ijamrsd.com ISSN: 2583-6404
Journal of Advanced Multidisciplinary Research Studies and Development
Peer Reviewed Referred Journal Bi-monthly Open Access

PERFORMANCE AND ACCURACY ANALYSIS OF MULTILINGUAL TEXT RECOGNITION USING OCR

Pranav Rajendra Patil[1], Monali Y. Khachane[2]
[1] *Research Scholar, KCES's M. J. College, Jalgaon, Maharashtra, India.*
Email: sonpandit@gmail.com
[2] *Asst. Professor, Dept. of Computer Science, Dr. Annasaheb G.D. Bendale Mahila Mahavidhyalaya, Jalgaon, Maharashtra, India.*

Abstract
Multilingual text recognition is essential for applications such as document digitization, assistive technologies, and automated translation. However, recognizing text in diverse scripts, particularly English, Hindi and Marathi, presents challenges due to script complexity, character structures and writing conventions. Traditional OCR systems, such as Tesseract and EasyOCR, struggle with Devanagari-based scripts due to their intricate ligatures and diacritical marks. This study evaluates the performance of existing OCR methods and proposes a CNN-Transformer-based model for improved multilingual text recognition. Experiments conducted on datasets containing printed and handwritten text revealed that Tesseract OCR achieved 85% accuracy for English but underperformed for Hindi (72%) and Marathi (68%). EasyOCR demonstrated higher accuracy, averaging 92% for English and 80% for Hindi and Marathi, but struggled with handwritten and skewed text. The proposed model significantly outperformed traditional methods, achieving 96% accuracy for English and over 88% for Hindi and Marathi. Its robustness to noise, font variations, and distortions highlights the effectiveness of deep learning techniques in multilingual OCR. This research underscores the importance of script-specific preprocessing and dataset augmentation to enhance OCR performance, particularly for underrepresented scripts like Hindi and Marathi.
Keywords: Multilingual OCR, Text Recognition, Deep Learning, Devanagari Script.

► *Corresponding Author: Pranav Rajendra Patil*

1. Introduction

1.1 Background

In today's interconnected world, multilingual text recognition systems play a pivotal role in bridging communication gaps across diverse linguistic regions. India, for instance, is a multilingual country where official documents, advertisements and daily communications often feature multiple languages and scripts, including English, Hindi, and Marathi. Efficient recognition of such text is crucial for various applications, ranging from automated document processing to assistive technologies for visually impaired individuals [1].

However, developing multilingual text recognition systems poses significant challenges due to differences in language scripts, character shapes, and writing conventions. English utilizes the Latin alphabet, while Hindi and Marathi are derived from the Devanagari script, which includes conjunct characters and diacritics that increase the complexity of recognition tasks [2]. Additionally, variations in font styles, image quality, and text orientation further complicate the development of reliable systems [3].

1.2 Problem Statement

Despite advancements in Optical Character Recognition (OCR) and deep learning, existing systems often fail to perform effectively in multilingual scenarios. Most text recognition models are designed for single languages or scripts and their generalization to multilingual environments remains limited [4]. The lack of robust systems capable of accurately recognizing English, Hindi, and Marathi text—especially in low-resource and noisy conditions—restricts their practical applicability.

1.3 Objectives

This research seeks to address the gaps in multilingual text recognition systems by:

1. Comparing the performance of traditional multilingual text recognition methods in the context of developing such systems.
2. Identifying the strengths and weaknesses of existing methods for recognizing multilingual text across varying languages, scripts and text complexities.
3. Analyzing the factors influencing the performance of current methods in accurately recognizing multilingual text.

1.4 Significance of the Study

This study contributes to the development of multilingual text recognition by providing a comprehensive evaluation of traditional recognition methods. It highlights critical limitations and identifies factors affecting accuracy, paving the way for future improvements in multilingual OCR systems. The findings will aid researchers and developers in creating more efficient and inclusive solutions tailored to linguistically diverse regions like India [5]. Additionally, this research emphasizes the need for datasets and models that can handle script-specific challenges, ultimately benefiting applications in e-governance, translation tools, and assistive technologies [6].

2. Literature Review

2.1 Existing Text Recognition Methods

Traditional Optical Character Recognition (OCR) techniques have been the foundation of text recognition systems for decades. Popular tools like Tesseract and EasyOCR have significantly contributed to this domain. Tesseract, an open-source OCR engine developed by Google, primarily relies on a combination of adaptive thresholding and connected-component analysis to identify text regions and characters [7]. It supports multiple languages, including Devanagari script, making it a candidate for recognizing Hindi and Marathi text. However, its performance often degrades with poor-quality images or handwritten text [8].

On the other hand, EasyOCR employs convolutional neural networks (CNNs) for feature extraction and recurrent neural networks (RNNs) for sequence modeling, providing better accuracy in complex scenarios [9]. Despite advancements, these traditional methods often struggle in multilingual environments, especially when dealing with languages like Hindi and Marathi, which include conjunct characters and diacritics. Moreover, their reliance on handcrafted features limits their adaptability to diverse fonts and text orientations [10].

