Tesseract OCR Recognition Based on Arabic Machine-Printed Document

Rakesh Ramteke(✉) and Mohammed Rashed Ali Omar Al Maamari

School of Computational Sciences, Kavayitri Bahinabai Chaudhary North Maharashtra University, Jalgaon, (MS), India
rakeshj.ramteke@gmail.com

Abstract. This paper provides technical aspects and the context of Recognizing and Detecting Arabic characters using Tesseract OCR Engine. OCR engine is freely available and gives a better result and also is supporting many languages such as Arabic etc. The procedure begins by transforming the Arabic documents into machine format (scanning) and then recognizing as well as extracting the text using the PyTesseract library. The OCR is a system that can afford the considerable values of split errors, particularly while working with cursive languages like the Arabic language with repeated overlapping between letters. Moreover, The performance is 99.5 accuracy in OCR-tesseract for converting the Arabic image documents to text editable.

Keywords: PyTesseract · OCR · Arabic · Recognizing · Detecting

1 Introduction

The Arabic language includes 28 letters and it is written differently from other languages such as English, and Hindi these languages can writhe from left to right but the Arabic language is totally different because it's written from right to left. There are many languages that depend on Arabic letters to write those languages such as Jawi, Urdu, and Persian. There are more than 26 nation whose formal language is Arabic and there are more than 280 million people speaks this language this makes the Arabic language so famous and used in the world. Each character has a different written format there are 5 letters from 28 letters written in two formats and the other 23 letters are having four formats(Initial, Middle, Isolate, End) [1]. OCR is a way to recognize the image text that is in a machine document format and transfer it into a machine-readable form to be used for data processing. The OCR system is improved its performance and started to be provided as a software package [2]. Python is the engine that used the PyTesseract library and it is one of the important libraries that are used for Arabic OCR, python is open source and it's easy to implement all the python libraries. Tesseract-OCR Engine is also used to detect the text in images such as line, word and character detection [3]. The optical character recognition to converts the images to the text editable with the.txt extension, then edit.py file is created to be compared between the predicate text and the truth text to check the accuracy of Tesseract OCR for recognizing the characters, that

R. Manza et al. (Eds.): ACVAIT 2022, AISR 176, pp. 347–355, 2023.
https://doi.org/10.2991/978-94-6463-196-8_27

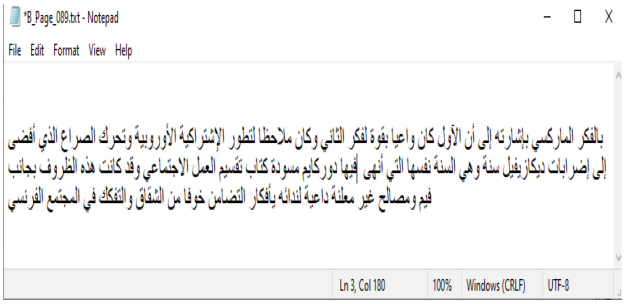


Fig. 1. (Output text)

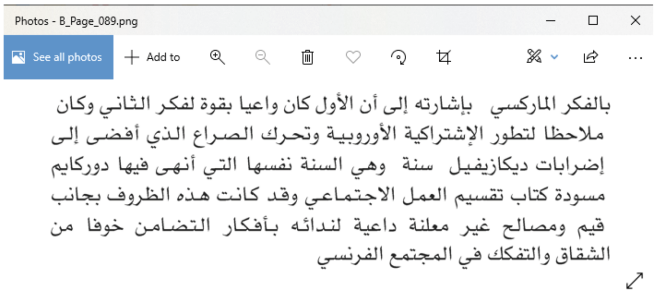


Fig. 2. (Image text)

run the edit.py file by cmd command to check the accuracy and how many characters are recognized wrong and it will account the error of recognizing and issuing the final accuracy the performance of the accuracy is 99.58% accuracy. OCR tesseract is also used in many languages such as English, Hindi, etc., just need to define the language that wants to use it in the OCR engine.

Figures 1 and 2 are the optical character recognition OCR firstly, input the text image that needs to convert to the text editable secondly, make some preprocessing methods as Noise Removal, Skew Correction, Image Resize, Grayscale conversion, Binarization then process it in pytesseract OCR engine to get the accurate text editable output with the .txt extension.

