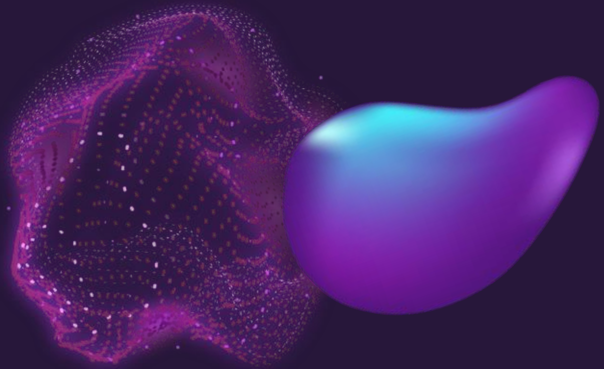


Image Processing & Computer Vision

By Keven Sakr & Charbel Hleyhel

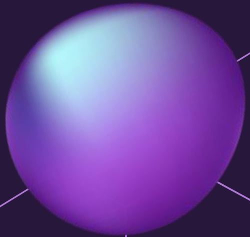


TABLE OF CONTENTS

01 Introduction

02 Image Formation

03 Image Transformation

04 Image Filtering

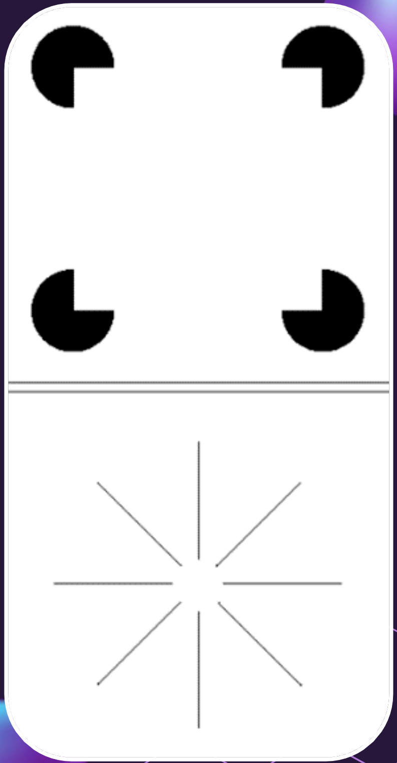
05 Code & Demo

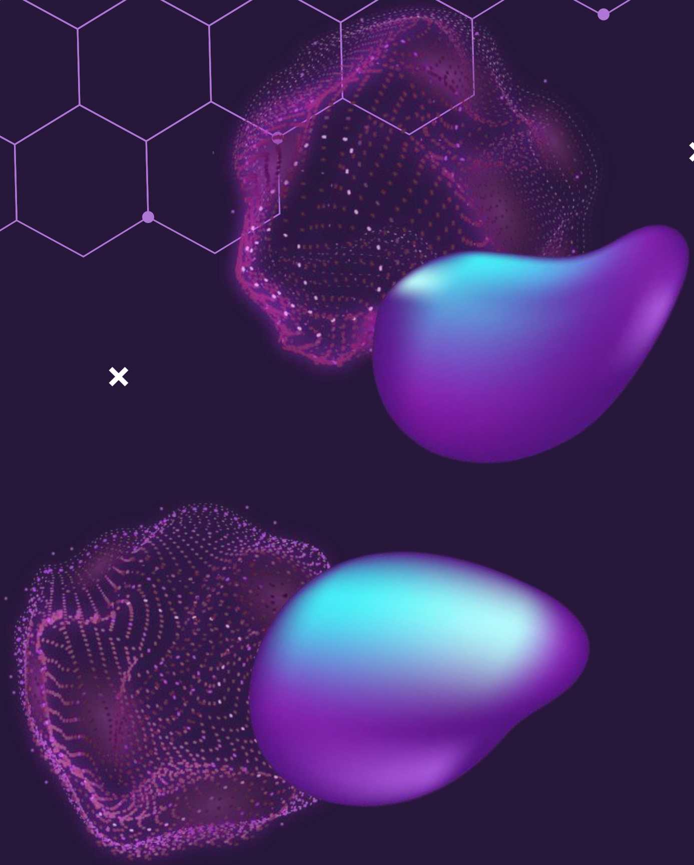
01

Introduction

Vision VS Computer Vision

The human vision system is made to "understand" the scene, not just measure the light.





