OCR with Machine Learning

Optical Character Recognition(OCR) is a process run by OCR software. The software will open a digital image, e.g., a tiff file containing full-text characters, and then attempt to read and translate the characters into recognizable full text and save them as a full-text file. This is a quick process that enables the automated conversion of millions of images into full-text files that can then be searched by word or character. This is a very useful and cost-efficient process for large-scale digitization projects for text-based materials, including books, journals, and newspapers. There are several OCR software packages on the market but a popular package for older material or that in languages other than English is Abbyy Finereader. This is currently being used by several newspaper digitization projects internationally.

Machine learning has emerged as a remarkable technology that empowers the automatic extraction and interpretation of text from images or scanned documents. This process entails training machine learning models on extensive datasets of images and their corresponding text labels, enabling them to accurately recognize and transcribe characters. To accomplish this, OCR systems employ an amalgamation of image processing techniques like noise reduction, image enhancement, and segmentation. These techniques facilitate the isolation of individual characters or words within an image. Subsequently, the extracted text undergoes further processing to enhance accuracy and overcome challenges posed by varying fonts, sizes, and orientations.

The OCR process is dependent upon a number of factors, and these factors influence results quite radically. Experience to date has shown that using OCR software over good quality clean images (e.g., a new PDF file) has excellent results, and most characters will be recognized correctly, therefore, leading to successful word searching and retrieval. However, over older materials, e.g., books and newspapers, the OCR is extremely variable, and for this reason, some projects advocate re-keying the text from scratch rather than attempting OCR. The process is labor intensive, and sometimes a combination of both re-keying and OCR will be performed for a project. It is usual to undertake sample tests on the actual source material to be digitized before making decisions about OCR and re-keying.

OCR Can help you save your time and your effort in extracting texts from images; you save the time spent typing the whole text by yourself.

There are some issues you should take care of :

- The quality of your image, the written content
- , the font size, you can separate the font from the background !! The font is skewed or distorted !!
- the size of the image

- , the quality of the light

ocr.space

It is an OCR engine that offers a free API. It means that it is going to do pretty much all the work regarding text detection. We only need to send through their API an image with the text we want to scan, and it will return the scanned text.

First of all, you need to get an API key.

Go to http://ocr.space/OCRAPI and then click on "Register for free API Key".

Note: The free OCR API plan has a rate limit of 500 requests within one day per IP address to prevent accidental spamming.

Code:

Importing Libraries

Loading the Image

Now we will load the image using OpenCV(CV2). Then, the image needs to be converted to a binary image, grayscaling it if it is an RGB image. Grayscaling takes the three RGB values of an image and transfers it with the following formula to a single value which represents a shade of gray. [0-255]: 255 being the brightest shade of grey (white) and 0 being the darkest shade of grey (black).

$$□ = 0{,}299.□ + 0{,}587.□ + 0{,}114.□$$

After grayscaling, there comes thresholding; thresholding is used to decide whether the value of a pixel is below or above a certain threshold.

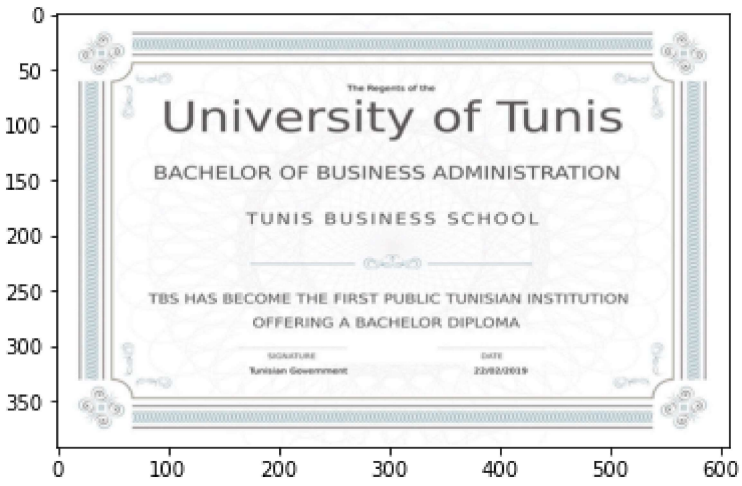
- If pixels < the threshold ===> turned to a white pixel
- If pixels > the threshold ===> turned to a black pixel

The result of 1 and 2 is that we get a binary image (white background and black foreground).

Output:

```
(608, 391)
```

Output:



After loading the image of the TBS bachelor, we need to set the OCR engine: send the image to the ocr. space server in order to be processed. Here there are a few notes :

1. Sending the image to the ocr. space server
2. Since we are using the free service, we can not send an image with a maximum of one MB of size, so we need to shrink the size of our image by compressing it.
3. Also, to send the image to the server, we need to convert the image into bytes.

