Document Data Extractor (OCR for Documents)

Adrian Roberto Carmona Rodriguez
Data Engineering
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 4448. CP 97357
Ucú, Yucatán. México
Email: st1809032@upy.edu.mx

Ing.Gabriela Flores Bracamontes
RICH IT
Pétalo 47, El Reloj, Coyoacán,
04640 Ciudad de México, CDMX.
Email: gflores@richit.ai

Dr.Juan Vázquez Montejo
Universidad Politécnica de Yucatán
Km. 4.5. Carretera Mérida — Tetiz
Tablaje Catastral 4448. CP 97357
Ucú, Yucatán. México
Email: juan.vazquez@upy.edu.mx

Abstract

In this Article you can find the development, the explanation and the application of an algorithm for extracting text in images, also called Optical recognition characters (ORC) for documents. Developed by and for the company RICH IT. The final product is an AI software capable of extracting information from official documents and to extract them into a database, which companies using the product can access and perform other activities.

Index Terms

OCR, RPA, Artificial intelligence, Matrix resolution, Pixel Dimension, Artificial vision, sharpness.

Document Data Extractor (OCR for Documents)

I. Introduction

THe intelligent automation(IA) is the combination of robotic process automation(RPA) and artificial intelligence(AI) technologies that together drive rapid end-to-end automation of business processes and accelerate digital transformation.

To expand the horizons of business process automation by an order of magnitude, intelligent automation combines the task execution of RPA with the analytics and machine learning capabilities of process discovery and process analysis, as well as cognitive technologies such as computer vision, natural language processing and partial logic.

Machine vision systems rely on digital sensors protected inside industrial cameras with specialized optics to acquire images, so that hardware and software can process, analyze and measure different features to make decisions. This allows companies to develop new sophisticated tools capable of solving multiple problems automatically.[1]

II. Objective

The purpose of this report is to document the work performed to achieve the text extraction for identifying documents, taking into account image evaluation, preprocessing, and finally text extraction. With this document, anyone who wants to continue developing this code can understand its operation without any problem, as well as to consult the changelogs in each of the code versions.

III. State of the Art

Research and development of OCR systems are considered from a historical point of view. The historical development of commercial systems is included. Both template matching and structure analysis approaches to RD are considered. It is noted that the two approaches are coming closer and tending to merge. Commercial products are divided into three generations, for each of which some representative OCR systems are chosen and described in some detail. Some comments are made on recent techniques applied to OCR, such as expert systems and neural networks, and some open problems are indicated. The authors' views and hopes regarding future trends are presented.[2]

Way back in 1943, the ENIAC, the computing machine was invented by J. Presper Eckert and John Mauchly at the University of Pennsylvania and was not completed until 1946. Some people say that abacus was the first computing machine. This is how the computing era started in decades back. Like a chameleon, the computing technology wearing different colors in terms of hardware like desktops, servers, laptops, mobiles and now entering into "rolltops". In addition to this, abundant software developments happened in terms of operating systems, applications, utilities and computing capabilities at the networking edge level with high bandwidth. Further, the organizational applications also stepped into many folds including punch cards, spreadsheets, office applications, management information systems and enterprise resource planning to assist in business applications. Nowadays the business operations are leapfrogging into the new technological land the so-called "Robotic Process Automation." Hence the year "2018" is known as the year of the "Robotic Process Automation." Industry entering into the in the new world of technology "Robotic Process Automation." known as "RPA".[3]

A. Actual Background

The process described here was based on the previous versions created and provided by RichIT. However, this part of the implementation only covers the detection of information on ID cards due to the short period assigned by this project.

The whole project includes three sprints divided as follows, describing how the project was at the beginning of the implementation described on this document:

1) Background:

- First iteration of Image Prepossessing.
  1) The image space is selected manually, however, it does not consider the possible defects that may exist when performing a scan, for example: incorrect orientation or position of the image.
  2) Automatic detection of the space where the text is located.
- Image processing
  1) Automatic contrast adjustment.
  2) Pixel mapping to identify the location of the text
- Information extraction

Based on the latter, the solutions and proposals for this project were developed as described in the rest of the document. All results were documented and shared just with the company by security question.