Computers can “See” Using cameras, sensors, etc...

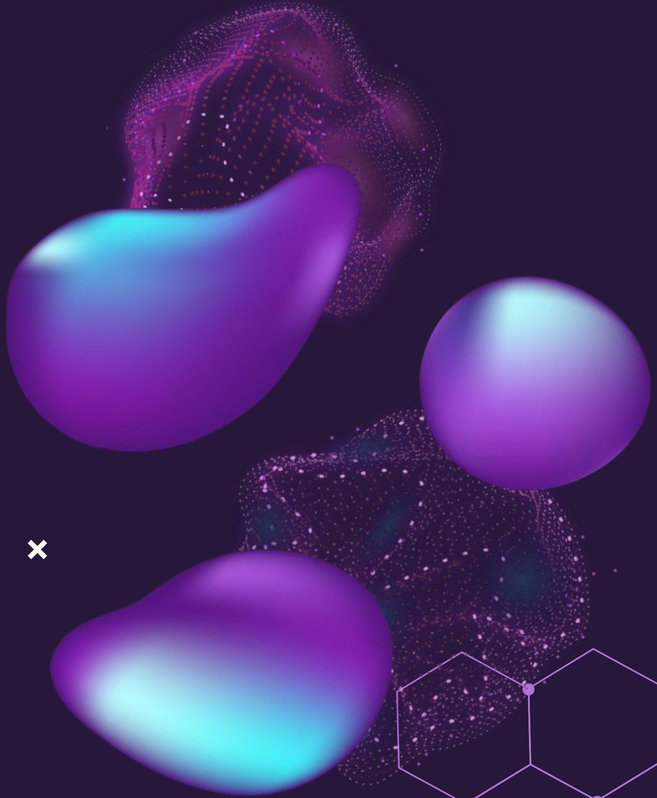
But can they understand?

Digital Image Processing

Attempts to make computers understand what they see

→ Computer Vision

- Where are the cars ?
- Where are the humans ?
- How far is that tree ?
- What is the plate number of this car ?



Using What?

