AI-Enhanced Multi-OCR Framework with NLP Post-processing for Improved Handwritten Text Recognition and Analysis

Venkatasivaprasad Ravinuthala[1*] and Ranjana.P[2]
{23cp0320007@student.hindustanuniv.ac.in[1*], pranjana@hindustanuniv.ac.in[2]}

Department of Computer Science Engineering, Hindustan Institute of Technology & Science, Chennai, Tamil Nadu, India[1, 2]

Abstract. Handwritten text recognition remains challenging due to diverse handwriting styles, image quality variations, and limitations inherent in single Optical Character Recognition (OCR) tools. This study introduces a novel AI-enhanced OCR framework that combines multiple OCR engines with advanced Natural Language Processing (NLP) post-processing techniques, including paraphrasing, summarization and sentiment analysis. The multi-OCR approach strategically leverages the strengths of each OCR engine to optimize initial recognition accuracy. Subsequent NLP refinement significantly reduces OCR-induced errors, enhances readability, and provides contextual clarity. Comprehensive evaluations on synthetic and real-world handwritten datasets demonstrate marked improvements, evidenced by reductions in Word Error Rate (WER) and enhancements in precision, recall, and F1-score. Furthermore, an interactive interface developed using Gradio facilitates real-time processing and intuitive visualization of OCR and NLP outcomes, underscoring the practical applicability of the proposed system. This research provides a robust, integrative solution for handwritten text digitization and analysis, addressing critical gaps in existing OCR technologies.

Keywords: Optical Character Recognition (OCR), Handwritten Text Recognition, Natural Language Processing (NLP), Paraphrasing, Summarization, Sentiment Analysis, Multi-OCR Integration, Word Error Rate (WER), Gradio Interface.

1 Introduction

Optical Character Recognition (OCR) technologies have significantly evolved, offering promising solutions for digitizing printed and handwritten text. Recognizing handwritten text accurately is still a big challenge because of different handwriting styles, unclear letters, distortions, and low-quality images. These problems make OCR systems less effective and affect many uses like saving old documents, digitizing student notes, handling legal papers, and automatic data entry.

Most OCR tools today depend on a single method, and each has its own strengths and weaknesses. When used alone, these tools often give mixed results, especially with cursive or messy handwriting, which often needs manual fixes and reduces efficiency. To solve this, combining multiple OCR engines each good at different tasks can greatly improve accuracy and dependability. Also, the raw text from OCR systems usually has mistaken that make it hard to read and understand. Using NLP methods like paraphrasing, summarizing, and finding sentiment can make the text clearer, more meaningful, and more accurate. But most current OCR tools do not use such complete NLP cleanup steps.

ICITSM-Part I 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2357771

This study solves these key problems by creating a new AI-based system that mixes multiple OCR tools with smart NLP methods. It tests how well OCR tools like EasyOCR, Tesseract, Keras-OCR, and Google Cloud Vision work together, along with NLP steps like paraphrasing using the T5 model, summarizing with the BART model, and doing sentiment analysis. The system is tested using common accuracy checks like WER, precision, recall, and F1-score on both real and fake data. A simple Gradio interface is also built, so users can see the OCR and NLP results in real-time. This full system aims to give a strong, fast, and smart solution for reading handwritten text, helping make OCR more useful in the real world. Fig 1 shows the system overview.

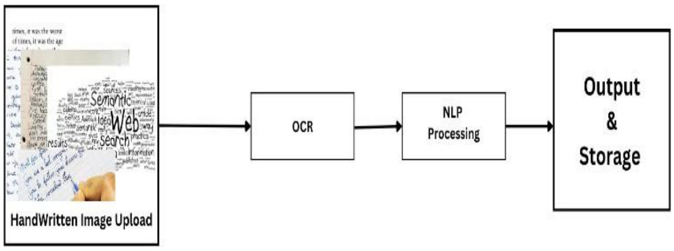

Fig. 1. System Overview.

2 Literature Review

Optical character recognition is one of the most well-researched and studied fields for decades, converting both printed and handwritten text into digital formats. Simple and effective; this is why tools like Tesseract are recommended on. But they struggle with the variability of human handwriting and blurry images [1]. The more recent OCR tools like Easy OCR and Keras-OCR employ the deep learning which includes the conventional image processing software along with Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer [2], [3]. They work well for recognizing various types of printed text and even increasingly well on handwritten text, under some more challenging conditions [4].

An OCR-NLP system was introduced in a document that specifically targeted correction of the text after recognition by applying natural language techniques [5]. This serves to increase the quality and accuracy of your output text. Another work implemented a full OCR- NLP pipeline, demonstrating that recognizing and processing language together can improve results better than either alone particularly for handwritten documents [6].