2 Literature Review

This paper, to the best of the authors' knowledge, is the first work in Arabic printed text OCR investigating a novel way to extract word features in the Block-based DCT (BDCT) domain. This is based on using a Discrete one-dimensional Hidden Markov (Bakis) Model (1D-HMM). The results in this area achieved on average 97.65% accuracy [4]. In this paper, the Tesseract engine is analyzed and modified for the recognition of the Urdu language which is a very difficult and cursive writing style of Arabic script. Original Tesseract system has 65.59% and 65.84% accuracies for 14 and 16 font sizes respectively, whereas the modified system, with reduced search space, gives 97.87%

and 97.71% accuracies respectively [5]. In this paper, they had created another OCR approach that depends on HMM for recognition. In their model, a tri-gram model of the character series is implemented. The process achieves a high accuracy rate between 98.9 and 97.71% [6]. In this paper, The system is used on actual printed word images with no overlap between the training and testing datasets. Word advantage vectors are extracted using block-based DCT. A Hidden Markov Models Toolkit (HTK) is used to build the recognizer. Author used Vector Quantization to map every feature vector to the closest symbol in the codebook. The result of the system is various recognition hypotheses (N-best word lattice). The output is hopeful when compared with other published research in this area achieving on average 97.65% accuracy which is significantly higher than previously published results [7]. In this paper, The systems are classified into two groups depending on the operating system MFR and REGIM-LITIS, SP-Curvelet-FR, RDI-CU, GU Font Recognition systems are built under Microsoft Windows environment and MindGarage systems under Linux. They tried to recognize the font size without recognizing the font or the content of the word image. The MindGarage system shows better results with a medium of 99.67% font-size identification rate. The REGIM-LITIS and RDI-CU systems show better results in some tests. The font-size identification rates with the Arabic word images created with bold font, font size 18 are respectively 99.98% and 94.96% with REGIM-LITIS and RDI-CU systems [8]. In this paper, they used rescaling and tesseract OCR to perform accurate Arabic handwritten character recognition. Moreover, used the long short term memory (LSTM) to get better accuracy in Arabic recognition and they got 52% accuracy in Arabic handwritten characters [1]. In this paper, they presented deep learning in OCR techniques to read and recognize text from the tax card image. Moreover, used a convolutional neural network in OCR(Tesseract) model that extracts the features from the tax image card. The score of the CNN-based OCR(Tesseract) model is 80% accuracy [16].

3 Methodology

A. Image preprocessing

The preprocessing step is the second step which is very important in any image detection and recognition system. The objective of this step in handwritten and machine-printed text recognition is to improve and enhance the readability of the text image and remove unnecessary details from it. The preprocessing steps usually include various operations like Image Resize, Grayscale Conversion, Correction Skew, Baseline Detection, and Noise Removal. The system can apply one or more from the preprocessing operations. It can also pass this stage without applying any of the operations if the input data are previously preprocessed [9].

- Grayscale conversion
  The colored image in this step is converted to a grayscale image. So, instead of working with a three-channel image, we can work only with a one-channel image. Color to grayscale conversion algorithm is required to preserve the salient features of the color images, like contrast, brightness and structure of the color images [10].
- Binarization

It is also called thresholding which is the operation of converting a text image into a binary format. Moreover, it is called a black and white image. The values of background pixels are 0 for black and 1 for white, this operation improves processing speed. There are many methods used in image binarization like the fixed threshold method, mean value threshold method and Otsu method [9].

- Image Resize
  The image for the best performance. It is maximum recommended for small images. The scaling property is used to scale the image. It is used to resize. The Tesseract OCR Engine works fully on images with a size of 150, 300 or 600 dpi. The image resize can play well in the image accuracy because the shape of the image helps to recognize the words well [11].
- Noise Removal
  Most of the data acquisition process get affected by noise, so for that most of the data get affected and there is no solution better than process remove noise. There are many ways used to remove image noise and perform smoothing like filtering and morphological operations [9].
- Skew Correction
  Skew correction and text line extraction are the main steps for tesseract OCR applications. For this objective, many access was developed, which conduct the analysis firstly in document images. Text lines in document images are often not aligned with the predictable direction, they are skewed. However, many text lines analysis processes and OCR engines require to skew corrected documents images and credible text line extraction for achieving perfect tesseract OCR performance [12].