Output:

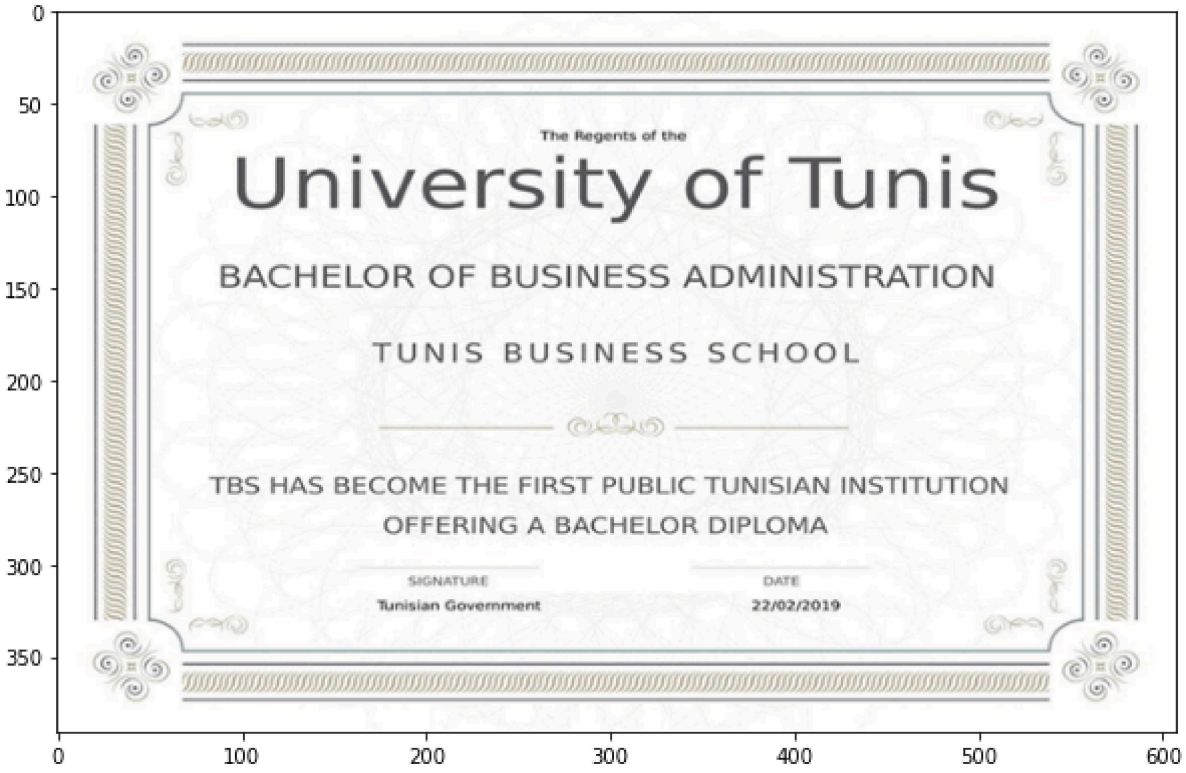
```
{'ParsedResults': [{'TextOverlay': {'Lines': [],
    'HasOverlay': False,
    'Message': 'Text overlay is not provided as it is not requested'},
   'TextOrientation': '0',
   'FileParseExitCode': 1,
   'ParsedText': "University of Tunis\r\nBACHELOR OF BUSINESS ADMINISTRATION\r\nTUNIS BUSINESS SCHOOL\r\nTBS HAS BECOME THE FIR
ST PUBLIC TUNISIAN INSTITUTION\r\nOFFERING A BACHELOR DIPLOMA\r\nGovement\r\n22/02'2019\r\n",
   'ErrorMessage': '',
   'ErrorDetails': ''}],
 'OCRExitCode': 1,
 'IsErroredOnProcessing': False,
 'ProcessingTimeInMilliseconds': '718',
 'SearchablePDFURL': 'Searchable PDF not generated as it was not requested.'}
```

Output:

```
"University of Tunis\r\nBACHELOR OF BUSINESS ADMINISTRATION\r\nTUNIS BUSINESS SCHOOL\r\nTBS HAS BECOME THE FIRST PUBLIC TUNISIA
N INSTITUTION\r\nOFFERING A BACHELOR DIPLOMA\r\nGovement\r\n22/02'2019\r\n"
```

Extracting Text Using Tesseract

Output:


The Regents of the
University of Tunis
BACHELOR OF BUSINESS ADMINISTRATION
TUNIS BUSINESS SCHOOL
TBS HAS BECOME THE FIRST PUBLIC TUNISIAN INSTITUTION
OFFERING A BACHELOR DIPLOMA
SIGNATURE
Tunisian Government
DATE
22/02/2019
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600

Output:

10)

5)

CA &

€

©
:
Sy
:
:
5
:
Ss
:
s
Ss
S$
Sy
:
s
S$
5
:

