Open-Source OCR Libraries: A Comprehensive Study for Low Resource Language

Anitha R, Rajeev R R, Meharuniza Nazeem, Navaneeth S

International Centre for Free and Open Source Solutions (ICFOSS)
Government of Kerala, Thiruvananthapuram, Kerala, India
anitha@icfoss.org, rajeev@icfoss.in, meharuniza@icfoss.org, navaneeths@icfoss.org

Abstract

This paper reviews different OCR tools and libraries employed for optical character recognition tasks. Tesseract OCR, an open-source program that supports multiple languages and image formats, is highlighted for its accuracy and adaptability. Python-based libraries like EasyOCR, MMOCR, and PaddleOCR are also mentioned, which provide user-friendly interfaces and trained models for text extraction, detection, and recognition. EasyOCR emphasizes ease of use and simplicity, while MMOCR and PaddleOCR offer comprehensive OCR capabilities and support for a wide range of languages. According to our study, which evaluates various OCR libraries, Tesseract OCR performs remarkably well in terms of accuracy for Indian languages like Malayalam. We focused on five OCR libraries—Tesseract OCR, MMOCR, PaddleOCR, EasyOCR, and Keras OCR—and tested them across several languages, including English, Hindi, Arabic, Tamil, and Malayalam. During our comparison, we found that Tesseract OCR was the only library that supported the Malayalam language. While the other libraries did not support Malayalam, Tesseract OCR performed well across all tested languages, achieving accuracy rates of 92% in English, 93% in Hindi, 78% in Tamil, 74% in Arabic, and 93% in Malayalam.

I. Introduction

Image classification and optical character recognition (OCR) are significant areas of computer vision research. With the continuous evolution of machine learning and deep learning techniques, many researchers are drawn to these fields to develop near-perfect models. Applications of OCR models include document digitization, automated data entry, invoice processing, text extraction from images, and even handwriting recognition. OCR aims to convert printed or handwritten text [1] in images or scanned documents into machine-readable language. While several character recognition models exist for modern languages, analyzing text in ancient manuscripts remains challenging due to the intricate structure of handwritten scripts. Curved lines in printed documents can further complicate text recognition, making character segmentation and recognition difficult. Despite the advancements of OCR systems in major languages like English and French, the OCR system for the Malayalam language [2] is still in its early stages. This paper attempts to explore popular OCR solutions and improve OCR systems for the Malayalam language by leveraging OCR models used for other languages.

Popular open-source OCR libraries like Tesseract OCR, Keras OCR, MMOCR, PaddleOCR, and EasyOCR have significantly advanced OCR technology, enabling the extraction and recognition of text from various sources, including scanned documents and images. Each OCR library offers unique benefits and features to address different OCR requirements. Tesseract OCR is widely used and known for its precision and extensive language support, while Keras OCR simplifies the development of OCR models with its high-level API. MMOCR, based on state-of-the-art deep learning algorithms, excels in complex text detection and recognition tasks. PaddleOCR, built on the PaddlePaddle platform, efficiently handles high-volume, real-time OCR tasks.

EasyOCR is designed for ease of use, making it accessible to developers of all experience levels. The rest of this paper is structured as follows: Section 2 explains the OCR system's primary goal, and Section 3 discusses related work. Section 4 highlights research gaps in the various OCR libraries. Section 5 outlines the methodology and provides key findings from different OCR libraries for languages like English, Hindi, Arabic, Tamil, and Malayalam. Finally, Section 5 summarizes the work and suggests potential future research directions.

II. Objective

The main objective of this study is to conduct a comparative analysis of various OCR libraries for character recognition in multiple languages, including Malayalam, Arabic, Hindi, English, and Tamil. We focus on Tesseract OCR, Keras OCR, PaddleOCR, MMOCR, and EasyOCR libraries, which offer specific strengths and features for different languages.

By assessing the efficiency and accuracy of each OCR library, we aim to identify the most effective method for character recognition in each language. This analysis provides valuable insights into the strengths and weaknesses of each library, helping us choose the one that delivers the highest accuracy for a particular language.