IV. Methods and Tools

A. Tools

- OpenCV
  1) OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.[4]
- Math
  1) The most popular mathematical functions are defined in the math module. These include trigonometric functions, representation functions, logarithmic functions, angle conversion functions, etc. In addition, two mathematical constants are also defined in this module.[5]
- Imutils
  1) A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3.
- Numpy
  1) NumPy is a library for the Python programming language that provides support for creating large multidimensional vectors and matrices, along with a large collection of high-level mathematical functions to operate on them.[6]
- PIL
  1) Pillow is a free and open source add-on library for the Python programming language that adds support for opening, manipulating and saving many different image file formats.[7]
- Exif
  1) Easy to use Python module to extract Exif metadata from tiff and jpeg files.[8]

V. Development process

A. Main.py

Jupyter Notebook containing the flow of the program as Main, calling the modules set aside. The functions must be stored in the same folder as the Jupyter Notebook.

B. Perspective.py

The first step on the whole preprocessing is to try to change the perspective of the image with the aim of avoiding unnecessary borders in the image.

The overall process of a change of perspective consists of the application of filters to prepare the image for the contours finding, mainly, for the Canny filter, which helps the next findContours function, within OpenCV, to identify the contour of the document ID.

Fig 1. Input image

Fig 2. Image Located

It is important to mention that the change of perspective might not be fulfilled as expected, however, if the image possesses the characteristics as stated on the requirements, there might not be a major problem on the extraction, so this step is just to have a close up of the whole document.

There might be some cases, however, when the document can look far from the perspective in which the photo was taken and the perspective will change successfully. Thus, this step on the preprocessing adds to the identification of the quality of the image parameter that might be validated on the following modules.

C. Rotation.py

In this module we cover a lot of steps, which were dedicated to specific tasks. The goal was to try to make a perfect ID rotation using all values that we have from the document. Another thing to take into account, is the fact that there might be the case in which some ID's can have a kind of Scan protection, something that can affect negatively on the data extraction. Also we got specific restrictions for processing an ID, the most important specification is the position of the document in the picture, in this case the picture must be in a horizontal position.

1) Steps::

- Rotating 90°

  First, the function tries to rotate the picture if the Y axis is bigger than X axis, a specific case in which the picture is in a vertical position. Since the

algorithm can not identify information in vertical position, this step takes considerable importance for the whole process.

- Rotating 180°

  After checking if the ID is in the right position we need to verify that it was successfully rotated. Thus, we need to check if the document is upside down since the algorithm will be affected on the text extraction if the image stays like that. For that, this step is to make a face detection to see if the ID has been completely rotated; as we know, if the scan shows us that the face is on the left side, then the ID is in the right position, otherwise, it will need a rotation.

Fig 3. Face Located

D. *Information.py*

This module was made trying to implement a checking section in which we can stop the algorithm if something would be wrong (requirements not fulfilled). In this case this function tries to check if the picture covers the dimensions, if it does not cover the range of dimensions, then the picture will not be able to continue with the process. Otherwise, the algorithm will continue with the rest of the process.

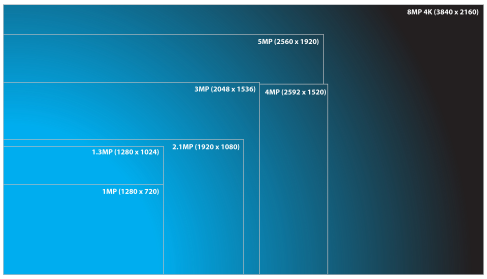

Fig 4. Types of resolutions

E. *Negative.py*

In this module the goal was to try to improve the extraction and detection of the text. It was considered using a black white filter, however, by doing some tests and researching, we could find that, if we use a negative filter and after that apply a black white filter, such preprocessing produces a clearer text even if the design of the ID have some noise.

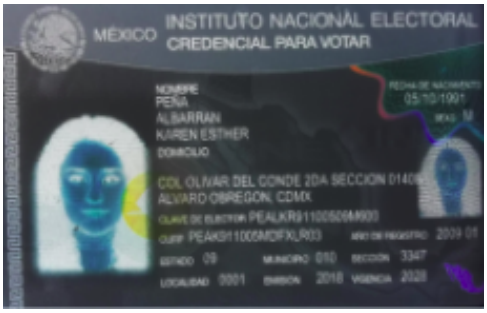

Fig 5. Negative filter

F. *Template.py*

After the negative filter is applied, a cut of the image will take part of the processing. This step allows us to adjust the image to only the area where the text is located, because if we pass to the algorithm any uncut credentials, the chances of a successful extraction decline. In addition, we can avoid the part of the photos and unnecessary backgrounds, if needed.

