iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.34926

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Optical Character Recognition based Webapp

Akshay Gharde[1], Asit Damke[2], Pratik Sahare[3], Sumit Tonge[4], Vipin Surwayanshi[5]
[1, 2, 3, 4, 5]*Computer Science and Engineering, Anjuman College Of Engineering And Technology, Nagpur, Maharashtra, India*

Abstract: ***As the use of computers in our daily lives increases, so has the need for a natural procedure to interact with the computers. The ultimate aim of human computer interaction is to bring the change that there is always a natural way of interacting with computers coupled with ease and flexibility. Printed and textual media such as prescriptions, invoices, receipts, etc. occupies a large segment of our day-to-day activities and given their volume, it is inefficient to manage them physically as there's always an associated risk of fading, damage, misplacing, etc. and hence a medium is required for their digital conversion. In this project, we have developed a robust, cross-platform web application that can process the images using PyTesseract based algorithms that can efficiently extract the textual data to facilitate the storage and retrieval of the same. The extracted text can be downloaded as a text file and can also be translated into the desired language. This is an active field of research and thus this paper also discusses various current implementations of the mentioned concept. The Optical Character Recognition framework finds applications in a variety of fields such as business process activities, number plate recognition, KYC and banking processes to name a few.***

Keywords: OCR, PyTesseract, Image Processing, Text detection, Text recognition, Digitization, Django, Web app.

I. INTRODUCTION

We are living in a data driven society where the generation and consumption of data is witnessing an exponential growth. As per Forbes 2018 report [1], we are creating 2.5 quintillion bytes of data every single day. While a large volume is in a digitized format, the physical form of data sources such as textbooks, invoices, prescriptions, reports, bulletins, etc. occupy a significant portion and are expected to grow in coming times.

The physical sources of data do possess inherent challenges of storage and safeguarding and are vulnerable to fading, damage or being misplaced, even the cloning requires a lot of manual effort. The data present in physical form needs to be transformed into digital format to facilitate efficient storage, retrieval, sharing and back up of the information.

The Optical Character Recognition (OCR) algorithm enables us to examine physical data stores (printed, typed or handwritten text) and transform them into editable digital formats. OCR based softwares find applications in various domains such as banking (example - KYC processes), business process, traffic control (number plate recognition), academics (physical books to PDFs), legal records, etc. This paper aims at developing a robust, cross-platform webapp wherein the users can scan and upload the images containing text, the OCR model that runs in the backend detects and recognizes the text present in the ingested image and outputs the same into desired language.

The OCR implementation faces the challenge of noise present in the images, as a result of which the text recognition accuracy stumbles. To overcome this, we make use of the image segmentation techniques to localize the areas that actually contain text and thereafter we use the OCR model to do text recognition. In this work, we have developed our OCR pipeline using PyTesseract which is a python wrapper of the open-source OCR engine which began as a research project at HP labs [2].

In the next section, we have discussed various current implementations of OCR in different domains to provide our readers a strong background and context of this active field of research.

II. LITERATURE SURVEY

In [3], Jaymer M. Jayoma et al. have developed an OCR based record management framework that can index and archive documents. They have developed this solution for *The Department of Social Worker and Development (DSWD), Caraga* which generates large volumes of data in physical formats out of their activities on a regular basis. The researchers have worked on eliminating the conventional process and automated the entire workflow to come up with a solution based on PyTesseract that can convert physical media into digital format which can then be efficiently archived, classified and indexed along with comfortable and safe storage and retrieval of the converted electronic documents. The researchers have used Django and MySQL to develop the backend.

©IJRASET: All Rights are Reserved

In [4], E.Pavithra et al. have developed a portable solution for specially-abled blind community that makes use of OCR methodology to identify and extract text from product labels and currency notes. This implementation is yet another interesting application of the wonders OCR can do and the ease it brings with it. Their solution is deployed on the portable miniature computer *Raspberry Pi* attached with a camera that acquires the image. The authors have used the OpenCV library to perform preprocessing and edge detection. Thereafter, text recognition is performed using the OCR and the extracted text is converted into a playable audio using the espeak library. In [5], Rekha M, et. al. have worked on building a framework that can process the invoice images and identify vendors and payment information from the images using Optical Character Recognition. This feature eliminates the labor-intensive manual invoice processing and automates the entire workflow resulting in improved efficiency and faster turn around time. The software can recognize principal identifiers such as invoice numbers, amount, bill and transaction ids from the receipts and can also calculate the net statement from all the receipts across vendors.