Through this comparative study of OCR libraries for Malayalam, Arabic, Hindi, English, and Tamil, we hope to gain a comprehensive understanding of each library's advantages and limitations. Our findings will contribute to the broader field of OCR research and help determine the best OCR library for specific language needs.

III. Related Works

Dr. V. Geetha et al. [4] When using photographs as input, programs that use optical character recognition (OCR) as their first step play a key role. OCR produces extremely accurate results when used to recognize printed text, with the majority of the data being accurately read. Even if a few fields showed mistakes, the main cause of these inconsistencies was the illegible or damaged quality of the scanned documents. According to our analysis, the best method was the combination of Local Binary Patterns (LBP) and Support Vector Machines (SVM), which had a remarkable accuracy rate of 96.5%. Furthermore, our survey found that data manually entered from forms by knowledgeable people had lesser errors than data read by an OCR system.

Chirag Patel et al. [7], algorithms for text preprocessing and segmentation may have an impact on OCR accuracy. Because of the image's varied size, orientation style, and backdrop complexity it can occasionally be challenging to extract text from it. This paper begins with an overview of the optical character recognition (OCR) technique. Its open-source OCR program is then explained along with its architecture, experiment results, and history. This work is concluded by a comparison of this tool with another commercial OCR program, Transym OCR, using vehicle license plate data as input. We attempted to extract the car number from the number plate using Tesseract and Transym [5], and we compared these tools based on several criteria. As vehicle number plates are the only type of input photographs, Tesseract performs better in these specific images whereas Transym may perform better in other types of images. Both tools have been considered for this particular task because we are interested in extracting the car number from the number plate.

In their work, Thomas Hegghammer et al. [8], benchmarked the performance of Tesseract, Amazon Textract, and Google Document AI on pictures of English and Arabic text. Document AI produced the best outcomes, and the server-based processors (Textract and Document AI) outperformed Tesseract significantly, especially for noisy documents. Arabic has substantially lower accuracy than English. Scholars can find better OCR solutions for their research needs by describing the relative performance of three prominent OCR systems and the differential effects of regularly encountered noise types. For upcoming benchmarking research, the test materials have been kept in the publicly accessible ”Noisy OCR Dataset (NOD)”.

In this research, Ahmad S. Tarawneh et al. [21] explore several machine learning algorithms, such as k-Nearest Neighbours (KNN), Random Forests, and Naive Bayes, to use deep characteristics to solve the invoice classification problem in a novel way. According to experimental findings, using Alexnet deep features in conjunction with the KNN classifier significantly improves classification rates. Principal Component Analysis (PCA) is used to reduce the dimensionality to 183 to handle the huge number of deep features (4096), which improves training efficiency without compromising performance. When evaluated on real invoice data with a variety of obstacles, such as handwriting styles, rotations, backdrops, illuminations, and noise, the suggested methodology displays reasonable classification rates and is validated by cross-validation methods using multiple assessment measures.

IV. Methodology

The fundamental framework or structure of an optical character recognition system is referred to as OCR architecture [6]. It describes the elements and how they work together to give the system the ability to accurately recognize text from photos. Here is a high-level description of a typical OCR architecture, albeit the specific design may vary based on the OCR implementation.

Pre-processing: To optimize text extraction, the input image is enhanced and cleaned during the pre-processing stage. It could involve procedures like normalization, de-skewing, and operations like noise removal and binarization (turning the image black and white).

Text Detection: The task of text detection is to identify the areas or bounding boxes of the image that contain text. To effectively identify text regions, a variety of techniques are used, including edge detection, linked component analysis, and deep learning-based object detection models.

Text Segmentation: It is utilized to separate individual characters or groups of characters after the text regions have been identified. The associated components inside the text regions are segmented and divided into distinct units for further processing.