B. Image detection and recognition

Image Detection and Recognition is a classic machine learning and Deep learning problem. It is a challenging job to detect an object or to recognize a documents image from a digital image [12]. Tesseract OCR performs in two different steps: recognition and detection. Firstly, we detect rectangular regions in the image potentially containing text. The detect can apply for line detect, word detect and character detect. Secondly, text recognition is performed to extract the text from document images and we can extract and recognize the text in images [12].

C. PyTesseract

OCR_PyTesseract is an optical character recognition engine with open source code, OCR_Tesseract is a more popular and specific OCR library. OCR_Tesseract engine was primarily developed at Hewlett Packard in 1984 to 1994, with many changes made in 1996 to transfer to Windows system. it appeared from nowhere for the 1995 UNLV yearly test of Optical character recognition (OCR) Accuracy, rise brightly with its results, and then vanished back under the gown of secrecy under which it had been developed. It suggested as well as demonstrated that Optical character recognition optical character recognition engines can feature from the use of an adaptive classifier. When the static classifier has to be perfect at generalizing to any form of font, its ability to recognize between various letters or between characters and non-characters is weakened [13]. Tesseract is finding templates in pixels, letters, words and sentences. It uses a two step

process that calls adaptive recognition, It requires one data phase for character recognition, then the second phase was to realize any letters, it wasn't covered, by letters that can identify the word or sentence context.

D. Working of Tesseract

pytesseract OCR works in a step by step method, First phase is Adaptive Thresholding which converts the image into binary image form. The next phase is connected component analysis which is used to extract character boundaries. This operation is most useful because it does the OCR of images with text and black background. Tesseract was probable the first, to provide this kind of processing. Then, the outlines are converted into Blobs. Blobs are orderly into text lines, and the lines and regions are analyzed for some fixed area or equal text size. Text is split into words using specific spaces and fuzzy spaces. Recognition of text is started as two-pass. An image with the text is given as input to the Tesseract engine that is a command based tool, Then it is treated by the Tesseract command. Tesseract command takes two cases: The first case is the image file name that contains text and the second case is the output text file in which, the extracted text is stored. The output file extension is given as txt by Tesseract, so it is not required to assign the file extension while assigning the output [14].

Figure 3 is showing the whole work in Arabic Document Recognition and Detection using pytesseract ocr the process is starting to convert the worK as two methods: Detection, Recognition. The detection is starting to use some methods like Thresholding is using to converts the color and grayScale image to binary image simply to black and white form to be easy to use and, and is combined three types of detection in this work like line, word and character will detect all and draw a rectangle for each char in the image. Finally, it recognizes the image text and it will be divided into three: line, word and char for recognition of every char in the image and give accurate output text using pytesseract ocr. Then edit.py file is created to be compared between the predicate text and the truth text to check the accuracy of Tesseract OCR for recognizing the characters

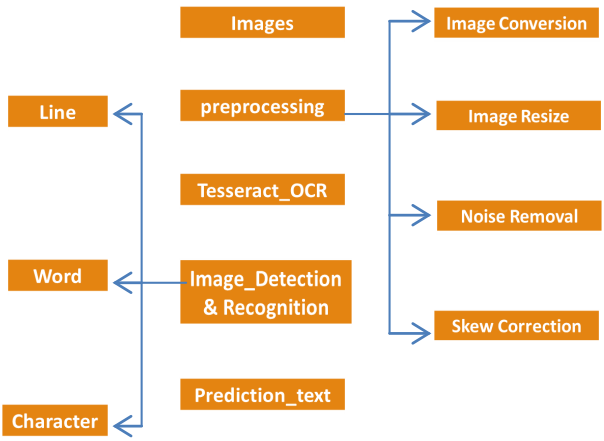


Fig. 3. METHODOLOGY

Table 1. Results And Discussion

No in references	Authors Names	Methodology and Adoptive	Accuracy
4	Krayem, A., Sherkat, N., Evett, L., & Osman, T.	HMM and block-based DCT	97.65%
5	Hussain, S., Niazi, A., Anjum, U., & Irfan, F.	Tesseract engine	97.71%
7	Nashwan, F., Rashwan, M. A., Al-Barhamtoshy, H. M., Abdou, S. M., & Moussa, A. M	Hidden Markov Models (HMM)	97.65%
16	Prawiro, T. G., & Khasanah, A.	Tesseract OCR Based on Long Short Term Memory	52%
17	Uddin, Q.	CNN-based OCR(Tesseract) model	80%
	Proposed method	Preproceesing method,Tesseract engine and cnn model	99.58

then run the edit.py file by cmd command to check the accuracy and how many characters are recognized wrong and it will account the error of recognizing and issuing the final accuracy (Table 1).