In [6], Ahmad P. Tafti et al. have performed a thorough review of the current state-of-the-art OCR platforms namely Docs OCR [Google Docs] [7], Tesseract [8], FineReader (ABBYY) [9], and Transym [10]. The authors have analyzed the reliability and robustness of the mentioned OCR platforms by calculating their accuracy on a rich dataset of 15 classes consisting of 1227 images. The authors conclude their review by providing a comparative analysis, accuracy scores and their insights of the same which can be utilized by the research community to find perfect fit of the OCR platform for a given problem statement.

In [11], Luke V Rasmussen et al. have worked on the development of an automated modular OCR system that can process and digitize the handwritten medical records which is otherwise highly labor-intensive. Medical records are highly critical and proper organization as well as maintenance is very necessary. The conversion of such records into Electronic Health Records or EHR not only eliminates the hassles involved with the conventional system but also makes it super flexible and portable. The EHR data can be made globally available at any given time which can really come in handy in a lot of critical situations.

In [12], Sharma, P. S. et al. have worked on the development of a real-time OCR traffic application that can detect vehicle license plates and extract the printed text even in unfavourable conditions such as poor lighting, weather, weird fonts, etc. This high performance is attributed to the fact that before using the OCR engine, the researchers use Haar cascade and filtering algorithms to process the image and localize the area of interest, once that is identified the localized image segment is ingested into the OCR engine for text recognition. This work is particularly useful to curb the traffic violators as well as easily keep a record of various checkpoints. One major assumption of this work was that license plate characters are present in a single line.

In [13], Singh, A. et al. have presented a thorough review of the various applications of different domains that are based on OCR. The authors have discussed the applications from the domains of invoice processing, legal industry, banking operations, digital libraries, healthcare tools, captcha recognition, musical note recognition, handwriting recognition and number plate recognition. In between, authors have also provided experimentations of a few out of the mentioned fields. The authors have also discussed the standard text extraction algorithm and proposed their own with an approach to redistribute the intensities after histogram equalization to enhance the performance of character recognition in bimodal images.

III. METHODOLOGY

A. System Architecture

Our proposed system as described in figure 1 consists of a Django (Python Framework) based web application wherein users can register and perform login/logout functionality. We have used sqlite3 as our database. Once the user successfully logs into the system, the user can upload the scanned image of the document or anything with embedded text and hit the process button. The process button triggers our OCR pipeline which consists of image pre-processing operations such as noise removal and localization. The processed image is fed into the PyTesseract OCR engine where textual characters are identified and categorized. The OCR engine outputs the extracted text which can be translated into the desired language using Googletrans Python library that internally implements Google Translate API.

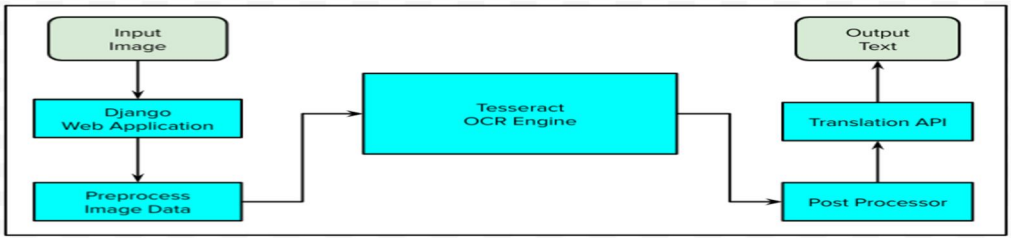


Figure 1. Architecture of the proposed system.

©IJRASET: All Rights are Reserved

B. Tesseract OCR

While there are many OCR tools available, Tesseract remains a leader throughout. Our OCR engine is based on PyTesseract which is a python wrapper for the open-source Tesseract toolkit. As mentioned in the introduction, the development of Tesseract began at HP labs (1984) as part of a research project and in the year 2005, it was open-sourced by HP. Currently, it is being maintained and developed by Google [14] with a superior ability to recognize text in more than 100 languages. This is an extensible implementation that can be further trained through machine learning to recognize other languages as well. One of the most common uses of this is Google's Gmail spam email image detection to identify the text from spam images and categorize a given email as spam/not spam.

