

ANKARA UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING



REPORT I

Optical Character Recognition

Mustafa Ahmet DENİZ	14290087
Mustafa GÖKSEVER	14290099

Introduction

Ocr anlat genel proje tanimi , OCR in python import pytesseract

OCR (optical character recognition) is the recognition of text by a computer. This involves photo scanning of the text character-by-character, analysis of the scanned-in image, and then translation of the character image into character codes, such as ASCII, commonly used in data processing.

Optical character recognition (OCR) is a process of converting a printed document or scanned page into ASCII characters that a computer can recognise.

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.

OCR methodology

-preprocessing

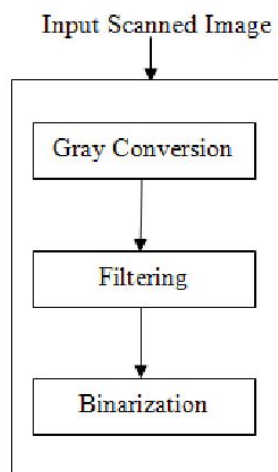


Figure number Preprocess Steps

-gray image

Grayscale images have many shades of gray. Grayscale images is result of measuring intensity of each pixel. For achieving accuracy input document should be grayscale.

To convert a colour from a color space based on an RGB colour model to a grayscale representation following function is used

$$Y = 0.2126R + 0.7152G + 0.0722B$$

-blurlama // yapmadim artik kodda

We can see that the image above needs further enhancement, therefore, we apply another blur to improve the looks

-binary

This step converts a multicolored image (RGB) to a black and white image. There are several algorithms to convert a color image to a monochrome image, ranging from simple thresholding to more sophisticated zonal analysis.

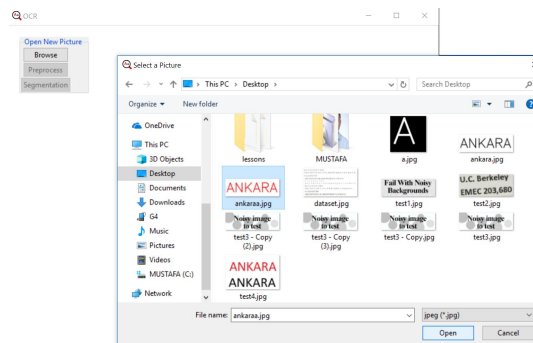
-segmentation

- find contour**
- sort contour**
- draws contour**
- crop contour**
- match template**

Implementation of methodology

-Browse

In this step, we store the path of the chosen image from the browse menu. Chosen image was read by cv2.imread() method. **.. yada bu cumle** Images can be imported into the GUI by clicking on the Browse menu and selecting Open. Both all file formats are supported.



-cv2.cvtColor()

When the preprocess button was pressed, this chosen image was converted to gray image by cvtColor() method. In this method converts an input image from one color space to another. COLOR_BGR2GRAY was used in our project. After the calculation gray image was shown.

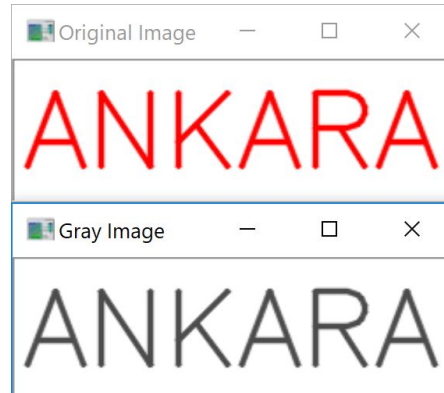


Figure number Original Image and Gray Image

`-cv2.GaussianBlur()` //sildim kodda siralamayi bozduyuyo bazen



Figure number Gaussian Blur Image

`-cv2.threshold()`

If pixel value is greater than a threshold value, it is assigned one value, else it is assigned another value . The threshold value is 127.



Figure number Binary Image

Preprocess is done. Now segmentation button is available to press.

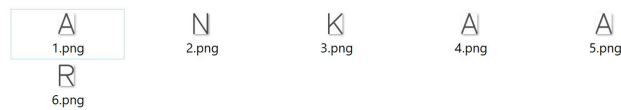
`-cv2.findContours()`

When the segmentation button was pressed, Contours in binary image was found and drawn yellow rectangle around the characters one by one.



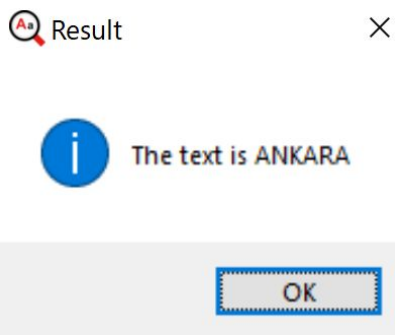
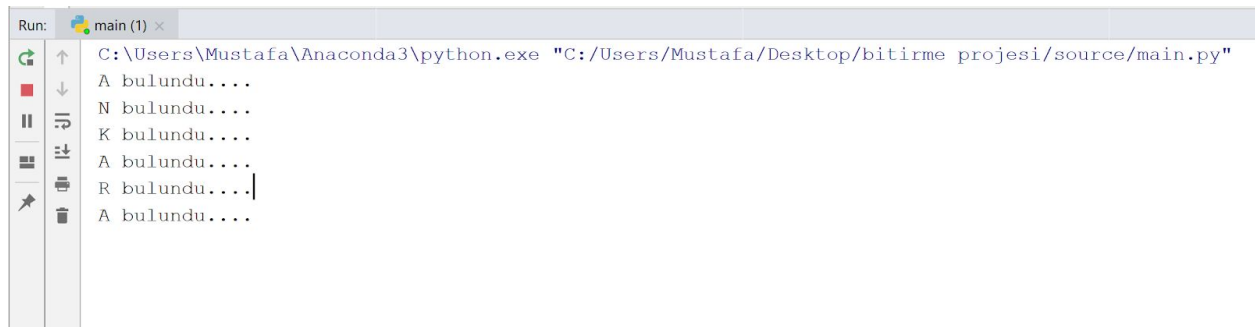
Figure number Segmentation

Characters were found and saved in IMAGEROI Folder. It's shown below.



cv2.matchTemplate()

Threshold max value is 0.8



Future Works

machine learning

Further work is initiated optical characters recognition for multiple font, lower case and size.

Conclusion

We have shown that Pattern Matching can be implemented successfully in optical character recognition. The system has image preprocessing and segmentation modules for image.