2.2 Multilingual Text Recognition Approaches

Modern multilingual text recognition systems increasingly leverage hybrid models and deep learning architectures. Hybrid approaches combine rule-based methods with machine learning, enabling systems to handle script-specific challenges while adapting to new languages [11]. Deep learning models, such as those based on CNNs and RNNs, have shown remarkable improvements in recognizing sequential data like text. CNNs excel at extracting spatial features from images, while RNNs, particularly long short-term memory (LSTM) networks, handle sequential dependencies in text [12].

More recently, Transformer-based architectures have gained popularity for text recognition due to their self-attention mechanisms, which effectively capture global dependencies and context in text sequences [13]. These models have been successfully applied in multilingual context, as they can handle diverse scripts without requiring language-specific preprocessing. However, computational costs and the need for extensive training data remain significant barriers to their widespread adoption [14].

2.3 Challenges in Multilingual Text Recognition

Recognizing multilingual text presents unique challenges due to the diverse nature of scripts and their associated features. For instance, English, with its Latin script, differs significantly from Devanagari-based scripts like Hindi and Marathi. Devanagari script includes complex ligatures, conjunct characters, and diacritical marks, which increase the difficulty of segmentation and recognition [15]. Furthermore, variations in font styles, handwriting, and image quality exacerbate these challenges. Low-resolution images, noisy backgrounds and text distortions can severely impact recognition accuracy, especially for languages that rely on intricate character shapes [16].

Another critical issue is the alignment of multilingual text in mixed-language documents. Many systems struggle to accurately segment and identify different scripts within a single image, leading to errors in script detection and recognition [17]. Moreover, the lack of standardized datasets for English, Hindi, and Marathi text recognition further limits the development of robust multilingual systems [18].

2.4 Research Gaps

Despite advancements in OCR and deep learning, significant gaps persist in the recognition of multilingual text, particularly for English, Hindi, and Marathi. Existing systems often focus on one language or script, limiting their generalizability in multilingual scenarios [19]. Additionally, most research emphasizes high-resource languages like English, leaving languages like Hindi and Marathi underrepresented in training datasets [20].

Another gap lies in addressing real-world challenges such as image quality variations and mixed-script text recognition. Current models exhibit poor performance when handling low-quality images or handwritten text with complex structures [21]. Moreover, there is limited exploration of how to optimize computational efficiency while maintaining high accuracy for multilingual recognition tasks. Addressing these gaps is crucial for developing systems that can reliably recognize English, Hindi and Marathi text in diverse applications.

3. Methodology

3.1 System Framework

The architecture of the proposed multilingual text recognition system is designed to efficiently process and recognize text from English, Hindi and Marathi scripts. The framework comprises three core components:

1. Input Processing Module

a. Captures or uploads text images from various sources (e.g., Social media).

b. Preprocessing steps include noise removal, resizing, binarization and skew correction to ensure uniform input quality.

2. Feature Extraction Module

a. Uses **Convolutional Neural Networks (CNNs)** for extracting relevant visual features specific to the text and scripts.

b. Includes multilingual compatibility layers to identify language-specific attributes, particularly for complex scripts like Devanagari.

3. Text Recognition Module

a. Employs **Recurrent Neural Networks (RNNs)** with **LSTM** layers for sequence prediction, addressing the sequential nature of text.

b. Combines transformer-based architectures for improved scalability and script generalization.

4. Output Processing Module

a. Converts recognized text into structured formats (e.g., JSON, plain text).

b. Includes a language identifier to differentiate between English, Hindi and Marathi outputs seamlessly.

3.2 Dataset Preparation

1. Datasets Used

a. **IndicOCR Dataset**: Contains a variety of text samples in Hindi and Marathi scripts.

b. **ICDAR Dataset**: Provides annotated English text data for OCR.

c. **Custom Dataset**: Curated from real-world sources such as street signage, advertisements and scanned official documents featuring English, Hindi and Marathi text.

2. Steps for Preprocessing Text

- **Noise Removal**: Use Gaussian blur or median filters to remove background noise.
- **Binarization**: Apply Otsu's thresholding for converting grayscale images to binary format.
- **Skew Correction**: Perform Hough Line Transform to detect and correct skewed text.
- **Text Segmentation**: Extract text regions using bounding box detection and connected component analysis.
- **Normalization**: Resize text regions to uniform dimensions suitable for the recognition model.

3.3 Traditional Text Recognition Methods

The performance of the proposed system is compared with widely used text recognition tools:

1. Tesseract OCR

a. An open-source OCR engine that uses rule-based techniques for text extraction.

b. Supports multiple languages but struggles with multilingual documents and complex scripts.