C. Tesseract Ocr Architecture

Tesseract OCR is made up of numerous building blocks, the first being the Adaptive Thresholding which is used to convert the input image into binary image. The converted binary image undergoes connected component analysis which is a prominent step of the pipeline as it identifies the outlines of the characters. THe identified outlines are transformed into blobs which are further structured into lines of text divided into defined and fuzzy spaces. Thereafter, the organized set of words undergoes first pass of text recognition where we attempt to recognize words from the text and the ones with sufficient confidence are fed to an adaptive classifier to subsequently train the OCR engine while performing text categorization. The entire pipeline is described in the Figure 2 below.

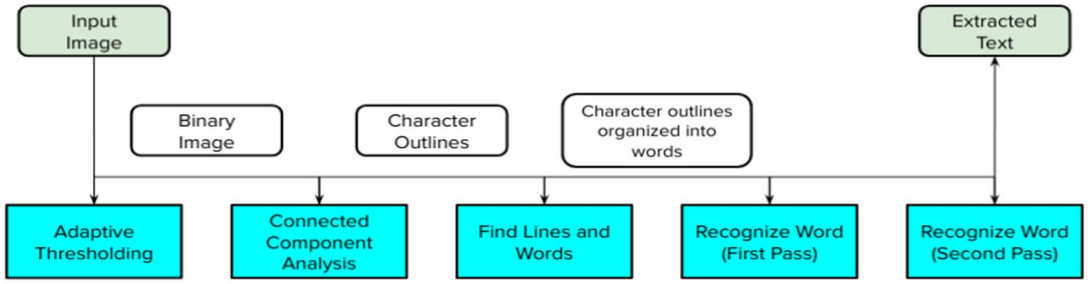


Figure 2. Tesseract OCR Architecture

D. Proposed System

In this work, as mentioned, we have a Django Web UI (Figure 3) where the users can create their account. We have implemented a logout/login functionality so that only the authorized users can access the application. The users simply need to upload the scanned images of the documents of which they need to extract the text. The users can also select the desired output language of the extracted text which is displayed on the application screen and also has the functionality to download the same in the form of a text file (Figure 4).

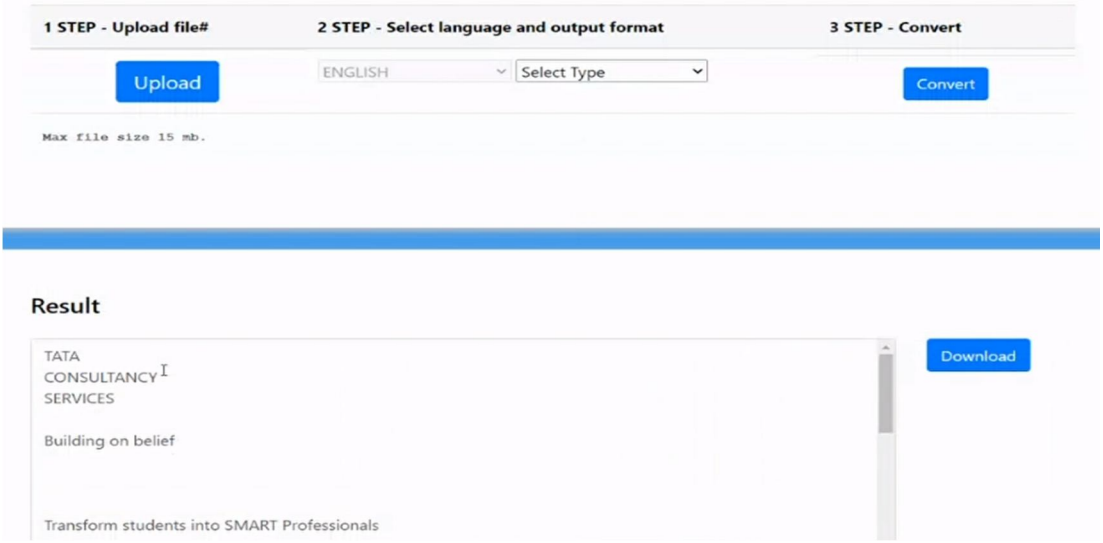


Figure 3. Django Web Application.

