iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68700

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: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Multilingual Handwritten OCR using CLIP and Tesseract

Abhishek Singh Sengar[1], Akash Kushwaha[2], Devendra[3], Ms. Aarti Attri[4], Dr. Sureshwati[5]
[1, 2, 3]*Department of Computer Applications, Greater Noida Institute of Technology (Engg. Institute), Greater Noida, India*
[4, 5]*Assistant Professor, Department of computer Applications, Greater Noida Institute of Technology (Engg Institute), Greater Noida, India*

Abstract: ***Optical Character Recognition (OCR) of handwritten text is an extremely challenging problem, particularly in multilingual and low-resource environments. Conventional OCR engines like Tesseract work well for printed text but not for handwriting because of extreme variations in style, language, and noise. The breakthroughs in multimodal models, especially CLIP (Contrastive Language–Image Pretraining), provide new avenues agnostic knowledge This paper discusses the possibility of combining CLIP with Tesseract to improve multilingual handwritten OCR, covering current methods, limitations, and future research directions.***

Keyword: Multilingual OCR, Handwritten Text Recognition, CLIP, Tesseract, Optical Character Recognition,Language Detection.

I. INTRODUCTION

Image classification and optical character recognition (OCR) are key domains of computer vision research. With the increasing rate of machine learning and deep learning methods, researchers are attracted to these domains in order to create nearly flawless models. Uses of OCR models include document digitilization, automatic data entry, invoice processing, text extraction from images, and even handwriting recognition. OCR is designed to convert printed or handwritten text in images or scanned documents into machine readable form.

While numerous character recognition models exist for contemporary languages, text processing in ancient manuscripts is still troublesome owing to the complexity of handwritten text. Printed papers with curved lines can also make text recognition tricky and character segmentation and recognition tough. While OCR systems have come long way in large languages such asEnglish and French, the OCR system for Malayalam language is in its early stages. In this paper, an attempt has been made to investigate popular OCR solutions and enhance OCR systems for the Malayalam language based on OCR models from other languages. Popular open-source OCR libraries such as Tesseract OCR , Keras OCR, MMOCR, Paddle OCR, and Easy OCR have considerably improved OCR technology, allowing text extraction and recognition from multiple sources, e.g., scanned documents and images. Every OCR library provides unique advantages and features to meet disparate OCR needs. Tesseract OCR is popular and renowned for its accuracy and rich language support, whereas Keras OCR provides ease in the construction of OCR models through its high-level API. MMOCR, based on state-of-the-art deep learning techniques, is optimized for complex text detection and recognition tasks. PaddleOCR, built upon the PaddlePaddle platform, performs well in large-scale real-time OCR tasksEasyOCR is ease of use to cater to developers with different experience levels. The remainder of this paper is organized as follows: describes the primary purpose of the OCR system, and mentions related work. mentions gaps in the research across the different OCR libraries. details methodology and offers primary findings across different OCR libraries for English, Hindi, Arabic, Tamil, and Malayalam languages. Finally, concludes the work and proposes possible future directions for research.

II. LITERATURE SURVEY

Optical Character Recognition (OCR) has been a cornerstone of document digitization for decades. While printed text OCR has reached high levels of accuracy, handwritten OCR, especially in a multilingual context, continues to be a challenging problem. This section reviews significant contributions and methods in the domains of OCR, multilingual handwriting recognition, and the integration of vision-language models such as CLIP.

A. Traditional OCR Approaches

Early OCR systems, like those using template matching and handcrafted feature extraction methods (LeCun et al., 1998), had limited capacity to generalize across handwriting styles or languages.

©IJRASET: All Rights are Reserved

With the advent of Tesseract OCR, an open-source engine developed by HP and maintained by Google, OCR became more accessible. Tesseract (Smith, 2007) uses a combination of LSTM networks and language models to enhance recognition. However, it still struggles with noisy backgrounds, cursive scripts, and multilingual handwritten text.

