

DEPARTEMENT OF INFORMATICS AND TELECOMMUNICATIONS, UOA

IMAGE PROCESSING PROJECT

HOW TO DETECT AND RECOGNIZE TEXT IN NATURAL IMAGES USING



THEODOROS LOUKIS

ATHENS 2020

# **1. Abstract**

There are many applications in which the automatic detection and recognition of text embedded in images is useful. These applications include digital libraries, computerized aid for visually impaired, automatic geocoding of businesses, and robotic navigation in urban environments. In this project, a two-step procedure which automatically detects and extracts text in images is proposed. First, a four- step algorithm is used to detect the text in the candidate image and finally we recognize the meaning of the text by printing it in the user interface and transforming it to speech. The tool we use to apply this method to a candidate image is MATLAB a high-performance language for technical computing.

Keywords: image processing, text reading system, character recognition, text detection, MATLAB, background removal, MSER, OCR, computer vision.

# **2. Introduction to Image Processing**

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps:

* Importing the image via image acquisition tools;
* Analyzing and manipulating the image;
* Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction. In our project we dig into the world of Digital Image Processing as we use MATLAB to detect and recognize text in natural images.

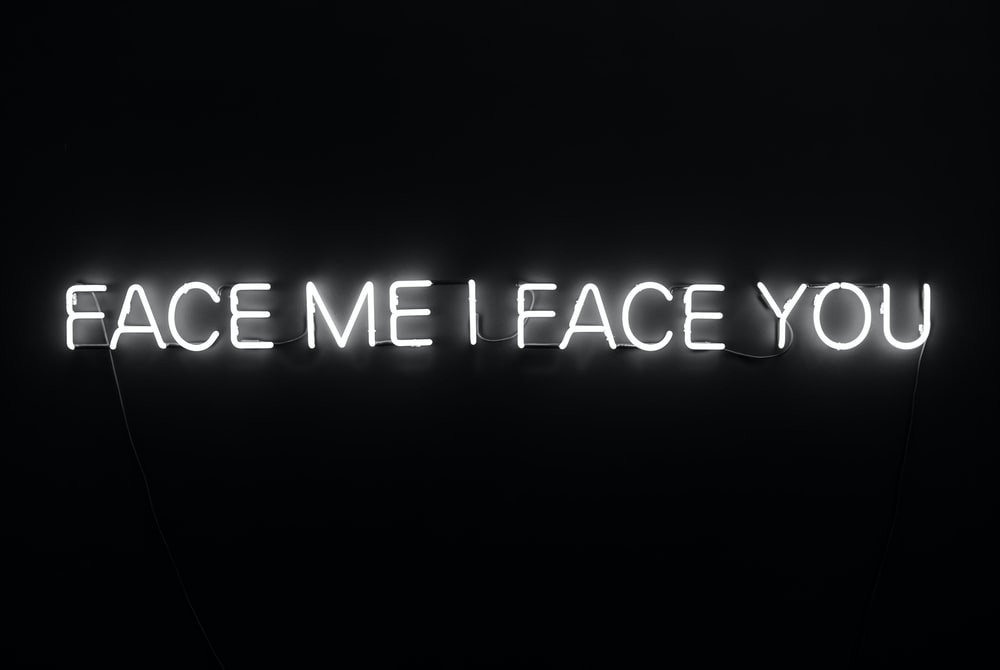
# **3. Detect and Recognize Text in Natural Images**

Detecting text in natural images is an important step for a number of Computer Vision applications, such as computerized aid for visually impaired, automatic geocoding of businesses, and robotic navigation in urban environments. Retrieving texts in both indoor and outdoor environments provides contextual clues for a wide variety of vision tasks. Detecting and recognizing text in natural images is a common task performed on unstructured scenes i.e. images that contain undetermined or random scenarios. For example, you can detect and recognize text automatically from captured video to alert a driver about a road sign. On the other hand, structured scenes, contain known scenarios where the position of text is known. Segmenting text from an unstructured scene greatly helps with additional tasks such as optical character recognition (OCR).

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text. This technique can be applied to a scanned document, a document’s photo, a scene-photo (for example the text on signs and billboards in a landscape photo) or a subtitle text superimposed on an image (i.e. television broadcast). OCR is widely used as a form of data entry from printed paper data records (e.g. passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data) and it is a widely used method to digitize hard copy text. In that way they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. Further, OCR is still an active field of research and its applications can be found in pattern recognition, artificial intelligence and computer vision.

In order to reach our task, we need to follow a two-step procedure:

1. detect text in natural images
2. recognize the meaning of the text and choose a compatible way to demonstrate it to the user (print, text-to-voice, etc.).



The image above is used to demonstrate our procedure step by step as we try to detect the text areas of the image and extract it by printing the text in the U.I. and by transforming the text to speech.