Image formation

Transformation Algorithm

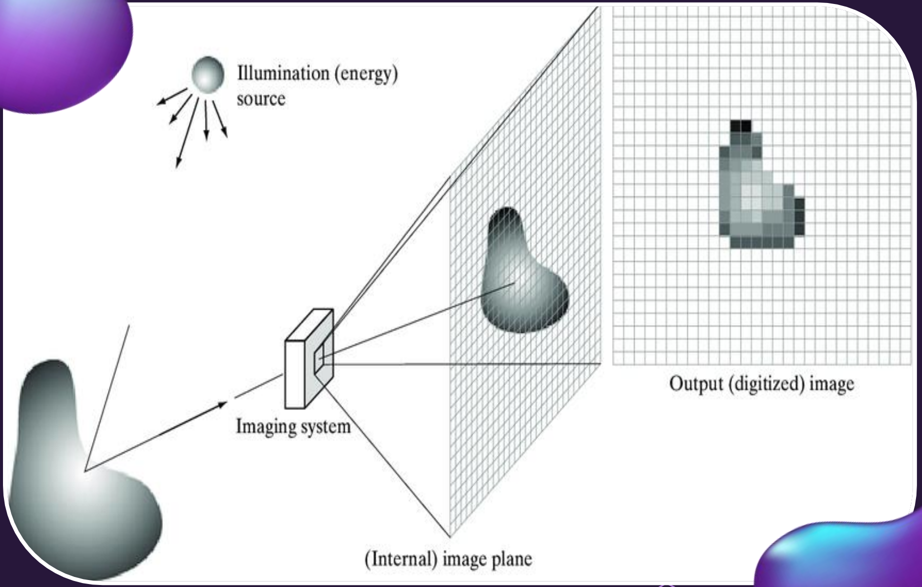
Filtering Algorithm

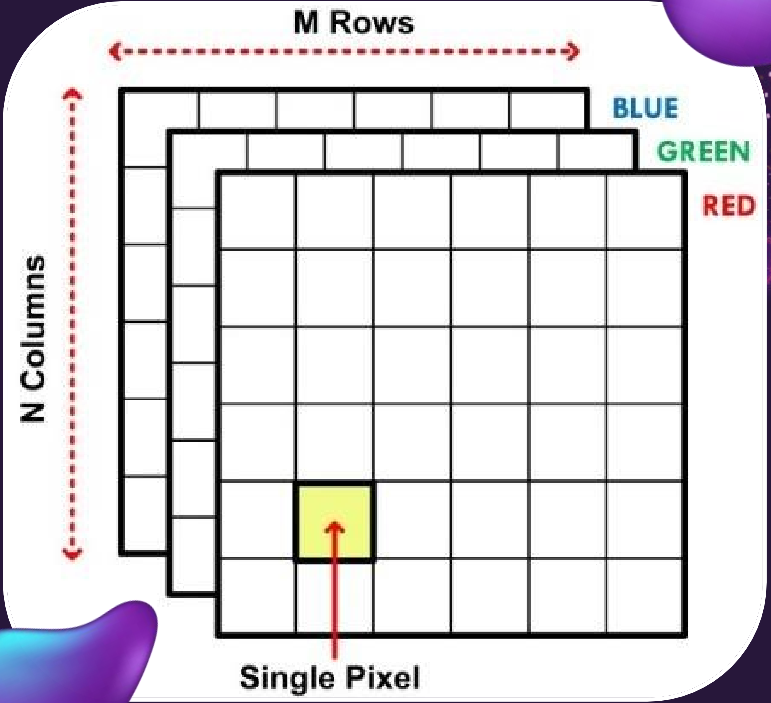
And others...

02

Image Formation

Process of capturing and representing visual information in the form of digital images





Digital Image

- 2D Function

> Position → Intensity (0-255)

> (x,y) → f(x,y)