4 Results and Discussion

E. Arabic language with pytesseract-OCR

Arabic letter shape is context-sensitive, some Arabic letters have up to four different shapes depending on their relative position in the word, This fact increases the number of classes to be recognized by the OCR Engine. The Arabic language character consists of 28 letters and is written from right to left in cursive. Arabic letters are used to write various languages like Urdu, Persian, and Jawi. Arabic is the official language of more than 26 countries and it is spoken by 280 million people worldwide. Every Arabic letter has two or four shapes depending on its location in the text [1] Arabic writing dots are very important and they are not few as fifteen letters out of twenty-eight have dots above or below them. Arabic writers do not place these dots carefully in their proper place and this leads to much confusion. The Arabic letters are 28 letters that one of the issues is faced in this work and every letter has two or four forms the number f letters with four forms is 22 letters Fig(c), and with two letters are 6 letters Fig(d) (Table 2).

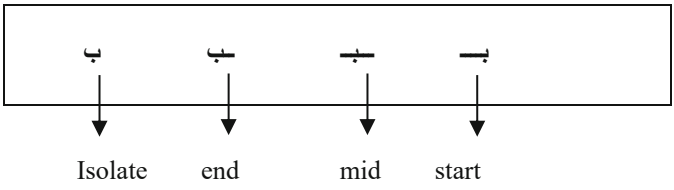
As shown in Table 3 is the letter of ‘ب’ is one of the 22 letters with four forms and the letter is written in the start, mid, end and isolated in every form is written differently that making the Arabic OCR is difficult.

As shown in Table 4 is the letter of ‘ا’ is one of the 5 letters with two forms and the letter is written in the Initial/isolate same the form and in the mid/end is written same the form.

Table 2. The Arabic letters

No	Characters	Initial	Middle	Isolate	End
1	Alif			ا	ـا
2	Baa	بـ	ـبـ	ب	ـب
3	Taa	تـ	ـتـ	ت	ـت
4	Thaa	ثـ	ـثـ	ث	ـث
5	Jiim	جـ	ـجـ	ج	ـج
6	Haa	حـ	ـحـ	ح	ـح
7	Khaa	خـ	ـخـ	خ	ـخ
8	Daal			د	ـد
9	Dhaal			ذ	ـذ
10	Raa			ر	ـر
11	Zaay			ز	ـز
12	Siin	سـ	ـسـ	س	ـس
13	Shiin	شـ	ـشـ	ش	ـش
14	Saad	صـ	ـصـ	ص	ـص
15	Daad	ضـ	ـضـ	ض	ـض
16	Taa	طـ	ـطـ	ط	ـط
17	Zaa	ظـ	ـظـ	ظ	ـظ
18	Eayn	عـ	ـعـ	ع	ـع
19	Ghayn	غـ	ـغـ	غ	ـغ
20	Faa	فـ	ـفـ	ف	ـف
21	Qaaf	قـ	ـقـ	ق	ـق
22	Kaaf	كـ	ـكـ	ك	ـك
23	Laam	لـ	ـلـ	ل	ـل
24	Miim	مـ	ـمـ	م	ـم
25	Nuun	نـ	ـنـ	ن	ـن
26	Haa	هـ	ـهـ	ه	ـه
27	Waaw			و	ـو
28	Yaa	يـ	ـيـ	ي	ـي

Table 3. Letters with four forms



There are three types of the letter with dots in Arabic letters are characters with one dot, two dots, three dots As shown in Table 5.