I. Detect Text in Natural Images:

At this point, a four-step algorithm which automatically detects and extracts text in images is proposed:

### Detect Candidate Text Regions Using MSER.

### Remove Non-Text Regions Based on Basic Geometric Properties.

### Remove Non-Text Regions Based on Stroke Width Variation.

### Merge Text Regions for Final Detection Result.

A). Detect Candidate Text Regions Using MSER:

MATLAB has a built-in function that can be helpful in order to detect features and it can be applied in projects to find text regions. MSER function can deal with the above because the consistent color and high contrast of text leads to stable intensity profiles. But what does MSER stand for? “In computer vision, **maximally stable extremal regions** (**MSER**) are used as a method of blob detection in images. This technique was proposed to find correspondences between image elements from two images with different viewpoints. This method of extracting a comprehensive number of corresponding image elements contributes to the wide-baseline matching, and it has led to better stereo matching and object recognition algorithms. “



B). Remove Non-Text Regions Based on Basic Geometric Properties:

After using the MSER to detect the candidate text regions we have to find a way to remove all non-text regions. MSER algorithm picks out most of the text, it also detects many other stable regions in the image that are not text regions. We can use a rule-based approach to remove non-text regions. For example, geometric properties of text can be used to filter out non-text regions using simple thresholds. Alternatively, we can use a machine learning approach where by training a text vs. non-text classifier the trained algorithm would be able to detect and classify the regions. Typically, a combination of the two approaches provides better results. In this work the above is beyond our scope and we propose a simpler rule-based approach to filter non-text regions based on geometric properties. There are several geometric properties that are good for discriminating between text and non-text regions including:

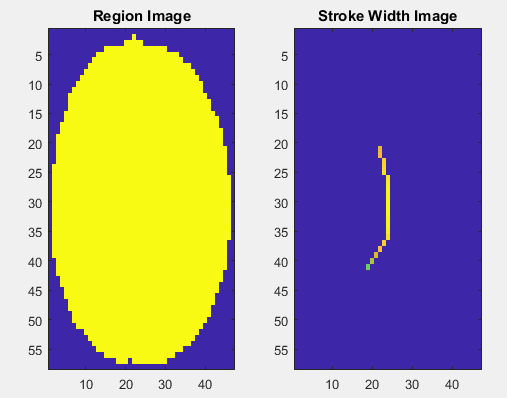
* Aspect ratio,
* Eccentricity,
* Euler number,
* Extent,
* Solidity.

Another built-in MATLAB tool “*region props*” will help us measure some of these properties and then remove the non-relevant regions based on their property values.



C). Remove Non-Text Regions Based on Stroke Width Variation:

Another common metric used to discriminate between text and non-text is stroke width. Stroke width is a measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations. To help understand how the stroke width can be used to remove non-text regions, estimate the stroke width of one of the detected MSER regions. You can do this by using a distance transform and binary thinning operation.

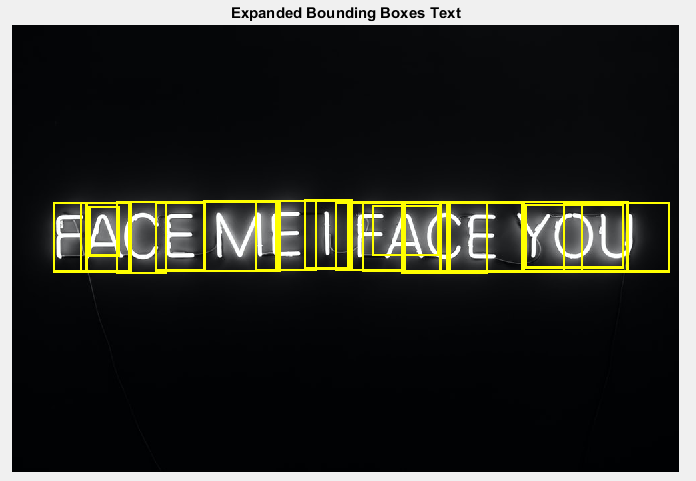


In the images shown above, notice how the stroke width image has very little variation over most of the region. This indicates that the region is more likely to be a text region because the lines and curves that make up the region all have similar widths, which is a common characteristic of human readable text. In order to use stroke width variation to remove non-text regions using a threshold value, the variation over the entire region must be quantified into a single metric. Then, a threshold can be applied to remove the non-text regions. Note that this threshold value may require tuning for images with different font styles. The procedure shown above must be applied separately to each detected MSER region.