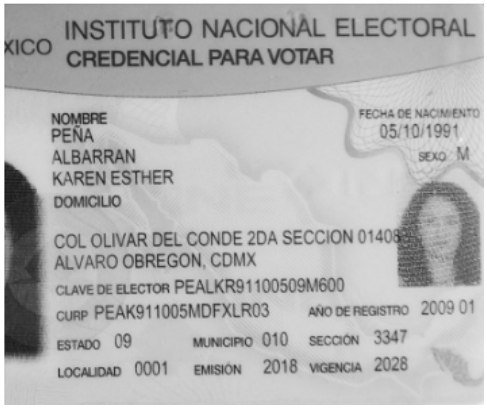

Fig 6. Image cut

G. *Getdata.py*

This function is one of the most complex. In this function we try to obtain an image with specific areas where text can be detected. So an array of coordinates is saved by means of a matrix calculation, in which data of the colors that are found are accumulated, based on 0 and 1, where the light colors and dark colors are indicated, knowing if any letter is found in the image or not.

Once all the text boxes are located, the algorithm tries to draw the image with all those boxes, so we can move on to the next task which would be to go through those boxes and extract text from those boxes.

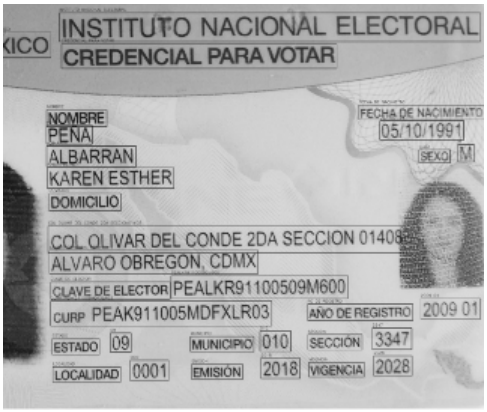

Fig 7. Boxes detected

Fig 8. Image of an valid ID updated with noise

H. Extract.py

In this part, the module is designed to be able to go through each part of the tables and make an extraction of each character. For this process we use a library called EasyOCR, so using the coordinates of the frames we can generate a more successful array of detected data, for that reason we must also take into account that if you do not have a GPU the time it takes for the algorithm to process an image is longer than a computer that possess said characteristic. On the other hand if you have a GPU the time is incredibly reduced.[9]

I. Ine$_a$ctual.py

This module contains two main functions. The first function is to give a check on the array obtained from the previous module. If there is any section within the extraction equals or even similar (60% of similitude as minimum), that means that we are dealing with a valid and non-updated ID. Said sections are either from “EMISIÓN”, “LOCALIDAD”, “ESTADO”, and “MUNICIPIO”, labels or keys that are present on the valid non-updated ID but omitted on the valid and updated ID.

The second function is to give a processing of the array-like format obtained from the previous module, in case that the latter function has verified that the ID being processed is actually an updated one.

Since the new ID presents a bigger area of the face image at the right side of the document, the representation letters on that small section can affect the text processing. Since the EasyOCR algorithm looks for text horizontally, the noise on the small right sided picture interferes with the analysis of the address, specifically. Therefore, the first step of this function is to clear that section out.

For that case, we implemented a solution based on the x-y coordinates of the rectangles in which text was detected. By having the coordinates and specific location provided by the array of the EasyOCR module, we got to obtain the middle point of the “AÑO DE REGISTRO” rectangle and the right bottom “SEXO” coordinates (taking the advantage of this part of the process, we store the value of “SEXO” to its variable from this step of the process). With the help of such an idea, we get the opportunity to state limits in which the noise might be located.

Fig 9. Limits for the noise to be eliminated

The process applied for the “NOMBRE” and “DOMICILIO” sections is exactly the same as explained in the following module, whereas the bottom part, from the “CLAVE DE ELECTOR” to “VIGENCIA” sections, the process might change from the process for the valid but non-updated ID’s. This part of the implementation is based on the coordinates of the rectangles of the label and the value, so the coordinates are located and the data stored on a single variable each.

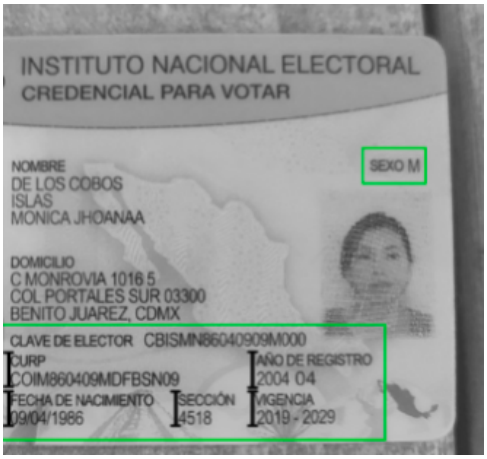

Fig 10. Sections to store

The output of this module is a list of values stored with the values from “NOMBRE” to VIGENCIA” including “SEXO”.