Feature Extraction: The most important step is feature extraction, which involves extracting pertinent features from the segments of segmented text or characters. While deep learning approaches use convolutional neural networks (CNNs) to automatically learn discriminative features from the characters or regions, traditional methods may involve extracting geometric features like the number of lines, curves, or endpoints.

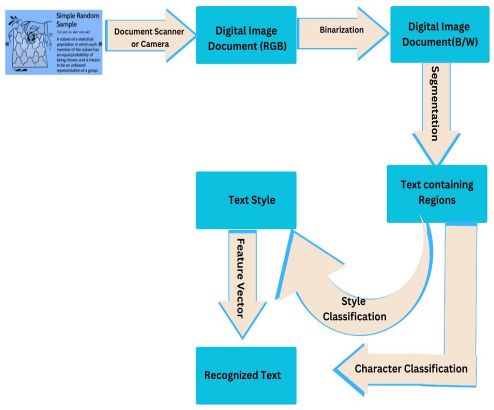


Figure 1. OCR Architecture

Character Recognition: The essential element of OCR is character recognition, which uses the retrieved properties to categorize and identify specific characters. To correctly identify the characters, methods like template matching, statistical models like Hidden Markov Models (HMMs), or contemporary deep learning models like CNNs and recurrent neural networks (RNNs) are used.

Post-processing: To increase the accuracy of the results, post-processing entails improving the recognized text using methods like error correction, language modeling, or contextual analysis. To provide accurate and cohesive output, this phase seeks to address problems like false positives or errors generated during recognition.

Output Generation: According to the demands of the application, the recognized text is then converted into the necessary output format, such as plain text, structured data, or searchable documents.

The OCR system used in this article seeks to identify printed text in an image that is written in Malayalam, English, French, Hindi,s and Arabic. We compare the most accurate model for each language and suggest an OCR system that can be constructed using Tesseract OCR, Easy OCR, Paddle OCR, MMOCR, and Keras OCR. Figure 1 OCR Works demonstrates how they operated.

A. *Dataset*

Images were gathered manually and through web scraping from various online sources. Various languages, including French, Arabic, Malayalam, German, Chinese, English, and Hindi, were included in the dataset we obtained.

B. *Text Extrapolated from an Image*

Using automated data extraction and storage capabilities, OCR technology is a useful business method that saves time, money, and other resources. OCR software retrieves information from scanned documents, camera images, and images in

Copyleft - Everything You Need to Know About This IP License
kashishipr@kashishipr.com
www.kashishipr.com
Copyleft-
EverythingYou Need
to Know About This IP
License
kashishipr@kashishipr.com
www.kashishipr.com

Figure 2. OCR Input images

PDF format as shown above in figure 2. We can access and edit the original content using OCR software, which separates the letters in the image, converts them into words, and then converts the words into sentences. Additionally, it eliminates the need for manual data entry.

The most crucial information can be extracted from large image documents by converting them to plain or rich text. Recognizing the text content is more important than its structure, font style, size, or language when using OCR to create text from images. Although useful, this information is not necessary for text recognition. OCR systems combine hardware and software to convert physical, printed documents into machine-readable text. Using hardware like an optical scanner or a special circuit board, we copy or read the text. The popular OCR software programs Tesseract, EasyOCR, Keras OCR, MMOCR, and Paddle OCR can all be used to extract text from images. Popular OCR frameworks used for text recognition in photos include MMOCR, EasyOCR, PaddleOCR, Keras OCR, and Tesseract OCR. Although their basic methods and applications are different from each other, we will give a brief description of each.

1) MMOCR: Based on the PyTorch architecture, the OpenMMLab team created an open-source OCR toolbox called MMOCR. In addition to text detection, recognition, and layout analysis, it offers a variety of OCR tasks. Modern models and methods such as TextSnake, Proposal Attention Network (PAN), and Scene text Attention Recognition (SAR) are all integrated into MMOCR [10]. This facilitates accurate text recognition in challenging environments.