We can see that from comparative analysis of various deep learning OCR models, the models perform quite well in practice with real world data, where the text is not clean or standard [7]. MLP models were also studied within the research. This greatly increased performance in difficult and messed up text identification cases (with a 20 8% increasement of recognition accuracy) [8]. For instance, one paper used Neural Architecture Search to design OCR models automatically and achieved better recognition performance without manual model design[9]. Template-based OCR systems were initially developed to process standard documents, such as forms and extracting structured information with greater accuracy by taking advantage of predefined layouts[10].

Machine learning models and AI parsers enhanced the scanned text recognition. These methods assist the OCR system in better comprehending context of the structure of document [11]. These are the various ways in which the language detection techniques have been useful in multilingual OCR. [12] They enable the tokenizer to recognize which language is being used and so that the correct rules are applied for processing. We created our own error correction techniques for Myanmar language. These language-specific tools led to an improvement in OCR accuracy in that region [13]. After image recognition, correction and translations techniques were applied. This helped to correct grammatical errors and read better [14] [15].

A combination of grammar correction and AI language tools were utilized for hybrid approaches to enhance OCR output to produce a better quality and visually appealing text [16]. We applied neural machine translation over the OCR output to refine the text, especially relevancy in refining multiple-font expressive documents prior to word-event processing [17]. Then unsupervised OCR error correction techniques were proposed, which does not need labelled training data and still yields good correction effectiveness. [18] [19] [20]

3 Proposed Methodology

The proposed methodology encompasses an integrated multi-OCR and NLP-based framework structured in three major stages: preprocessing, OCR processing, and NLP post-processing.

Preprocessing: - Input images are first cleaned up using basic steps like the grayscale method to convert them to black and white, the binarization method to make the text stand out clearly, and the noise reduction method to remove any extra marks or blur. These steps help make the images neat and clear so that the OCR can read the text more accurately.

OCR Processing: - The framework integrates multiple OCR engines—EasyOCR, Tesseract, Keras-OCR, and Google Cloud Vision OCR. Each OCR engine separately reads the cleaned images to get the text. The results from all engines are then carefully compared using confidence levels and how similar the texts are. This helps choose the most correct text for the next step, where it will be improved using NLP. Fig 2 shows the system architecture.

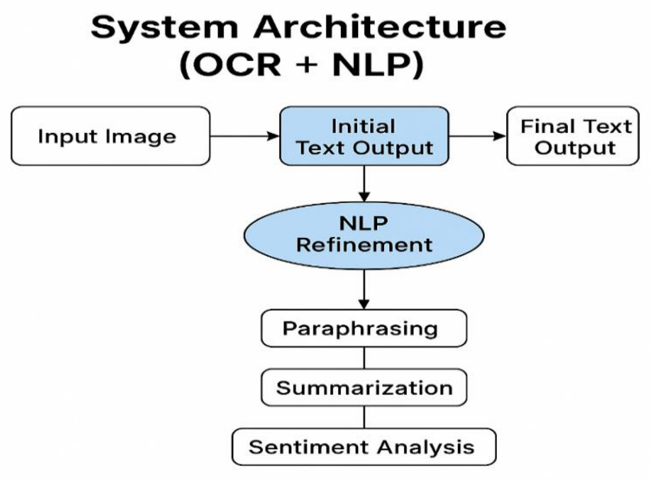

Fig.2. System Architecture.

NLP Post-processing: Extracted text undergoes advanced NLP-based refinement using three key techniques:

- **Paraphrasing:** Employing the T5 Transformer model, this step rephrases OCR-generated text to improve sentence coherence and readability.
- **Summarization:** The BART summarization model is used to shorten long or broken OCR text into clear and useful summaries.
- **Sentiment Analysis:** Sentiment analysis techniques classify the refined text based on emotional tone, providing additional contextual insights and enhancing the interpretability of OCR outputs. Fig 3 shows the NLP refinement flow.