J. First.py

This is a small module that allows us to make an advanced function, in which we can locate a specific label called ”INSTITUTO”, since we know that this label is the one that starts the extraction and it is also where we can grab an area to locate things.

Once located at the coordinate of this box, we can make the following data location, since we only have to look for the label ”NOMBRE”. Once these two coordinates are obtained we have a specific area to select the data we want, thus eliminating unnecessary labels such as ”FECHA DE NACIMIENTO”.

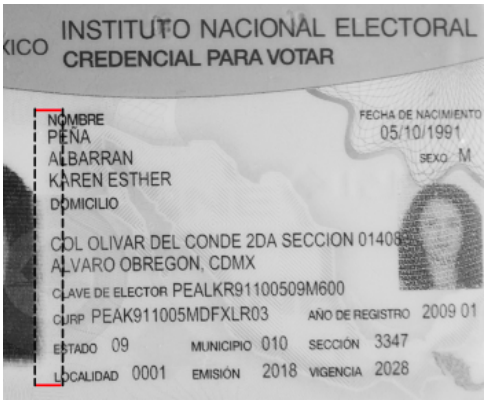

Fig 11. Area detected for “Nombre” and ”Domicilio”

K. Process.py

The purpose of this module is to be able to separate the detected text to a word in the array, which allows to split possible words that were detected together, thus improving the accuracy of the things that are being extracted. At the end of this process an array with the separated words is generated, which can be analyzed individually.

L. Values.py

The purpose of this module is to be able to generate rules used to correct our text detected and or extracted, so that the information is more accurate.

One of the most relevant changes that were made was: replace 1 by i, 0 by o, among other possible misdetected letters, and we know that the CURP and the CLAVE DE ELECTOR are data that already maintain some rules or a certain amount of digits that must be numbers and letters.

Another very big change was the correction of the sex, as mentioned above, the CURP and the CLAVE contain the sex of the person within a specific position, so a rule was developed in which we can know if the person is H or M according to their previous data.[10]

NOMBRE	APELLIDO_PATERNO	APELLIDO_MATERNO	SEXO	CLAVE_ELEC	CURP	ESTADO	MUNICIPIO
MARIO EDUARDO	ABURTO	GUTIERREZ	SEYa H	ABGTIR87103015}400	AUGMB71O3OHMCBTRO1	15	O4O
MARIO EDUARDO	ABURTO	GUTIERREZ	H	ABGTIR87103015H400	AUGMB71030HMCBTRO1	15	40
MONICA JHOANA4	DELOS COBOS	ISLAS	M	CBISMI86O4O9O9MOOO	COIM86O4O9MDFBSNO9		
MONICA JHOANAA	DELOS COBOS	ISLAS	M	CBISMI86040909M000	COIM860409MDFBSNO9		

Fig 12. Results of corrections

M. Actions.py

In this module the only thing to do is the organization of the information in a database, in this case it is a csv file. For its development we must first check if the folder exists and as second point to cover, we must create it, if it does not exist and inside it, generate the results of each extraction, besides generating the csv and the insertion.

VI. Time Results

One important thing about a whole process and/or algorithm it is about time complexity. It has been mentioned that the time of the code execution will decrease or increase based on the computer system used. In this section, characteristics of the system used are described to have an idea of the resources needed.

Computer features:

- Shell: bash 5.0.17
- Resolution: 1366x768
- Disk: 75G / 390G (21
- CPU: Intel Core i3-7020U @ 4x 2.3GHz [49.0°C]
- GPU: Intel Corporation HD Graphics 620 (rev 07)
- RAM: 3286MiB / 7746MiB

Archivo	Tiempo(M)	Dimensiones	Tipo
scan_03.jpeg	2.09692978	(3303, 2056)	6 MP
scan_17.2.jpg	1.59111483	(2229, 1574)	3 MP
scan_08.jpeg	0.61387333	(999, 639)	4CIF
scan_05.jpeg	0.64349619	(1013, 668)	4CIF
scan_06.jpeg	0.57457961	(1045, 747)	4CIF
scan_01.jpeg	2.03955402	(4128, 1908)	8 MP
scan_02.jpeg	2.36170853	(3889, 2442)	8 MP
scan_10.jpeg	0.62769737	(1007, 635)	4CIF