2) EasyOCR: EasyOCR is an open-source OCR package that offers a simple user interface for text recognition. The Tesseract OCR engine is built on top of it and uses deep-learning models for accurate text extraction. Regions of interest (ROIs) in the input image that include text are first identified by EasyOCR [12] using a text identification technique. While many different algorithms can be used to recognize text, common strategies include those that rely on edges, contours, or neural networks. To create a user-friendly and precise text recognition solution, EasyOCR combines the strength of

Tesseract OCR with deep learning models [11]. To accomplish its objectives, it makes use of text discovery, deep learning-based recognition, language modeling, and post-processing approaches. The library simplifies the OCR's complexities and offers developers an easy-to-use API to incorporate text recognition functionality into their projects.

3) PaddleOCR: It is an open-source OCR framework created by the AI technology company PaddlePaddle [13]. It offers a complete range of tools for text detection and recognition that use trained models. With a focus on effectiveness and usability, PaddleOCR [14] makes use of the PaddlePaddle deep learning technology. Convolutional neural networks and recurrent neural networks (RNN) are both used in the construction of these models.

4) Keras OCR: The Keras OCR [12] library uses the Keras Deep Learning framework to enable OCR features. It strives to offer a straightforward and understandable user interface for text recognition operations. Typically, OCR models based on deep learning architectures like convolutional neural networks (CNN) or recurrent neural networks (RNN) are combined with pre-processing methods like picture binarization and noise removal in Keras OCR. Keras OCR's specific procedures may change depending on the model or technique selected.

5) Tesseract OCR: One of the most well-known and often-used OCR engines is Tesseract OCR [18] [19]. It has excellent text recognition and is an open-source project originally from HP now funded by Google. Tesseract combines Hidden Markov Models (HMMs) and other machine learning algorithms in addition to conventional computer vision approaches to recognize text. The most recent Tesseract version is Tesseract 4, which incorporates deep learning approaches to enhance performance as compared to earlier versions which mostly used conventional methods. The deep learning models used by Tesseract are built using Long Short Term Memory (LSTM) [20]networks.

C. Estimation

We experimented with the models using two metrics: Word Error Rate(WER) [16] and character error Rate(CER). By contrasting the recognized text with a ground truth or reference text, the WER, a metric used to assess the accuracy of an optical character recognition (OCR) system, is calculated. Based on how many word-level errors the OCR system made, the WER is determined [9]. The following equation can be used to get the word error rate:

$$WER = \frac{S\ +\ I\ +\ D}{N}$$

Where: *'D'* represents the number of deletions (words missing from the recognized text compared to the reference text), *'I'* represents the number of insertions (words present in the recognized text but absent from the reference text), *'S'* represents the number of substitutions (words incorrectly recognized), and 'N' represents the total number of words in the reference text.

In Character Error Rate (CER), the concept of Levenshtein distance serves as the foundation for CER calculation [17], which counts the fewest character-level operations necessary to convert the ground truth text (also known as the reference text) into the OCR output.

$$CER = \frac{S\ +\ I\ +\ D}{N}$$

S is the number of "Substitutes".
D stands for "Deletions".
I indicate the number of "Insertions".
N is the number of characters in the ground truth, or reference text.

WER and CER are calculated in these models to find out each model to apply, and how many words or characters can be inserted, deleted, and substituted manually after that comparison.

D. Implementation

A brief explanation of how to compare and assess various OCR libraries, including Tesseract OCR, MMOCR, Keras OCR, Paddle OCR, and EasyOCR, is given in the document. It lists important things including precision, language support, pre-trained models, customization and training choices, integration and API accessibility, speed and performance, community and documentation, pricing and licensing, and availability of integration and training.

The recognition accuracy of five OCR libraries—Tesseract OCR, MM OCR, Paddle OCR, Keras OCR, and Easy OCR—across five languages—English, Hindi, Tamil, Arabic, and Malayalam—is shown above table I. With accuracy rates of 93% and 91%, respectively, the table demonstrates that Tesseract OCR and Paddle OCR perform comparatively better in a variety of languages, especially Malayalam. In comparison, Easy OCR performs the lowest across most languages, particularly Hindi and Arabic, while MM OCR and Keras OCR exhibit reasonable performance.