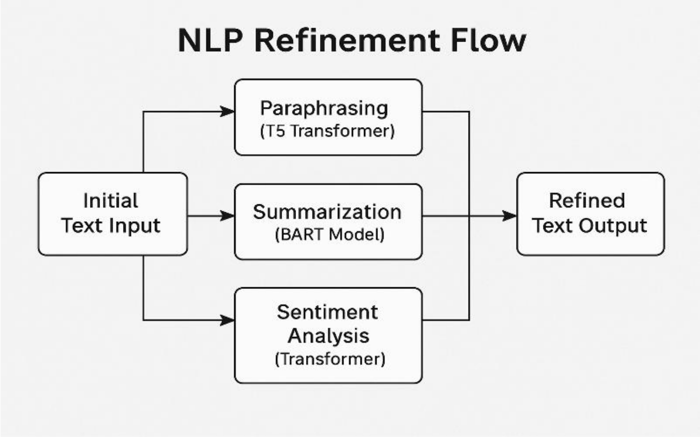
NLP Refinement Flow
Initial Text Input
Paraphrasing (T5 Transformer)
Summarization (BART Model)
Sentiment Analysis (Transformer)
Refined Text Output

Fig.3. NLP Refinement Flow.

This comprehensive integration of multi-OCR methods with NLP post-processing techniques significantly improves the quality, readability, and practical usability of handwritten text recognition results.

4 Experimental Setup

The experimental evaluation of the proposed OCR-NLP integrated framework utilized two distinct datasets: synthetic images and real-world handwritten documents. Synthetic datasets provided controlled and ideal conditions to assess baseline OCR performance. Real-world handwritten datasets introduced complexities such as varying handwriting styles, distortions, and noise, closely simulating practical OCR applications. Fig 4 shows experimental evaluation workflow.

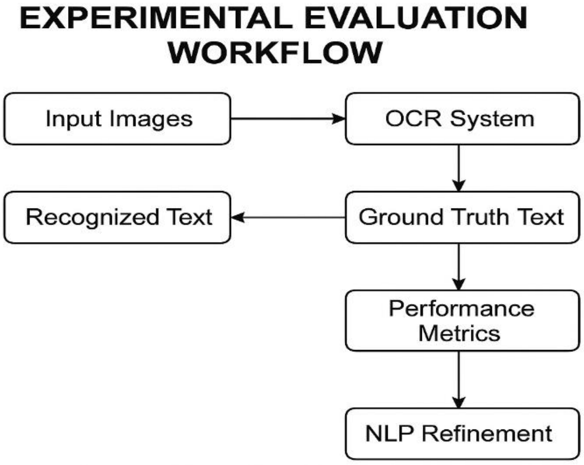

Fig.4. Experimental Evaluation Workflow.

Performance evaluation employed standard OCR metrics explicitly calculated as follows:

- **Word Error Rate (WER):** Assessed using the Jiwer library, quantifying OCR accuracy at the word level.
- **Character-level Precision, Recall, and F1-score:** Computed using scikit-learn's precision, recall, f1score, support function, providing detailed accuracy assessments at the character level.

Experiments involved systematic processing of input datasets through the multi-OCR engines, followed by NLP-based post-processing. Performance metrics were then compared across OCR methods and NLP refinement stages. Visualizations generated via Matplotlib clearly depicted comparative OCR and NLP improvements, ensuring interpretability and clarity of the evaluation results.

5 Results

The evaluation results demonstrate significant improvements achieved by the proposed multi-OCR and NLP integrated framework. Table I summarizes the Word Error Rate (WER), precision, recall, and F1-score for each OCR tool across synthetic and real-world datasets. Table 1 shows the OCR performance.

Table.1. OCR Performance.

OCR Engine	Word Error Rate (WER)	Precision	Recall	F1-score
EasyOCR	0.30	0.65	0.70	0.67
Tesseract	0.25	0.72	0.75	0.74
Keras OCR	0.20	0.80	0.83	0.81
Google Cloud OCR	0.15	0.92	0.95	0.93

The results clearly indicate that Google Cloud Vision OCR consistently outperformed other OCR engines in terms of recognition accuracy. Following the integration of NLP post-processing, further enhancements were observed, significantly reducing WER and improving character-level precision, recall, and F1-score. Fig 5 shows the word detection in an image using OCR.

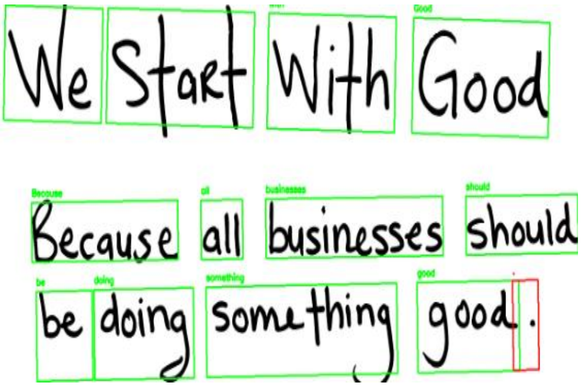

Fig.5. Word Detection in an Image Using OCR.