Fig 13. Time results for each image

Archivo	Tiempo(M)
3 MP	1.403951
4 CIF	0.615415
6 MP	2.044593
8 MP	2.212461

Fig 14. Means for each type of dimension

VII. Future Possible Changes

- Detailed correction (CURP, Clave de Elector).
  - An important improvement on the "CURP" and "CLAVE DE ELECTOR" fields, the processing rules that we already have might be a good approach to confirm that we are presenting accurate information. There is a chance of implementing more specific information rules focused on each of the characters and advanced text processing, since errors can still happen. The correction of the first characters of said fields can be performed since they are officially based on the first consonants of the full name, the same can be corrected for the section that makes references of the "FECHA DE NACIMIENTO" field.
- Other types of documents (Passport, license, etc.)
  - As you can see the algorithm is made for a single application. By this we mean that it can only process one type of document, in this case the INE, official Mexicans ID. But it is expected that in the future, this algorithm will be able to perform large multi-text analysis regardless of the type of text or document. So a similar process might be applied for the rest of the algorithms, considering each one of their limitations and requirements, just like the passports, which might be very strict on the way the document is captures as an image.
- Processing of any image
  - Another important point to cover is the fact of being able to make a wider extraction of photos, both of documents and credentials. In the future it is expected that the extraction limits will not be so specific and the algorithm will be able to extract even poorly taken, with a different position, or low quality images, as for now it only has certain dimensions and specific requirements.
- Automate labeling according to type of document
  - What is expected in the future with this improvement would be to be able to develop an algorithm that can extract information with a specific labeling. The objective is that it will not be necessary to do the location of coordinates. It would only be necessary to understand what type of word is being used, so that at the moment, if the algorithm detects a name, that data would be sent to a check.

    It is important to mention that research was done about NPL algorithms, with which they tried to do labeling, but it is not a very effective method when we talk about a massive load of documents. However, for other types of documents such as documents with more extensive redactions, it could be a great tool.

VIII. Conclusion

Artificial intelligence is becoming a very powerful tool to help people in times like this, when situations like the actual pandemic are limiting the society to do paperwork and other procedures that are most commonly to be required in person.

The development of this type of projects allows us to have a bigger vision of what could be our future, since many people think that technology will replace us. The reality is other, though, because technology is to improve activities with optimization on resources, since there are tasks that a person cannot do, such as reading millions of books in less than 5 minutes, among other things.

Despite our background and previous experience on programming and Data Science, we gather a main objective of proposing an implementation that might turn useful for the society, and that can be adopted in our daily lives forever. Having the opportunity of experience this project, along with a formal enterprise in charge of offering real life solutions with technology, has enhance our knowledge not only on hard skills, getting to solutions to be implemented within a code, for example, but also on soft skills, specifically by using the SCRUM methodology that requires a teamwork advance every day.

Artificial intelligence should be promoted, since thousands of people can be included in new activities, besides that it can greatly improve our technology and the efficient way we work. That is the importance of this type of algorithm, and the importance of a good team's organization might depend on that in order to achieve the desired goals.

Acknowledgment

I want to thank the Universidad Politécnica de Yucatán and the entire RICHIT team for giving us the support of being able to do our internships with them, since we were able to test our knowledge and be under pressure of the real labor field, which helps us to improve as future graduates. I also thank Dr. Juan Vazquez who gave us support in checking and writing the document, as well as contacting us and monitoring our progress.

IX. References

1) Madakam, S., Holmukhe, R. M., Jaiswal, D. K. (2019). The future digital work force: robotic process automation (RPA). JISTEM-Journal of Information Systems and Technology Management, 16.
2) Mori, S., Suen, C. Y., Yamamoto, K. (1992). Historical review of OCR research and development. Proceedings of the IEEE, 80(7), 1029-1058.
3) Madakam, S., Holmukhe, R. M., Jaiswal, D. K. (2019). The future digital work force: robotic process automation (RPA). JISTEM-Journal of Information Systems and Technology Management, 16.
4) Beyeler, M. (2015). OpenCV with Python blueprints. Packt Publishing Ltd.
5) Johansson, R., Johansson, R., John, S. (2019). Numerical Python (Vol. 1). Apress.
6) Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362.
7) Umesh, P. (2012). Image processing in python. CSI Communications, 23, 2.
8) Toevs, B. (2015, November). Processing of Metadata on Multimedia Using ExifTool: A Programming Approach in Python. In 2015 Annual Global Online Conference on Information and Computer Technology (GOCICT) (pp. 26-30). IEEE.
9) Vijayarani, S., Sakila, A. (2015). Performance comparison of OCR tools. International Journal of UbiComp (IJU), 6(3), 19-30.
10) Paliza, P. A. Corrección automática de errores de OCR en documentos semi-estructurados (Bachelor's thesis).