Libraries	English	Hindi	Tamil	Arabic	Malayalam
Tesseract OCR	92%	93%	78%	74%	93%
MM OCR	89%	62%	72%	73%	73%
Paddle OCR	89%	56%	59%	91%	91%
Keras OCR	72%	68%	70%	71%	71%
Easy OCR	71%	56%	62%	67%	67%

TABLE I
COMPARISON OF OCR LIBRARIES ACROSS DIFFERENT LANGUAGES

Visualizing the differences in accuracy is made easier by the bar graph, which compares the performance of the five OCR libraries in various languages. In Arabic, Paddle OCR performs better than Tesseract OCR, which is superior in Tamil, Hindi, and Malayalam. With the exception of Keras and Easy OCR, all libraries have a high recognition rate for English. This graphic illustrates the large discrepancies in

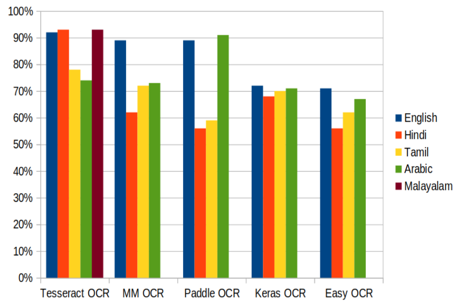
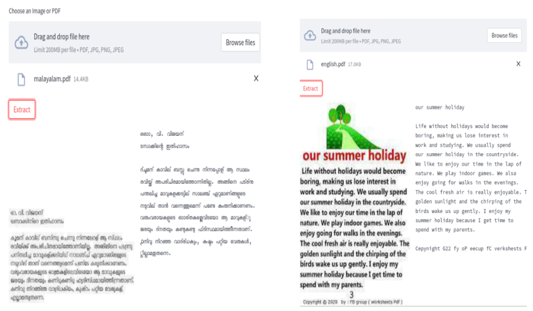


Figure 3. Comparison of OCR Library Performance in different Languages

accuracy for languages such as Tamil and Hindi, particularly with libraries such as Easy OCR.

We can ascertain the library that has the lowest error rate across several languages by carefully examining the WER and CER. Tesseract OCR fared better than the other four OCR libraries in our study, where we carried out this evaluation, in terms of accuracy and precision. Tesseract OCR was found through study to regularly attain greater accuracy rates, making it the best option for our particular OCR requirements. Figure 4 shows the output of Tesseract OCR in Malayalam and English. It is important to keep in mind that the evaluation findings may change based on the dataset, languages, and particular use cases. To choose the best OCR library for your particular needs, it is crucial to conduct a thorough analysis and modify the evaluation procedure to our particular situation.



ഒ. വി. വിജയന്
ഖസാക്കിന്റെ ഇതിഹാസം

കൂമന് കാവില് ബസ്സു ചെന്നു നിന്നപ്പോള് ആ സ്ഥലം രവിയ്ക്ക് അപരിചിതമായിത്തോന്നിയില്ല. അങ്ങിനെ പടര്ന്നു പന്തലിച്ച മാവുകള്ക്കിടയില് നാലഞ്ച് ഏറുമാടങ്ങളുടെ നടുവില് താന് വന്നെത്തുമെന്ന് പണ്ടേ കരുതിക്കാണണം. വരുംവരായകളുടെ ഓര്മ്മകളിലെവിടെയോ ആ മാവുകളുടെ ജരയും ദൈന്യവും കണ്ടുകണ്ടു ഹൃദിസ്ഥമായിത്തീര്ന്നതാണ്. കനിവു നിറഞ്ഞ വാര്ദ്ധക്യം, കുഷ്ഠം പറ്റിയ വേരുകള്, എല്ലാമതുതന്നെ.