©IJRASET: All Rights are Reserved

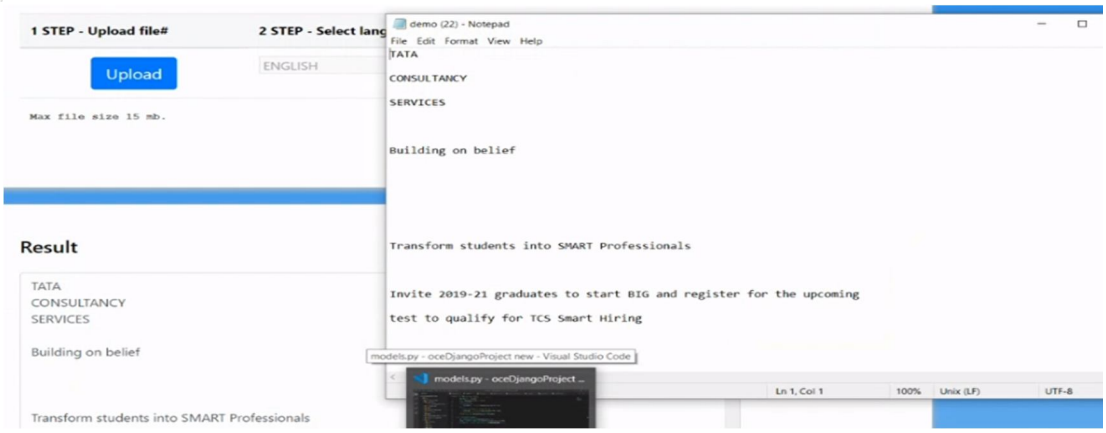


Figure 4. OCR Extracted Text.

E. Software And Hardware Requirements

We have developed this Django project on a 64-bit Windows 10 Machine with 8GB RAM, 10th Gen Intel Core i5 processor for our local deployment but this application can be containerized using Docker. The docker images are completely portable and hence can run on any given environment and platform. Moreover, we can also deploy this web app on cloud so that it can be conveniently accessed from anywhere. We plan to take up the design of the deployment pipelines as part of the future work.

The software libraries used in this project are - Python 3.6 (programming language), Django (Web framework), PyTesseract (Python OCR Engine), Pillow (Python Imaging Library), GoogleTrans (Google Translation Library), MySQL database and HTML, CSS, JS for the minimal UI.

V. RESULTS AND DISCUSSION

In this project, we were successfully able to develop a robust and modular web application for image text extraction and multilingual translation using the Pytesseract based OCR Engine. The system was able to extract text from handwritten and printed documents with high accuracy which further strengthens the fact that OCR based applications can bring a lot of convenience to our daily activities and streamline a lot of workflows that can result in the efficient storage, retrieval, sharing and back up of the information. Although the results look promising, the OCR systems need to be made customizable so that they can be trained so as to efficiently recognize the characters from all sorts of handwritten data sources.

VI. FUTURE WORK

While this project is in itself a standalone application ready to be used, we plan to extend it further through designing a deployment pipeline and make this application truly cross-platform or implement a cloud-based solution. We also aim at conducting a thorough comparative analysis of different OCR Engines and measure their performance across different applications. We have also planned to build a custom OCR model through training so that it can recognize multilingual text. Last but not the least, we would also like to attempt the development of an ensemble model of different OCR engines coupled with state-of-the-art deep learning algorithms such as Long Short Term Memory Networks or LSTMs. The idea behind this ensemble is to exploit the strengths of multiple models while also eliminating their shortcomings.

