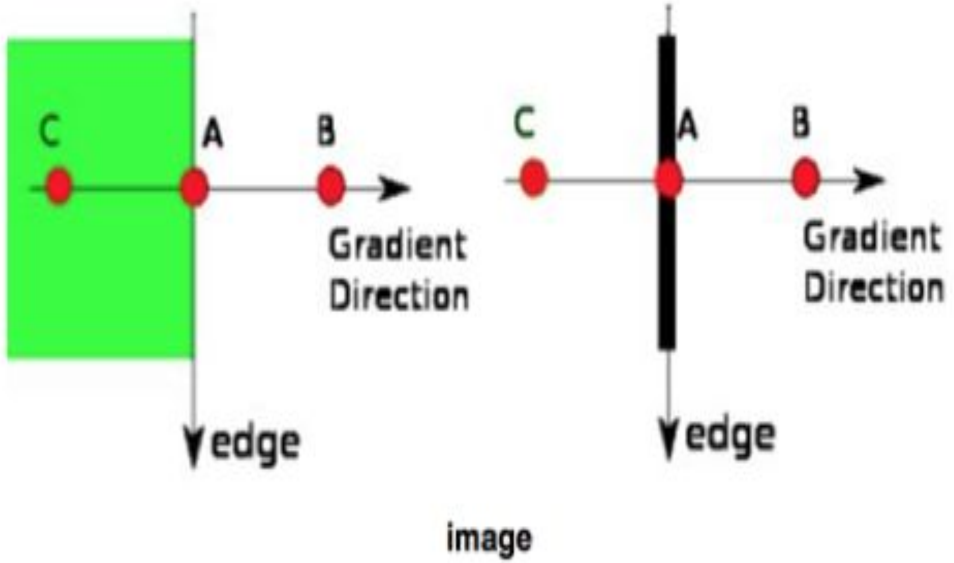


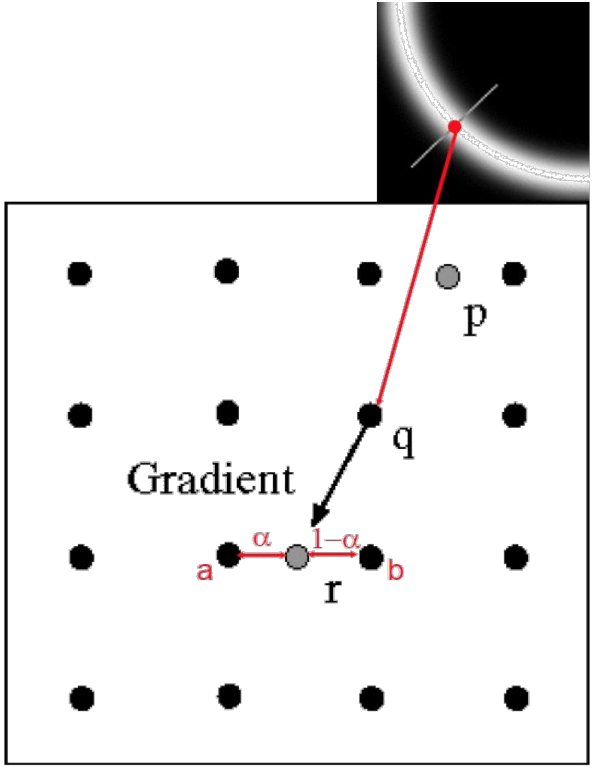
Edge Detection

- Canny Edge Detection

1. **Noise Reduction:** 5x5 Gaussian Filter for smoothing.
2. **Finding Intensity Gradient of the Image:** Using Sobel kernels.
3. **Non-maximum Suppression:** remove any unwanted pixels.