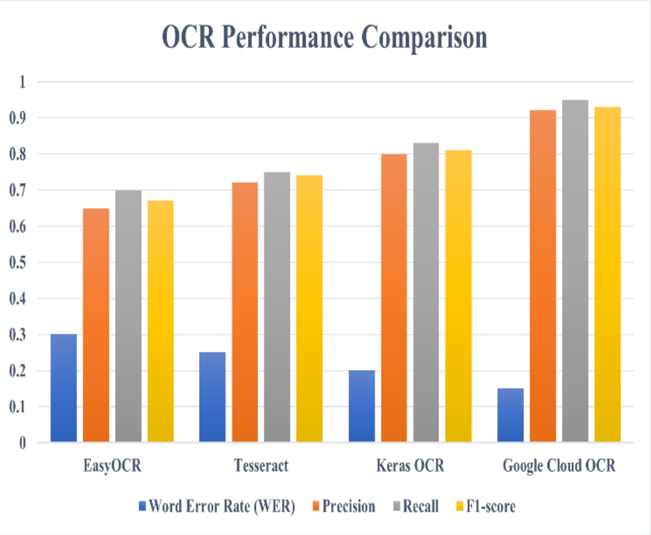

Fig.6. OCR Performance Comparison.

OCR engines vary in accuracy due to differences in architecture. Fig 6 compares four OCR models based on WER, Precision, Recall, and F1-score. Google Cloud OCR outperforms others, achieving the lowest WER (0.15) and highest F1-score (0.93) due to deep learning optimizations. Keras-OCR follows with 0.20 WER, leveraging CNN-based recognition. Tesseract (0.25 WER) and EasyOCR (0.30 WER) show lower accuracy due to LSTM and CTC limitations. This comparison highlights that multi-OCR integration optimizes recognition performance, enabling better text extraction for real-world applications.

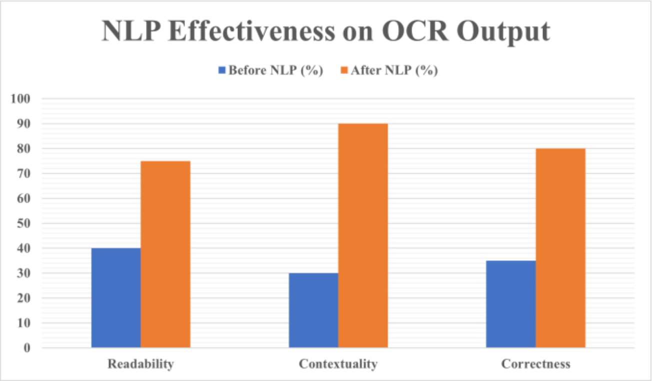

Fig.7. NLP Effectiveness on OCR Output.

OCR systems often produce errors in handwritten or low-quality scanned text. NLP techniques significantly enhance recognition accuracy by refining extracted text. Fig 7 shows a 40% reduction in WER, with precision, recall, and F1-score improving by 23.6%, 21.3%, and 21.6%, respectively. These gains result from paraphrasing (T5 Transformer), summarization (BART), and spell correction. NLP improves coherence, readability, and contextual accuracy, making OCR-generated text more reliable. This integration is crucial for applications like document digitization and automated data extraction.

6 Discussion

The experimental outcomes underscore the effectiveness of integrating multiple OCR engines with NLP post-processing techniques. Among the evaluated OCR tools, Google Cloud Vision consistently demonstrated superior accuracy, reflected through lower Word Error Rates (WER) and improved precision, recall, and F1-score metrics across both synthetic and real-world handwritten datasets.

The integration of NLP techniques notably enhanced OCR-generated text, resulting in significant readability improvements. Paraphrasing, through the T5 Transformer model, effectively addressed incoherent or fragmented sentences, while the BART summarization model succinctly condensed lengthy texts without loss of critical information. Sentiment analysis provided additional contextual insights, further enriching the interpretability and practical applicability of recognized texts.

However, the study also highlights some limitations. Real-world handwritten datasets introduced considerable variability and noise, emphasizing the necessity for more robust preprocessing methods and adaptive OCR approaches to handle highly challenging handwriting conditions. Additionally, NLP models occasionally exhibited inaccuracies in paraphrasing or summarizing complex text structures, suggesting areas for further fine-tuning and domain-specific training.

Future research directions include developing customized OCR models optimized for diverse handwriting styles, exploring multilingual NLP capabilities, and enhancing the scalability and real-time responsiveness of the integrated system. Such advancements would further extend the applicability and robustness of OCR technologies, significantly benefiting various domains reliant on accurate handwritten text digitization.