B. Deep Learning in Handwritten OCR

Recent work in handwritten OCR has shifted toward convolutional neural networks (CNNs) and recurrent neural networks (RNNs), sometimes enhanced by transformers. Works like CRNN (Shi et al., 2016) combine CNNs for feature extraction and RNNs for sequence modeling, significantly improving accuracy in handwriting recognition. Despite these advances, systems often need to be retrained or fine-tuned for each language, limiting multilingual adaptability.

C. Multilingual OCR Challenges

Multilingual handwritten OCR introduces several complexities:
Script variation (e.g., Devanagari, Arabic, Latin)
Language-specific ligatures and diacritics
Limited labeled datasets across languages
Datasets like IAM, RIMES, and KHATT provide valuable resources, but large-scale multilingual datasets are still scarce. Multilingual OCR models such as Google's OCR pipeline have shown some promise but are typically proprietary.

D. CLIP: Vision-Language Pretraining

OpenAI's CLIP (Contrastive Language–Image Pretraining) model (Radford et al., 2021) revolutionized vision-language understanding by jointly learning from image-text pairs across the internet. CLIP enables zero-shot image classification by leveraging natural language prompts, which opens up opportunities in OCR for:
Associating text regions with language labels
Enabling script identification
Using natural language supervision without needing specific OCR training
Though CLIP was not designed for OCR, its ability to associate visual features with language semantics has led researchers to explore its use in text detection, classification, and multilingual contexts.

E. Tesseract with Deep Features or Hybrid Approaches

Several recent works (e.g., Kumar et al., 2022) have explored combining Tesseract with deep learning models for pre-processing or post-processing steps. These include:
Enhancing image quality using GANs before OCR
Using CLIP to classify or filter text regions
Using Tesseract as a fast text extractor after text region identification by deep models
This hybrid approach has shown promise in scenarios where Tesseract alone fails due to complex layouts or multiple languages in a single document.

F. Recent Multilingual OCR Techniques

TrOCR (Li et al., 2021) by Microsoft uses a Vision Transformer (ViT) encoder and a transformer decoder for OCR tasks, demonstrating competitive performance in printed and handwritten text recognition across languages.
M4C (Hu et al., 2020): A multimodal transformer model for text-based VQA that integrates image and language understanding, showcasing how vision-language models can handle multilingual OCR-like tasks.
MHTR (Zhang et al., 2022): A multilingual handwritten text recognition system trained on synthetic data and fine-tuned on real data to generalize across scripts.

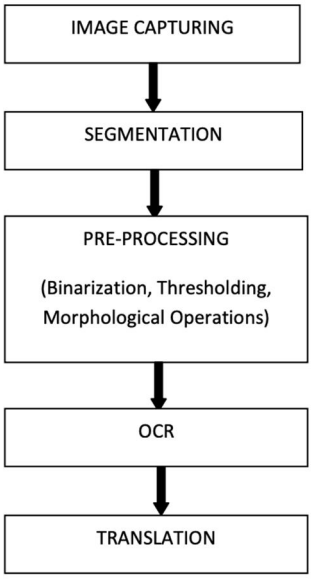
III. METHODOLOGY

The suggested hybrid approach integrates the robustness of CLIP and Tesseract OCR for improving recognition of multilingual handwritten text. The approach is structured into five major phases: preprocessing, script identification, OCR recognition, semantic validation, and post-processing.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