‘The Regents of the

: ‘University of Tunis

BACHELOR OF BUSINESS ADMINISTRATION

TUNIS BUSINESS SCHOOL

TBS HAS BECOME THE FIRST PUBLIC TUNISIAN INSTITUTION
OFFERING A BACHELOR DIPLOMA

“unision Goveenment 22027019

ooo @

—=_£iiiiiittiiiiii iii iiiiiiiiiiiii © 5°

Alternative Method

Output:

```
Collecting https://github.com/myhub/tr/archive/1.5.1.zip
  Downloading https://github.com/myhub/tr/archive/1.5.1.zip
     \ 177.5 MB 9.3 MB/s
Building wheels for collected packages: tr
  Building wheel for tr (setup.py) ... done
  Created wheel for tr: filename=tr-1.5.0-py3-none-any.whl size=162900360 sha256=3249bfc56f0b3d37802a0f0feb8f1dbe0c58fcd67799a1
f1e8a79f497cee78f2
  Stored in directory: /tmp/pip-ephem-wheel-cache-pg2o16in/wheels/e8/24/f6/fa325b41760077cc82fdb1745a4cb3ef7f6ac8fd8c5f37e6f2
Successfully built tr
Installing collected packages: tr
Successfully installed tr-1.5.0
```

Output:

```
197 255
```

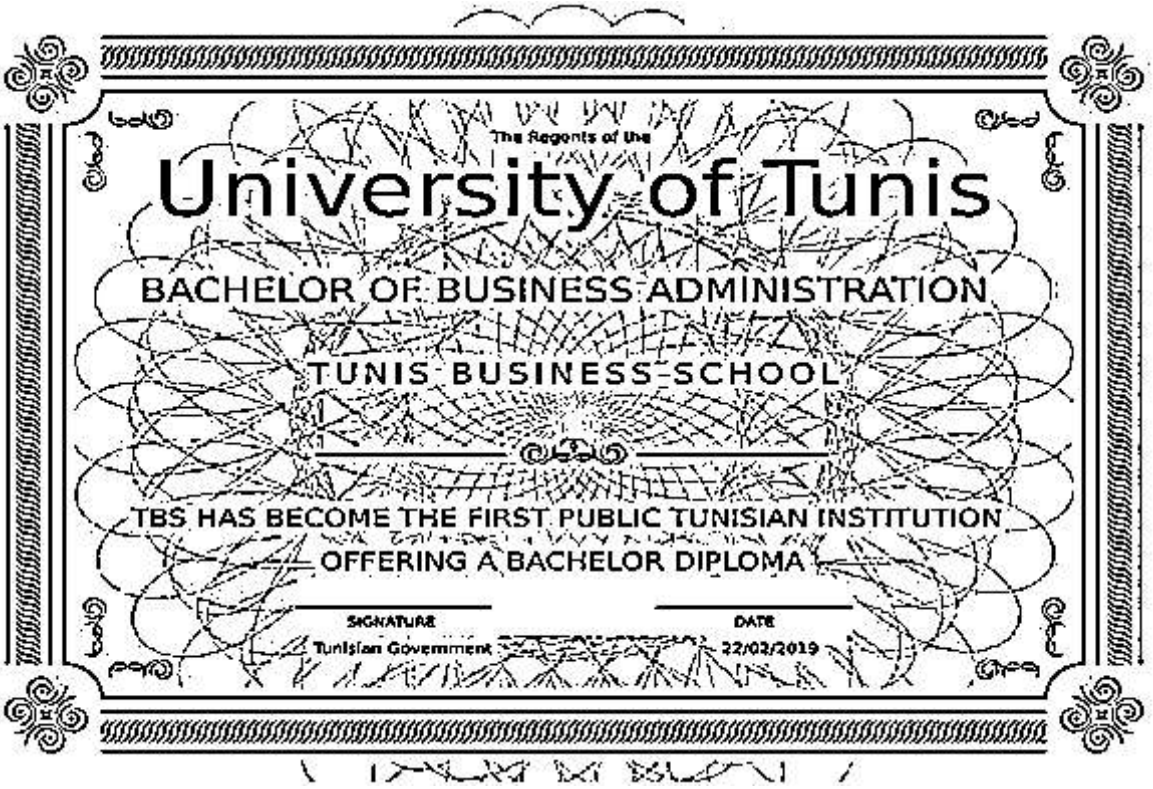
Output:

OpenCV

Output:

```
---------Start Recognize text from image---------
Cains a ae
& SKK LLL LL LLL



SSSI

SIKKIM LTE
Y 123k 1

--------Done----------
```

Files that were Generated during the above process:



Conclusion

In conclusion, OCR powered by machine learning is a transformative technology that revolutionizes the way we extract and interpret text from images and scanned documents. By leveraging large datasets and training sophisticated machine learning models, OCR systems achieve remarkable accuracy in recognizing and transcribing characters. The impact of OCR using machine learning

extends across various industries, enabling document digitization, streamlining form processing, and facilitating data analysis through text extraction from images. With its ability to automate information management tasks and enhance efficiency, OCR with machine learning stands at the forefront of innovation, opening up new possibilities for improved productivity and streamlined workflows in the digital age.