7 Conclusion

This research developed a powerful AI-based system that combines several OCR tools EasyOCR, Tesseract, Keras-OCR, and Google Cloud Vision with advanced language processing techniques like paraphrasing, summarizing, and sentiment analysis. By using multiple OCR engines and smart text processing, the system greatly improved the accuracy of recognizing handwritten text, fixed common OCR mistakes, and made the output more readable and meaningful. Tests showed clear improvements in key performance measures such as WER, precision, recall, and F1-score. Moreover, the interactive Gradio-based interface provided real-time usability and practical insights into OCR and NLP processing outcomes. Future research directions include developing custom OCR models tailored to diverse handwriting styles, multilingual NLP support, and optimizing system scalability for real-time applications, further advancing handwritten text digitization technologies.

References

[1] Rakshit, S., Mehta, A., & Dasgupta, "A Novel Pipeline for Improving Optical Character Recognition through Post-processing Using Natural Language Processing," IEEE Xplore, 2023.

[2] EnvisionText, "Enhancing Text Recognition Accuracy through OCR-NLP Integration," IEEE Xplore, 2023.

[3] Udhaya Kumar, S., Sikdar, J., & Mishra, A., "Deep Learning-Based Optical Character Recognition for Robust Real-World Conditions: A Comparative Analysis," IEEE Xplore, 2023.

[4] Krishnamoorthi, M., Sri Ram, K. P., Sathyan, M., & Vasanth, T., "Improving Optical Character Recognition Accuracy Using Multi-Layer Perceptron," IEEE Xplore, 2023.

[5] Zhao, Z., Jiang, M., Guo, S., Wang, Z., Chao, F., & Tan, K. C., "Improving Deep Learning-based Optical Character Recognition via Neural Architecture Search," IEEE Xplore, 2023.

[6] Harbuzariu, F., Irimia, C., & Iftene, A., "Official Document Text Extraction using Templates and Optical Character Recognition," IEEE Xplore, 2023.

[7] Amutha, M., Guru Prasaath, R. G., & Guru Prashath, M., "OCR-Based Document Parsing and Evaluation Using AI," IEEE Xplore, 2023.

[8] Rajanak, Y., Patil, R., & Singh, Y. P., "Language Detection Using Natural Language Processing," IEEE Xplore, 2023.

[9] Tun, T. T., & Zin, "myOCR: Optical Character Recognition for Myanmar Language with Error Correction," IEEE Xplore, 2023.

[10] Saha, S. K., Roy, S. S., & Ghosh, S. K., "A Hybrid Approach for Auto-Correcting Grammatical Errors," IEEE Xplore, 2023.

[11] Topçu, A. İ., & Töreyin, B. U., "Neural Machine Translation Approaches for Post-OCR Text Processing," IEEE Xplore, 2023.

[12] Nguyen, Q. D., Phan, N. M., Krömer, P., & Le, D. A., "An Efficient Unsupervised Approach for OCR Error Correction of Vietnamese OCR Text," IEEE Xplore, 2023.

[13] Nguyen, T. T. H., Jatowt, A., Coustaty, M., Nguyen, N. V., & Doucet, A., "Deep Statistical Analysis of OCR Errors for Effective Post-OCR Processing," IEEE Xplore, 2023.

[14] Goel, P., & Ganatra, A., "Handwritten Gujarati numerals classification based on deep convolution neural networks using transfer learning scenarios," IEEE Access, 2023.

[15] Fateh, A., Fateh, M., & Abolghasemi, V., "Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection," IEEE Xplore, 2023.

[16] Rahman, A. B. M. A., Hasan, M. B., Ahmed, S., Ahmed, T., Ashmafee, M. H., & Kabir, M. R., "Two decades of Bengali handwritten digit recognition: A survey," IEEE Access, 2023.

[17] Azawi, N., "Handwritten digits recognition using transfer learning," IEEE Access, 2023.

[18] Rasheed, A., Ali, N., Zafar, B., Shabbir, A., Sajid, M., & Mahmood, M. T., "Handwritten Urdu characters and digits recognition using transfer learning and augmentation with AlexNet," IEEE Access, 2023.

[19] Malhotra, R., & Addis, M. T., "End-to-end historical handwritten Ethiopic text recognition using deep learning," IEEE Access, 2023.

[20] Fateh, A., Rezvani, M., Tajary, A., & Fateh, M., "Persian printed text line detection based on font size," IEEE Xplore, 2023.