ഒാ, വി. വിജയന്
ടസാക്കിന്റെ ഇതിഹാസം

ദ്ച്ചമന് കാവില് ബസ്സു ചെന്നു നിന്നപ്പോള് ആ സ്ഥലം
രവിയ്ക്ക് അപരിചിതമായിത്തോന്നിയില്ല. അങ്ങിനെ പട്ന്ന
പന്തലിച്ച മാവുകളക്കടയില് സാലഞ്ച് ഏറുമാടങ്ങളുടെ
നടുവില് താൻ വന്നെത്തുമെന്ന് പണ്ടേ കംതനിക്കാണണം.
വരുംവരായകളുടെ ഓാര്തകളെവിടയോ ആ മാവുകളു
ജരയും ദ്നതയും കണ്ടുകണ്ടു ഹ്ദിസ്ഥമായിത്തീന്നതാണ്.
ഗ്നിവു നിറഞ്ഞ വാദ്ധക്യം, കഷ്ടം പറ്റിയ വേരുകള്,
്ല്ലാമതുതന്നെ.

our summer holiday

Life without holidays would become boring, making us lose interest in work and studying. We usually spend our summer holiday in the countryside. We like to enjoy our time in the lap of nature. We play indoor games. We also enjoy going for walks in the evenings. The cool fresh air is really enjoyable. The golden sunlight and the chirping of the birds wake us up gently. I enjoy my summer holiday because I get time to spend with my parents.
3
Copyright © 2020 by : FB group (worksheets Pdf)

```
our summer holiday

Life without holidays would become
boring, making us lose interest in
work and studying. We usually spend
our summer holiday in the countryside.
We like to enjoy our time in the lap of
nature. We play indoor games. We alsa
enjoy going for walks in the evenings.
The cool fresh air is really enjoyable. T
golden sunlight and the chirping of the
birds wake us up gently. I enjoy my
summer holiday because I get time to
spend with my parents.

Cepynight G22 fy oP eecup fC verkshests F
```

Figure 4. Output of Malayalam and English Text using Tesseract Library

V. Conclusion and Findings

Tesseract OCR is a well-known and flexible open-source OCR tool praised for its precision and adaptability. It is appropriate for a variety of OCR applications because it supports different languages and image formats. With a user-friendly interface and deep learning models for text extraction and identification, EasyOCR is a Python-based OCR toolkit. A complete OCR toolbox with cutting-edge capabilities like text detection, recognition, and layout analysis is offered by MMOCR, created by OpenMMLab. It offers pre-trained models based on cutting-edge deep-learning architectures and covers a variety of languages. Users can train and enhance models using their datasets and also offers pre-trained models. Keras OCR is a well-known deep-learning package that can be used for OCR jobs. Overall, each OCR technology presented in this context has particular advantages, and the best tool to use relies on the particular requirements, languages, and desired level of customization or simplicity.

Evaluating Tesseract OCR, MMOCR, Paddle OCR, Easy OCR, and Keras OCR libraries for several languages, including English, Hindi, Arabic, Tamil, and Malayalam, can be valuable. Tesseract OCR stands out among these libraries because it specializes in Malayalam OCR and achieves an outstanding 93% accuracy. However, Tesseract OCR routinely surpasses the competing libraries when it comes to additional languages including English, Hindi, Tamil, and Arabic. The key advantage is Tesseract OCR's outstanding capability to handle Malayalam OCR assignments. Users can take advantage of Tesseract OCR's enhanced ability to precisely recognize Malayalam text by utilizing it.

Acknowledgment

We express our sincere gratitude to Shabeerali K P, Swathy A S, Aiswarya J, Sofi A Francis, Nahadia Bint Nazar, D. Nivedhitha for their significant contributions and valuable insights in this project. The collaboration and expertise of these individuals were instrumental in shaping the project's direction and refining the research objectives and methodology.