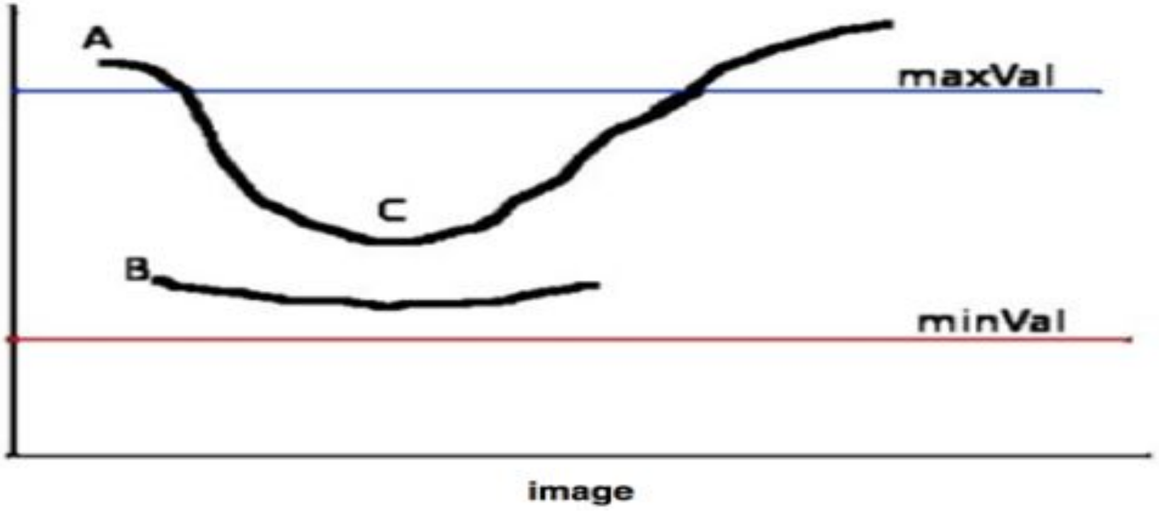
Edge Detection





Edge Detection

- **Hysteresis Thresholding** : two threshold values, **minVal** and **maxVal**.



Tesseract OCR Engine & OpenCV

Optical Character Recognition (OCR) ?

- It includes the mechanical and electrical conversion of scanned images of handwritten, typewritten text into machine text.
- Turning image files into fully searchable documents.

Tesseract OCR Engine?

- Tesseract is an **open source** text recognizer (OCR) Engine, available under the Apache 2.0 license.

Install on Mac OS : brew install tesseract

Source Code : https://github.com/tesseract-ocr/

Tesseract OCR

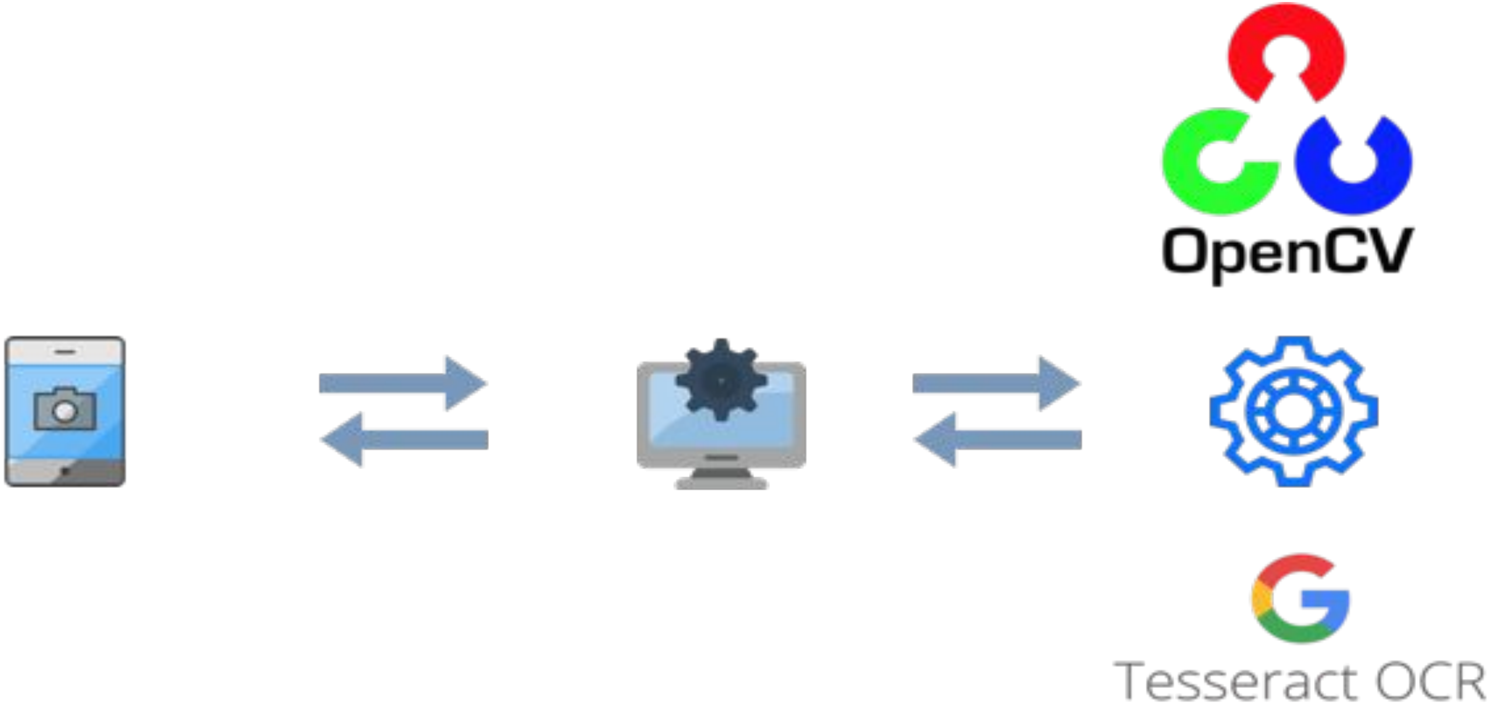
OpenCV ?