Table 4. Letters with two forms

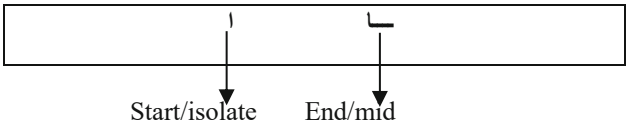


Table 5. Letters with dotst form

Type	Characters
One dot	ب ج خ ذ ز ض ظ غ ف ن
Two dots	ت ق ي
Three dots	ث ش

5 Conclusion

Firstly, Tesseract OCR is used in the Arabic language to convert the document images to the text editable. Secondly, we applied to preprocess techniques to convert the colour image to grayscale colour then image resize or reshape, remove the noises and skew detection. Thirdly, we applied the tesseract OCR to detection the line, word, character from the image and make a rectangle for each line, word and character. Moreover, The performance is a 99.5 accuracy in OCR tesseract for converting the Arabic image documents to text editable.

References

1. Prawiro, T. G., & Khasanah, A. (2020). The Effect of Rescaling on the Performance of Recognition with Arabic Characters Using Tesseract OCR Based on Long Short Term Memory. Journal of Advances in Information Systems and Technology, 2(2), 59-62.
2. Ko, D., Lee, C., Han, D., Ohk, H., Kang, K., & Han, S. (2018). Approach for Machine-Printed Arabic Character Recognition: the-state-of-the-art deep-learning method. Electronic Imaging, 2018(2), 176-1
3. Jayoma, J. M., Moyon, E. S., & Morales, E. M. O. (2020, December). OCR Based Document Archiving and Indexing Using PyTesseract: A Record Management System for DSWD Caraga, Philippines. In 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM) (pp. 1–6). IEEE.
4. Krayem, A., Sherkat, N., Evett, L., & Osman, T. (2013, August). Holistic Arabic whole word recognition using HMM and block-based DCT. In 2013 12th International Conference on Document Analysis and Recognition (pp. 1120–1124). IEEE.
5. Hussain, S., Niazi, A., Anjum, U., & Irfan, F. (2014, April). Adapting Tesseract for complex scripts: an example for Urdu Nastalique. In 2014 11th IAPR International Workshop on Document Analysis Systems (pp. 191–195). IEEE.
6. Alkhateeb, F., Doush, I. A., & Albsoul, A. (2017). Arabic optical character recognition software: A review. Pattern Recognition and Image Analysis, 27(4), 763-776.

7. Nashwan, F., Rashwan, M. A., Al-Barhamtoshy, H. M., Abdou, S. M., & Moussa, A. M. (2018). A holistic technique for an Arabic OCR system. Journal of Imaging, 4(1), 6.
8. Slimane, F., Ingold, R., & Hennebert, J. (2017, November). ICDAR2017 Competition on Multi-Font and Multi-Size Digitally Represented Arabic Text. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 1466–1472). IEEE.
9. Balaha, H. M., Ali, H. A., & Badawy, M. (2021). Automatic recognition of handwritten Arabic characters: a comprehensive review. Neural Computing and Applications, 33(7), 3011-3034.
10. Güneş, A., Kalkan, H., & Durmuş, E. (2016). Optimizing the color-to-grayscale conversion for image classification. Signal, Image and Video Processing, 10(5), 853-860.
11. Malathi, T., Selvamuthukumaran, D., Chandar, C. D., Niranjan, V., & Swashthika, A. K. (2021, February). An Experimental Performance Analysis on Robotics Process Automation (RPA) With Open Source OCR Engines: Microsoft Ocr And Google Tesseract OCR. In IOP Conference Series: Materials Science and Engineering (Vol. 1059, No. 1, p. 012004). IOP Publishing.
12. Li, W., Breier, M., & Merhof, D. (2015, September). Skew correction and line extraction in binarized printed text images. In 2015 IEEE International Conference on Image Processing (ICIP) (pp. 472–476). IEEE.
13. Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018, December). Convolutional neural network (CNN) for image detection and recognition. In 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) (pp. 278–282). IEEE.
14. Borisyuk, F., Gordo, A., & Sivakumar, V. (2018, July). Rosetta: Large scale system for text detection and recognition in images. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 71–79).
15. Smith, R. (2007, September). An overview of the Tesseract OCR engine. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp. 629–633). IEEE.
16. Uddin, Q. (2021). Features Extraction of Tax Card by Using OCR Based DeepLearning Techniques (Master's thesis, Itä-Suomen yliopisto).