- Matrix for each color

> RGB → 3 Matrixes

> Grayscale → 1 Matrix

03

Image Transformation

Image Transformation

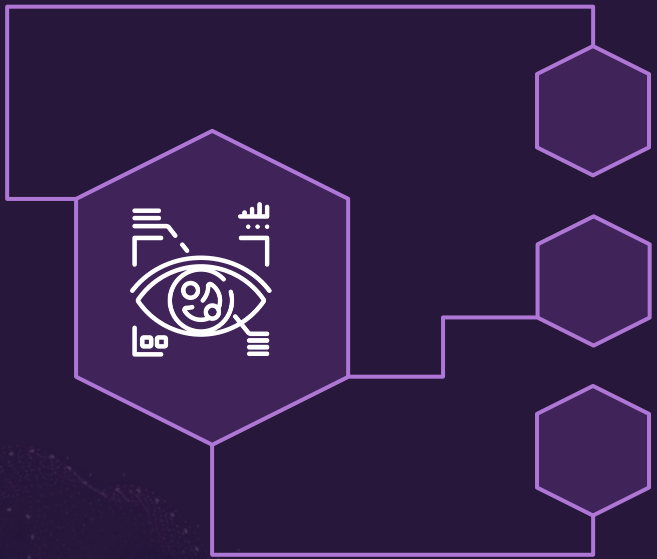


Image Negative

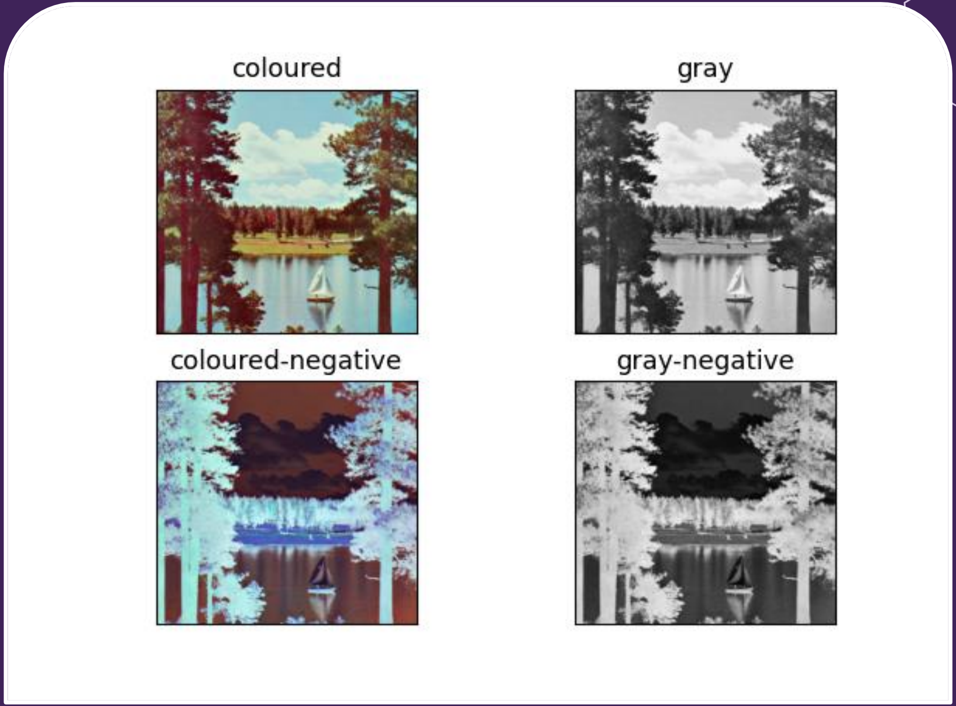
Reverse Colors for greyscale images

Thresholding

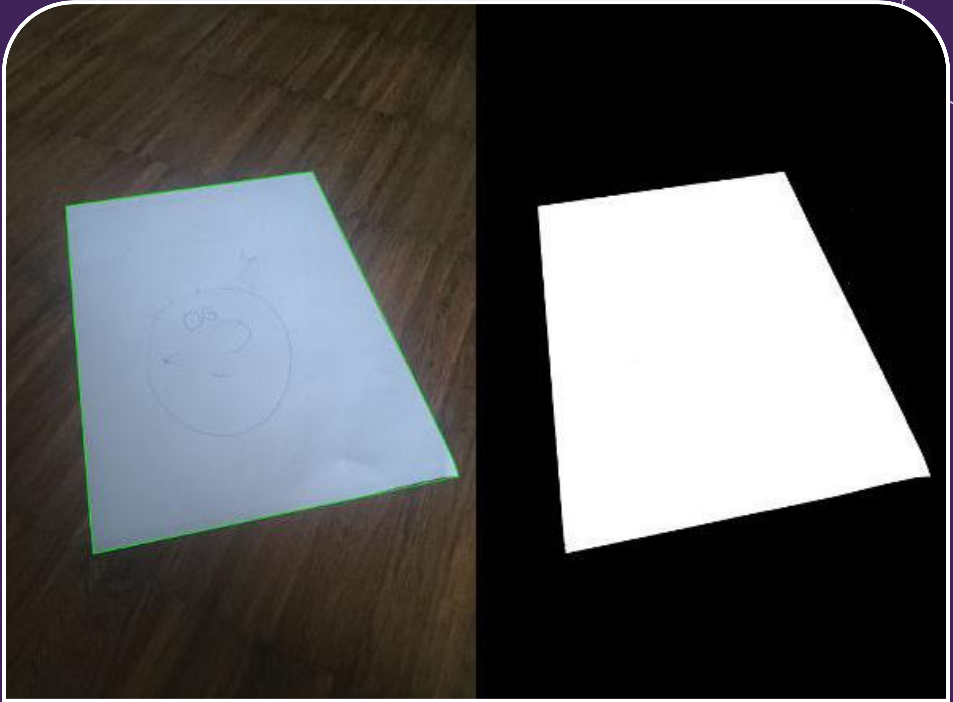
Binarization: Black and White

Histogram Equalization

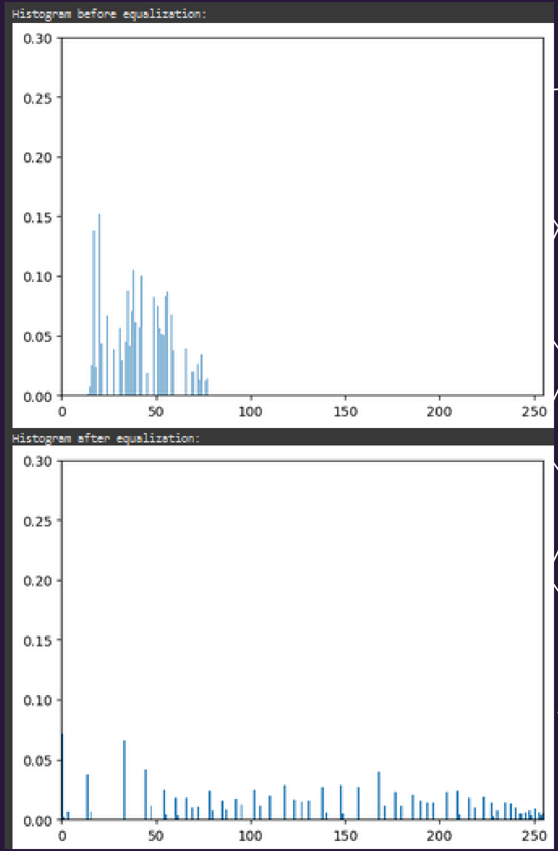
Used for contrast enhancement

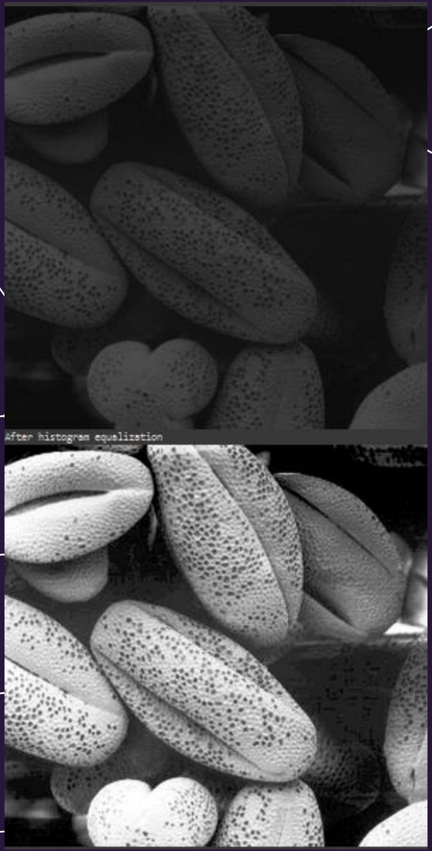


Negative Image