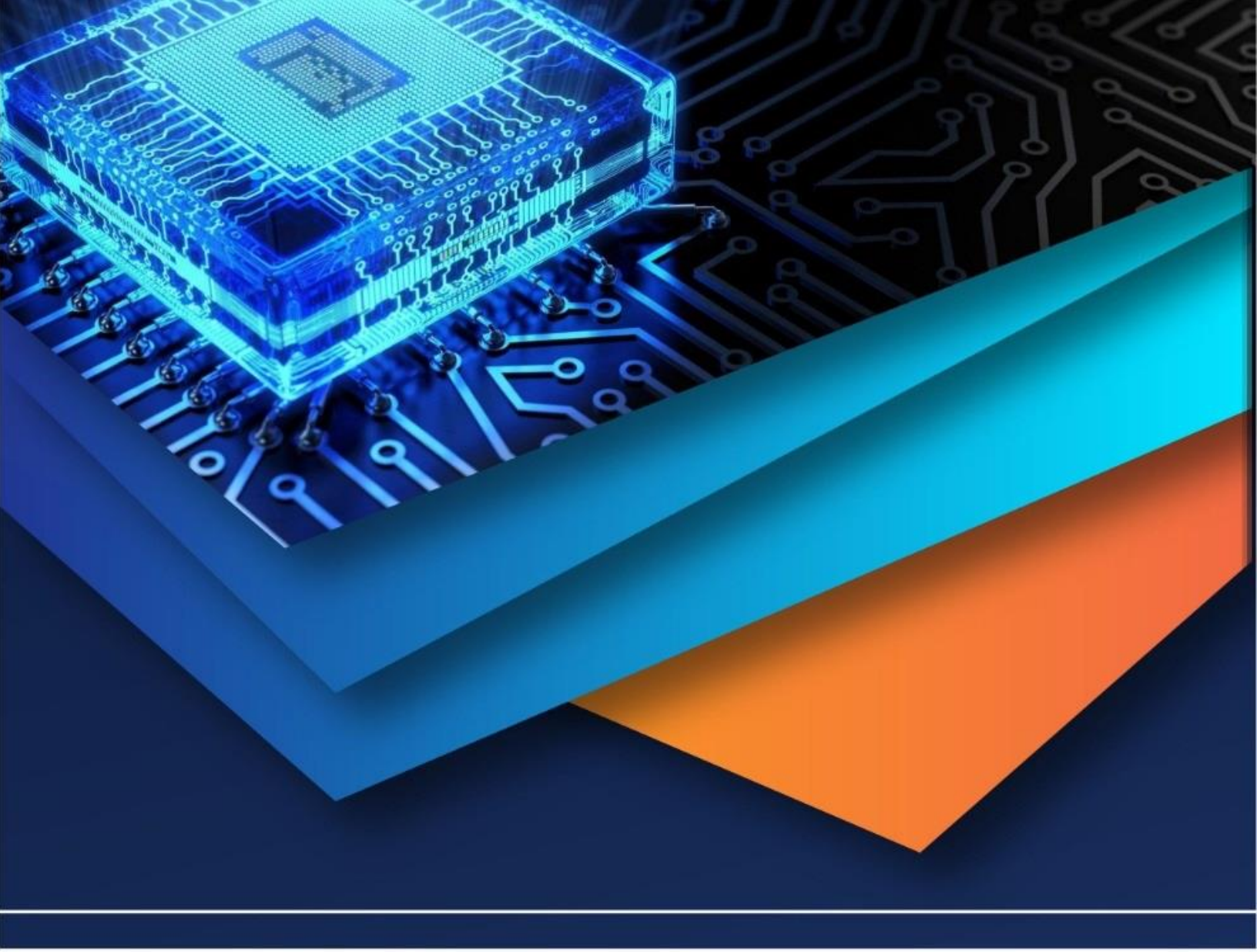
REFERENCES

[1] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/?sh=77b6535560ba
[2] 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.
[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] Pavithra, E., & Kumar, M. A. (2017). Portable Camera Based Text, Product Label and Currency Note Reading from the Hand Held Objects for Blind Person. Asian Journal of Applied Science and Technology (AJAST), 1(3), 66-69.
[5] Rekha, M. (2021). Educational Training For Processing Invoice Of Vendor Identification And Payments Using Python-Tesseract. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(11), 224-228.

©IJRASET: All Rights are Reserved

[6] Tafti, A. P., Baghaie, A., Assefi, M., Arabnia, H. R., Yu, Z., & Peissig, P. (2016, December). OCR as a service: an experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym. In International Symposium on Visual Computing (pp. 735-746). Springer, Cham.
[7] Google docs - http://docs.google.com
[8] Tesseract OCR - https://github.com/tesseract-ocr
[9] Abbyy OCR - https://finereaderonline.com/en-us/Tasks/Create
[10] Transym - http://www.transym.com/
[11] Rasmussen, L. V., Peissig, P. L., McCarty, C. A., & Starren, J. (2012). Development of an optical character recognition pipeline for handwritten form fields from an electronic health record. Journal of the American Medical Informatics Association, 19(e1), e90-e95.
[12] Sharma, P. S., Roy, P. K., Ahmad, N., Ahuja, J., & Kumar, N. (2019, March). Localisation of License Plate and Character Recognition Using Haar Cascade. In 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 971-974). IEEE.
[13] Singh, A., Bacchuwar, K., & Bhasin, A. (2012). A survey of OCR applications. International Journal of Machine Learning and Computing, 2(3), 314.
[14] Google Tesseract OCR - https://opensource.google/projects/tesseract

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)