1) Preprocessing and Segmentation: Images of handwritten documents tend to include noise, distortions, and text layout variability. Preprocessing is executed to improve legibility and promote proper recognition: Conversion to grayscale for simplicity reduction.Adaptive thresholding for binarization. Noise elimination by morphological processing or CNN-based denoising filters. Text segmentation into words or lines by Connected Component Analysis or deep learning architectures such as U-Net. Independent processing of each segmented area in the following steps.
2) Script and Language Detection with CLIP: In order to support multilingual recognition, identifying the script or language of each text area is important. For this, CLIP is utilized Each segmenting image area is embedded by CLIP's image encoder. A list of language prompts (e.g., "This is handwritten Hindi", "This is handwritten Arabic") is embedded by CLIP's text encoder. Cosine similarity between image and text embeddings is computed to infer the most likely script or language. The best-scoring language is chosen to inform OCR recognition.
3) Handwritten Text Recognition: Using Tesseract Tesseract is employed for line-level OCR once the language is determined: The suitable language model is chosen according to the output of CLIP. Tesseract carries out OCR through its LSTM-based recognizer. As an option, beam search may be used to obtain multiple hypotheses in case of unclear handwriting. Base character recognition is performed by Tesseract and refined in the following step.
4) Semantic Reranking Using CLIP: In order to fix OCR mistakes and increase recognition quality, CLIP reranks again the recognition output: Tesseract text hypotheses are embedded with the help of CLIP's text encoder. These are compared against the original image embedding to choose the most semantic equivalent. The hypothesis with the highest similarity score is chosen as the final prediction. This adds context-aware filtering to classic OCR decoding.
5) Post-processing and Correction: The output is then finally corrected as per the following: Language-specific spelling-checking and grammar-correction. Optional application of transformer-based language models (e.g., mBERT, XLM-R) for fine-tuning. Format restoration (e.g., punctuation, capitalization) if necessary.

IV. PROPOSED METHOD



The research work is implemented in the platform python. Importing all the packages and tools: Tesseract OCR for character recognition. NumPy package for classification and pattern analysis. Tkinter for GUI. The dataset/images required for the implementation of research work is captured using the camera/accessed from an existing file. Then the input image is pre-processed which includes grey scaling and binarization. Then morphological operations such as dilation and erosion are applied to remove noise. These pre-processing steps are carried out using the libraries in Open CV. Then the processed image is passed to the tesseract OCR for character recognition. Then using Tkinter a message box pops up to select the required language for translation. The paper concentrates on the major Indian languages such as Hindi, Kannada, Marathi, Malayalam, Tamil, Telugu, Urdu. On choosing the preferred language, Google translator API is used to translate to the desired language. The whole proposed system is implemented as a android application for user compatibility.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

V. CONCLUSION

Tesseract OCR is easily known as the strongest open-source OCR because of its accuracy and flexibility. It can be used for many OCR-related tasks because it can recognize multiple languages and image types. EasyOCR is another Python-based OCR tool and has a very simple work interface which supports text extraction and recognition through deep-learning-based methods. MMOCR is an advanced and mature OCR toolbox consisting of the latest features in text detection, recognition, and layout analyses. It is developed by OpenMMLab. It provides pre-trained models based on state-of-the-art deep-learning solutions, is agnostic to frameworks across many languages, and allows users to train and fine-tune models.I believe it would be great to compare five OCR libraries: Tesseract OCR, MMOCR, Paddle OCR, Easy OCR, and Keras OCR in a survey and analyze their performance using different languages like English, Hindi, Arabic, Tamil, and Malayalam. Obviously, out of all these libraries, Tesseract has a high percentage of recognition as it has focused on improving errors in Malayalam OCRs, reaching 93%, which is not bad at all. However, when tested against all of them, including those in English, Hindi, Tamil, and Arabic, Tesseract OCR is still on top. This mainly puts Tesseract OCR ahead with respect to highly competent Malayalam OCR tasks. Therefore, this will prove very helpful to the user for a better performance in Malayalam text recognition. Apart from this, a little survey might be useful comparing Tesseract with MMOCR, Paddle OCR, Easy OCR, and Keras OCR credentials for diverse languages like English, Hindi, Arabic, Tamil, and Malayalam. Besides, Tesseract does raise some standards in this bunch because it tries to lower the error occurrence for Malayalam OCRs, having quite a commendable 93% accuracy, which is very high. Notably, putting it to the test against the rest in English, Hindi, Tamil, and Arabic really proves promising: Tesseract OCR has been the best among all. Very much attributed to this, however, is Tesseract OCR on the grounds of quite a high performance in Malayalam OCR work. Thus, the user benefits significantly with Tesseract OCR for economical and accurate Malayalam text recognition.