Thresholding





Histogram Equalization

04

Image Filtering

Convolution

An image kernel is a small matrix used to apply effects

→ as blurring, sharpening, outlining....

Also used in machine learning for 'Feature Extraction'

→ a technique for determining the most

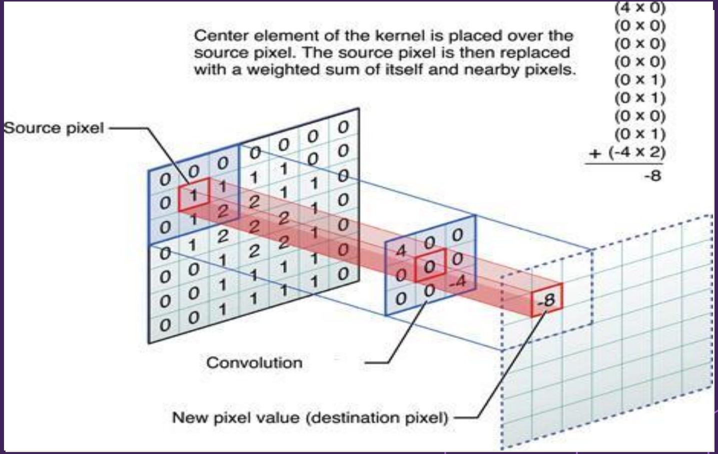

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
Source pixel
Convolution
New pixel value (destination pixel)
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8