- OpenCV is a library of programming functions mainly aimed at real-time computer vision.

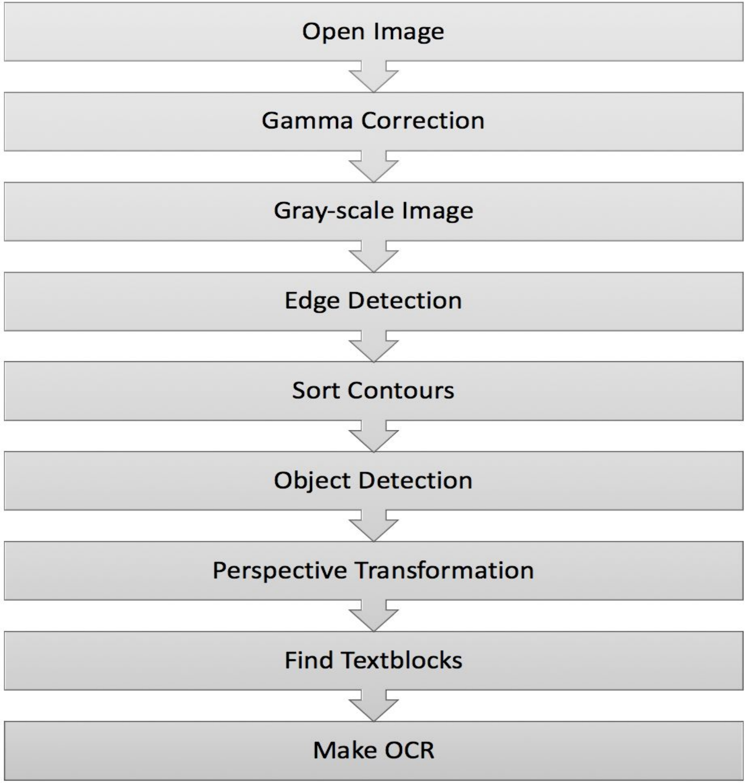
Supported Languages : C++, Python and Java