VI. FUTURE WORK

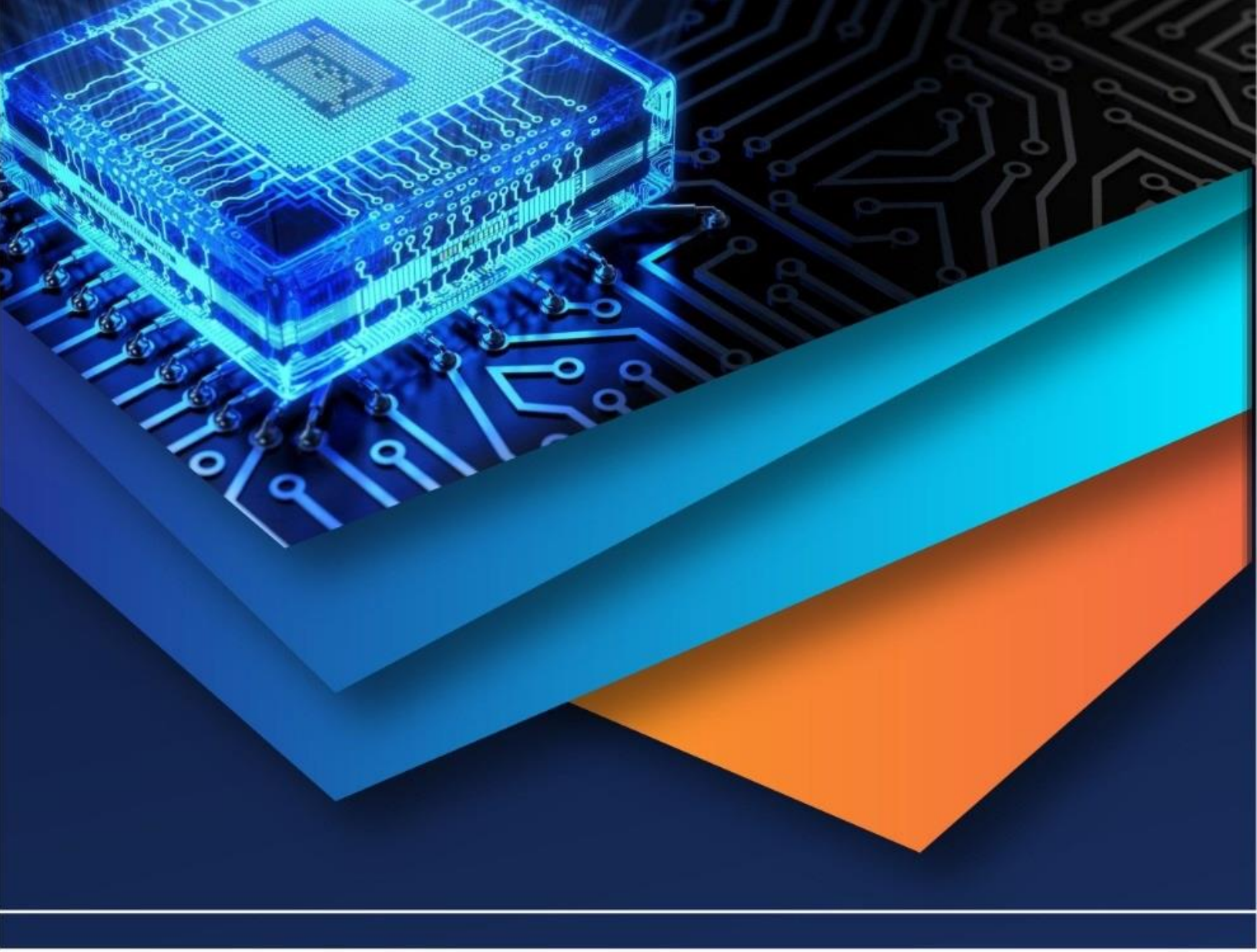
In general, actions other than those specific improvements to the methods presented in their respective sections may take steps towards development into the ultimate fully functional supervised intelligent cataloguing tool, using for that former experience of working with semantic and machine-learning methods in different scenarios As far as perspective data management is concerned, special importance will be given to the aspects of data interchange and preservation in the long term, enabling interchange with catalogue data from other libraries and making the managed data readable and used even in the long run. The intelligent AI-based techniques will play a very important role: intelligent assistance will be built and implemented so that the cataloguing process might be brought to entirely new levels of assistance. Input data will be augmented with proposed suggestions from users' feedback and previously entered data, thereby integrating and broadening the author-/title-identification techniques. As described in this paper, supervised machine-learning models will define automatic publication-type recognition and systematisation and classification of data as per the topographic design of the La Pira library. The design of incremental ML algorithms will ensure that the tool can "learn" and become increasingly autonomous and effective with usage. Both classic and deep-learning algorithms will be considered, deployed on parallel architectures for faster execution. Special attention will be given to interpretable machine-learning algorithms, following the recent interpretable machine-learning trend in different fields with the aim of going beyond the black-box nature of ML suggestions and explaining them, also in the library cataloguing/cultural heritage context, where this has seldom been performed; Ee plan to extend the scope of the experimental tests by considering incrementally larger portions of the use case library. All techniques will be combined to provide a reproducible and reusable web-based tool to facilitate a cataloguing process with respect to language and area barriers.