Image kernel

blur

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

De-emphasizes differences

sharpen

0	-1	0
-1	5	-1
0	-1	0

emphasizes differences in adjacent pixel values

outline

-1	-1	-1
-1	8	-1
-1	-1	-1

Highlight large differences in pixel values

05

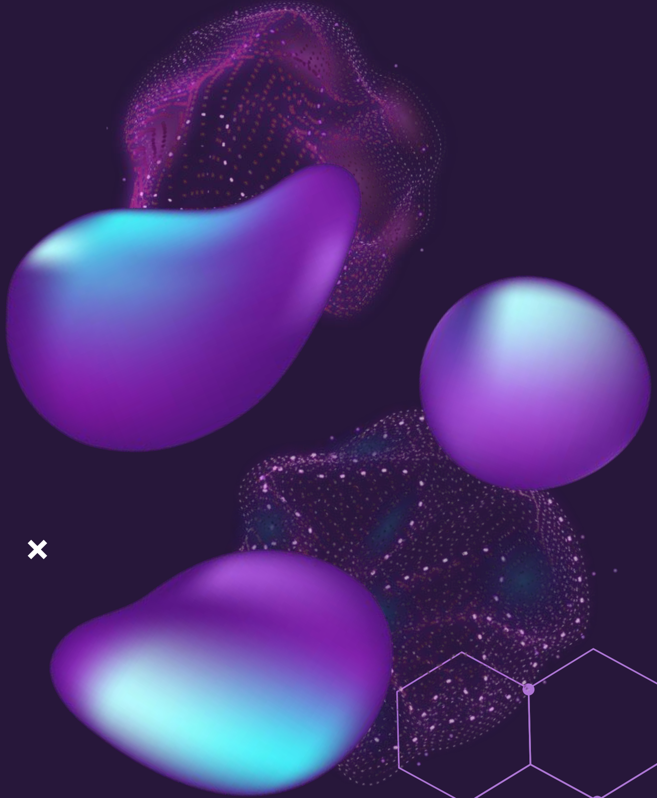
Code & Demo

Plate number recognition program

Text detection and recognition in images

Acording to Chen, Odobez & Bourlard (2004), the method is split into two main parts:

- the detection of text lines,
- followed by the recognition of text in these lines.



Our Libraries

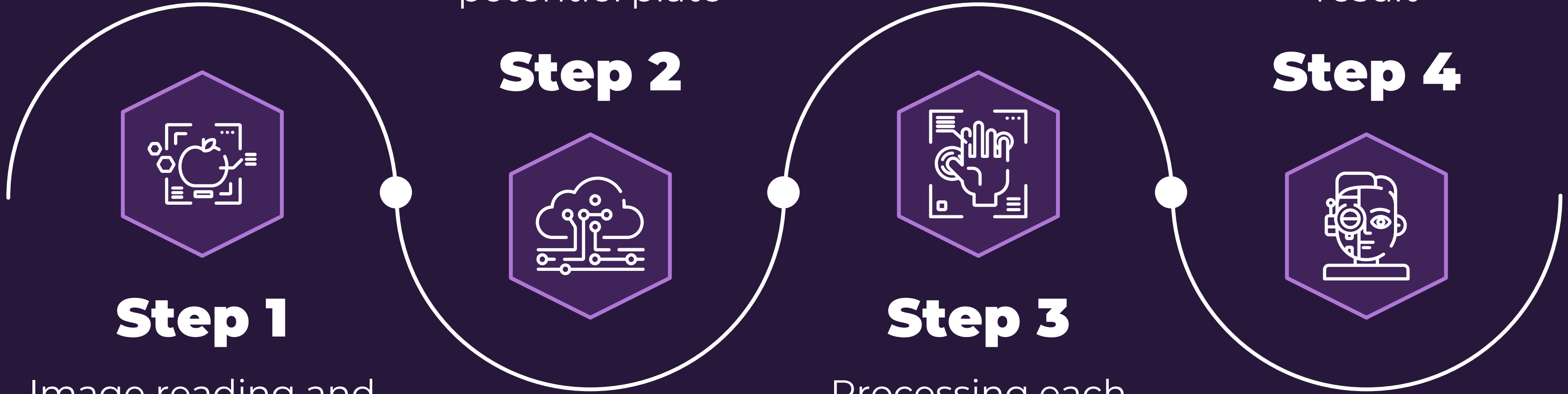
OpenCV

Real-time optimized Computer Vision library, tools, and hardware.

EasyOCR

Python package that allows to perform Optical Character Recognition.

Plate Number Recognition Road Map



Step 1

Image reading and processing (trans to gray, bluring)

Step 2

Contour and edge detection for potentiel plate

Step 3

Processing each region using OCR (easyOCR)

Step 4

Analysing obtained data and showing result

References

Basavarajaiah, M. (2019). 6 basic things to know about convolution. Retrieved from https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411

Chen, D., Odobez, J., & Bourlard, H. (2004). Text detection and recognition in images and video frames. Pattern Recognition, 37(3), 595-608. doi:10.1016/j.patcog.2003.06.001

Klette, R. (2014). Concise computer vision. Lodon: Springer. doi:10.1007/978-1-4471-6320-6

Powell, V. Image kernels explained visually. Retrieved from https://setosa.io/ev/image-kernels/

Rosebrock, A. (2020). Getting started with EasyOCR for optical character recognition. Retrieved from https://pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/