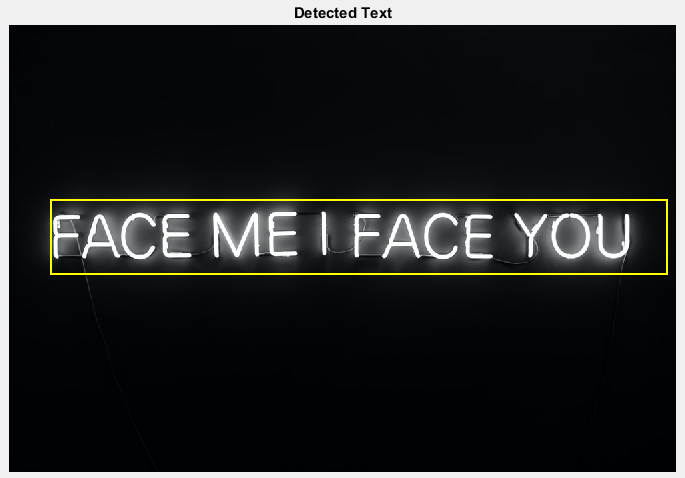


D). Merge Text Regions for Final Detection Result:

At this point, all the detection results are composed of individual text characters. In order to use these results for recognition tasks (i.e., OCR) the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the word “CANCEL” vs the set of individual characters, i.e., {C, L, A, E, N, C}, where the meaning of the word is lost without the correct ordering. An approach to merge individual text regions into words or text lines is to first find neighboring text regions and then form a bounding box around them. In order to find neighboring regions, we expand the bounding boxes computed earlier with “*region props*” function. This makes the bounding boxes of neighboring text regions overlap in a way that text regions that are part of the same word or text line form a chain of overlapping bounding boxes.



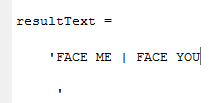
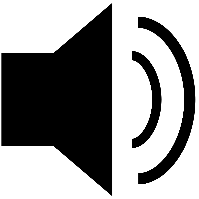
Now, the overlapping bounding boxes can be merged together to form a single bounding box around individual words or text lines. In order to do this, we need to compute the overlap ratio between all bounding box pairs. This quantifies the distance between all pairs of text regions so that it is possible to find groups of neighboring text regions by looking for non-zero overlap ratios. Once the pair-wise overlap ratios are computed, we use a graph to find all the text regions "connected" by a non-zero overlap ratio. Finally, before showing the final detection results, suppress false text detections by removing bounding boxes made up of just one text region. This removes isolated regions that are unlikely to be actual text given that text is usually found in groups (words and sentences).



II. Recognize and Display Text:

After detecting the text regions, we use MATLAB’s built-in OCR function to recognize the text within each bounding box. It should be noted here, that without finding the text regions (at the first step), the output of the OCR function would be considerably noisier. Finally, we choose between a variety of ways to showcase our results:

1. printing the text in the UI,
2. transforming the text to speech using the speech synthesizer class, available in .NET Framework.

a. b. 

# **4. Summary**

To sum up, detecting and recognizing text in natural images is a useful area of image processing that will open up many opportunities in future projects all around the computer science & engineering industry. In this project, a text detection algorithm is presented which employs maximally stable extremal regions (MSER), geometric properties and stroke width, to recognize text in natural images. Furthermore, this proposed method has a great variety of uses, such as in automotive technology (to recognize plate number and road signs), text extraction applications, image to text websites (such as easypdf.com) etc.

# **5. Future Work**

A passionate CS student and future engineer can always find time to improve his/her projects outside of the course’s radius, thus there is some future work that will be done in order to make the presented project as perfect as it can be. Like a lot of new projects, this one is not perfected yet. For example, the MATLAB script we use does not work as well with .png image files as with .jpg files. Furthermore, it does not always recognize text in images, as the text might be blurred or mixed with other geometrical shapes. Finally, there are some problems with the shape of the images too. The script cannot detect text in bigger images, as efficiently as in images with smaller dimensions. To conclude these corner cases, need to be dealt with if we want to talk about a fully functional tool that can automatically detect and recognize text in (all) natural images.

# **6. References**

[1] Chen, Huizhong, et al. "Robust Text Detection in Natural Images with Edge-Enhanced Maximally Stable Extremal Regions." Image Processing (ICIP), 2011 18th IEEE International Conference on. IEEE, 2011.

[2] Gonzalez, Alvaro, et al. "Text location in complex images." Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012.

[3] Li, Yao, and Huchuan Lu. "Scene text detection via stroke width." Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012.

[4] Neumann, Lukas, and Jiri Matas. "Real-time scene text localization and recognition." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[5] <https://en.wikipedia.org/wiki/Maximally_stable_extremal_regions>

[6] <https://www.mathworks.com/help/vision/ref/ocr.html>

[7] <https://en.wikipedia.org/wiki/Optical_character_recognition>