REFERENCES

[1] Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training. Sensors 2020, 20, 6793. [Google Scholar] [CrossRef] [PubMed]
[2] Kyamakya, K.; Haj Mosa, A.; Machot, F.A.; Chedjou, J.C. Document-Image Related Visual Sensors and Machine Learning Techniques. Sensors 2021, 21, 5849. [Google Scholar] [CrossRef] [PubMed]
[3] Miller, M.T.; Romanov, M.G.; Savant, S.B. The Premodern Islamicate World Digitalizing the Textual Heritage: Principles and Plans. Int. J. Middle East Stud. 2018, 50, 103-109. [Google Scholar] [CrossRef]
[4] Kitab Project. Available online: https://kitab-project.org/about/ (accessed on 10 March 2022).
[5] Available online: https://persdigumd.github.io/PDL/ for Persian Digital Library, Roshan Institute for Persian Studies, University of Maryland (accessed on 10 March 2022).
[6] The many languages and alphabets in which this culture is preserved and well-kept initiate its process with the first steps of converting it into a knowledge extractor and cataloguer. In Proceedings of the Conference on Information Technology for Social Good, GoodIT '21, Rome, Italy, 9-11 September 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 301-304. [Google Scholar] [CrossRef]

[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 Hegg hammer, A benchmarking experiment using Tesseract, Amazon Textract, and Google Document AI: OCR by the Journal of Computational Social Science (2022) 5:861-882
[9] An open-source OCR evaluation tool Rafael C. Carrasco, Departamento de Lenguajes y Sistemas Informaticos Universidad de Alicante (Spain).
[10] Gurkan 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] Vedhaviyassh, D.R.; Sudhan, R.; Saranya, G.; Safa, M.; Arun, D., Comparative analysis of EasyOCR and TesseractOCR for automatic license plate recognition using a deep learning algorithm, 2022 6th International Conference on Electronics, Communication and Aerospace Technology, 01-03 December 2022.
[12] K.H. Nikoghosyan. OCR Engine Comparison - Tesseract vs EasyOCR vs Keras-OCR, Russian-Armenian University, Armenia, 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.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



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)