Example Project: “sahibinden.com” Staff Identity Card OCR Project

Project Architecture



Project Architecture



Open Image

Gamma Correction

Gray-scale Image

Edge Detection

Sort Contours

Object Detection

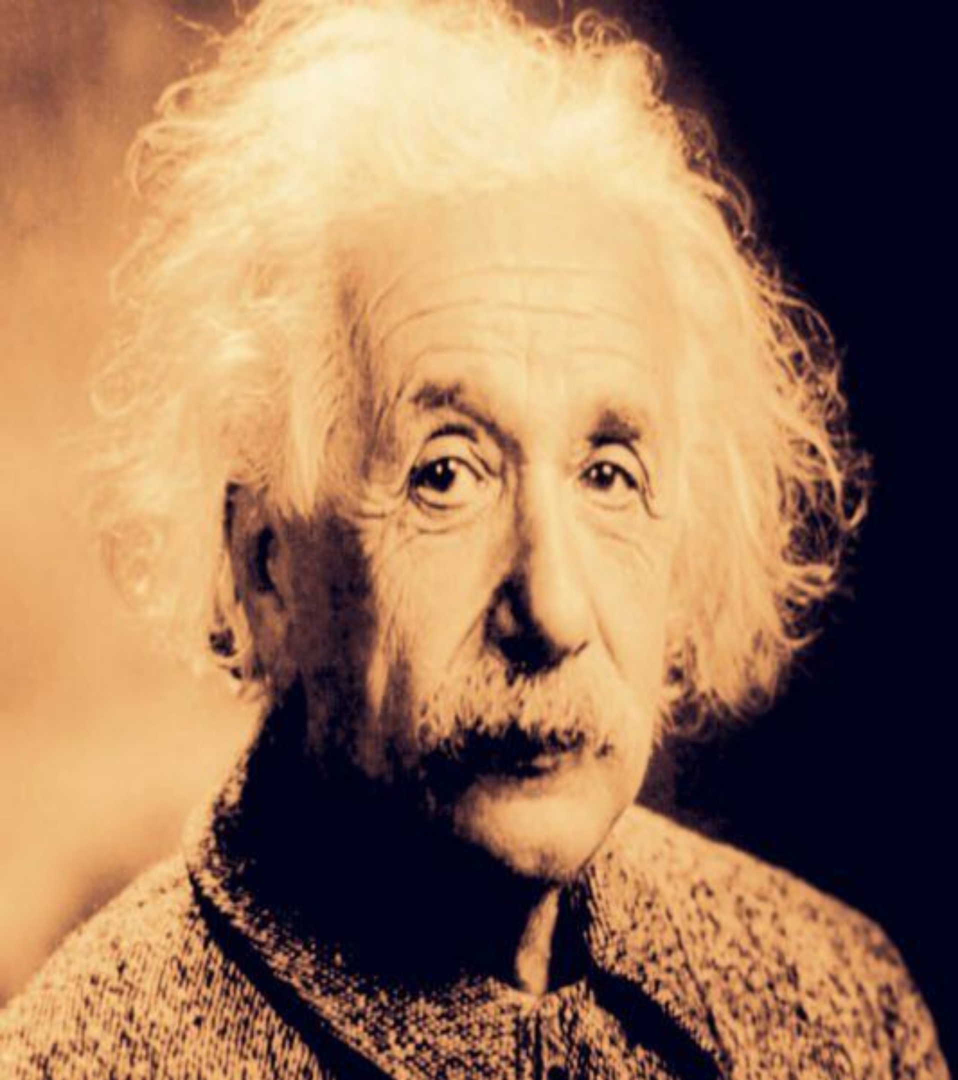
Perspective Transformation

Find Textblocks

Make OCR

The Project Source Code :)

https://github.com/batux/staff_identity_card_ocr_project



“We now accept the fact that learning is a lifelong process of keeping abreast of change.”

Albert Einstein

References

OpenCV ?

https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Gradient_Sobel_Laplacian_Derivatives_Edge_Detection.php

https://docs.opencv.org/3.1.0/da/d22/tutorial_py_canny.html

https://www.tutorialspoint.com/dip/index.htm

https://medium.com/@almutawakel.ali/opencv-filters-arithmetic-operations-2f4ff236d6aa

https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html

https://docs.opencv.org/3.4.2/dd/dd7/tutorial_morph_lines_detection.html

https://docs.opencv.org/3.4.2/dd/dd7/tutorial_morph_lines_detection.html

https://docs.opencv.org/3.1.0/da/d22/tutorial_py_canny.html

https://www.pyimagesearch.com/2017/07/17/credit-card-ocr-with-opencv-and-python

Thanks!

in /in/batuhanduzgun

@batux

batuhan.duzgun@windowslive.com

www.batuhanduzgun.com