References

[1] Jamshed Memon; Maira Sami; Rizwan Ahmed Khan Faculty of IT, Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR), IEEE Access (Volume: 8).
[2] Ashitta T Jia; Yahkoob Ayappally; K Syama, Malayalam OCR: N-gram approach using SVM classifier, 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 22-25 August 2013.
[3] Thi-Tuyet-Hai Nguyen; Adam Jatowt; Mickael Coustaty, Deep Statistical Analysis of OCR Errors for Effective Post-OCR Processing,2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL).
[4] Dr. V. Geetha, Ch V V Sudheer, A V Saikumar, Dr. C K Gomathy, Optical Character Recognition(2022), Journal of Engineering Computing and Architecture Optical Character Recognition.
[5] OCR as a Service: An Experimental Evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym, Ahmad P. Tafti, Ahmadreza Baghaie, Mehdi Assefi, Hamid R. Arabnia, Part of the Lecture Notes in Computer Science book series (LNIP, volume 10072).
[6] Vamvakas, G., Gatos, B., Stamatopoulos, N., Perantonis, S.J., A Complete Optical Character Recognition Methodology for Historical Documents, IEEE, The Eighth IAPR International Workshop on Document Analysis Systems (DAS) - Nara, Japan (2008.09.16-2008.09.19)] 2008.
[7] Chirag Patel, Atul Patel, Dharmendra Patel, Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study in 2012, International Journal of Computer Applications (0975 – 8887)
[8] Thomas Hegghammer, OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment in 2021, Journal of Computational Social Science (2022) 5:861–882
[9] An open-source OCR evaluation tool Rafael C. Carrasco, Departamento de Lenguajes y Sistemas Informáticos Universidad de Alicante (Spain).
[10] Gürkan Soykan, Deniz Yuret, Tevfik Metin Sezgin, A Comprehensive Gold Standard and Benchmark for Comics Text Detection and Recognition, Computation and Language (cs.CL); Artificial Intelligence (cs.AI).

[11] D.R. Vedhaviyassh; R. Sudhan; G. Saranya; M. Safa; D. Arun, Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm, 2022 6th International Conference on Electronics, Communication and Aerospace Technology, 01-03 December 2022.

[12] nikoghosyan k.h, OCR engine comparison - tesseract vs easyocr vs keras-ocr, russian-armenian university, armenia in 2022.

[13] Lei Feng; Zongwu Ke; Na Wu, ModelsKG: A Design and Research on Knowledge Graph of Multimodal Curriculum Based on PaddleOCR and DeepKE, 2022 14th International Conference on Advanced Computational Intelligence (ICACI), 15-17 July 2022.

[14] Dan Zhang and Yunjie Li Research and Application of Health Code Recognition Based on Paddle OCR under the Background of Epidemic Prevention and Control, Journal of Artificial Intelligence Practice Vol 6, Issue 1, 2023.

[15] R. Deepa; S. Gayathri; P. Chitra; J. Jeno Jasmine; R. Renuga Devi; A. Thilagavathy, An Enhanced Machine Learning Technique for Text Detection using Keras Sequential model, 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), 02-04 March 2023.

[16] Rohit Saluja, Devaraj Adiga, Parag Chaudhuri,2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR),9 January 2018.

[17] Robin Schaefer, Clemens Neudecker, Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities, and Literature, December 2020.

[18] R. Smith Google, Inc., USA. An Overview of the Tesseract OCR Engine, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September 2007.

[19] Nikita Kotwal, Ashlesh Sheth, Gauri Unnithan, Nehal Kadaganchi, Optical Character Recognition using Tesseract Engine, International Journal of Engineering Research & Technology (IJERT), Vol. 10 Issue 09, September-2021.

[20] Jaskirat Singh; Bharat Bhushan, Real-Time Indian License Plate Detection using Deep Neural Networks and Optical Character Recognition using LSTM Tesseract, 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 18-19 October 2019.

[21] 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Invoice Classification Using Deep Features and Machine Learning Techniques, 20 May 2019, Amman, Jordan.