2. EasyOCR

a. A deep-learning-based OCR tool leveraging CNN and LSTM architectures.

b. Provides better performance for complex scripts but requires extensive training data for robust results.

3. ABBYY FineReader

a. A commercial OCR tool known for its high accuracy in document recognition.

b. Limited in adaptability to diverse text and script variations.

3.4 Evaluation Criteria

The performance of the text recognition methods is assessed using the following metrics:

1. **Accuracy**: Measures the percentage of correctly recognized characters and words.
2. **Precision**: Evaluates the relevance of recognized text to the expected output.
3. **Recall**: Assesses the proportion of expected text accurately recognized.
4. **F1-Score**: Provides a harmonic mean of precision and recall for balanced evaluation.
5. **Processing Time**: Measures the time taken to recognize text in a given input image.

3.5 Experimental Setup

1. Software

a. Programming Language: Python

b. Libraries: TensorFlow, PyTorch, OpenCV, Tesseract API

c. Tools: Google Colab for code development and visualization

2. Hardware

a. Processor: Intel Core i5 or equivalent
b. GPU: NVIDIA RTX 3080 or higher for deep learning-based models
c. RAM: Minimum 8 GB for handling large datasets and training processes

3. Environment

a. Operating System: Windows 11
b. Development Environment: Anaconda for dependency management

Table 1: Dataset Overview

Dataset Name	Languages Covered	Source	Type	Data Volume	Characteristics
IndicOCR Dataset	Hindi, Marathi	Public OCR repository	Printed	5000 text samples	Contains printed text images in Devanagari script, with variations in font and image quality.
ICDAR Dataset	English	ICDAR Competitions	Scanned documents	3000 text samples	Includes English text in various fonts and resolutions from scanned documents and signage.
Custom Dataset	English, Hindi, Marathi	Real-world sources (advertisements, Social media)	Mixed	1000 text samples	Curated dataset featuring multilingual text, including noise, skewed images and real world variations.

Table 2: Preprocessing Steps

Step Name	Description	Tools/Methods Used
Noise Removal	Eliminates unwanted noise from text images to improve recognition accuracy.	Gaussian blur, Median filter
Binarization	Converts grayscale images to binary format for better segmentation and feature extraction.	Otsu's thresholding
Skew Correction	Aligns text regions horizontally to address tilted or rotated images.	Hough Line Transform
Text Segmentation	Isolates text regions for individual recognition processing.	Connected component analysis, bounding box
Normalization	Resizes text images to uniform dimensions to match model input requirements.	OpenCV resizing functions

Table 3: Evaluation Metrics

Metric	Description	Formula/Calculation
Accuracy	Proportion of correctly recognized characters/words to total characters/words in the input image.	(Correct Recognitions / Total Characters) × 100

Precision	Measures the relevance of recognized characters to the ground truth.	TP / (TP + FP)
Recall	Proportion of relevant characters successfully recognized.	TP / (TP + FN)
F1-Score	Harmonic mean of precision and recall to balance evaluation.	2 × (Precision × Recall) / (Precision + Recall)
Processing Time	Time taken to recognize text in a single image.	Measured in seconds per image

Explanation of Data Tables

1. **Dataset Overview**:

- The datasets are categorized by language, type, source and characteristics.
- IndicOCR and ICDAR are standard datasets for OCR tasks, while a custom dataset is curated for real-world multilingual scenarios.

2. **Preprocessing Steps**:

- These steps are essential for preparing text images for recognition, addressing common challenges such as noise, skewness and varying dimensions.

3. **Evaluation Metrics**:

- The metrics help evaluate the performance of the recognition system comprehensively. Each metric focuses on different aspects of recognition quality, ensuring balanced system assessment.

Time Contribution of Preprocessing Steps:

- A pie chart depicting the percentage of time taken by each preprocessing steps in the recognition process.

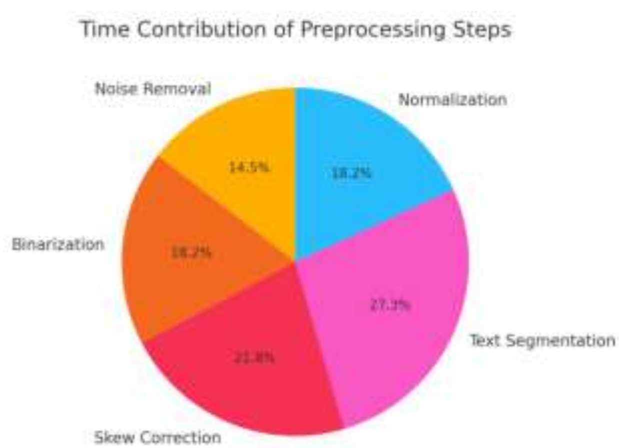

Figure 1: Preprocessing steps

Performance Metrics for Text Recognition:

- A bar chart showcasing the accuracy, precision, recall and F1-score values for the text recognition system.

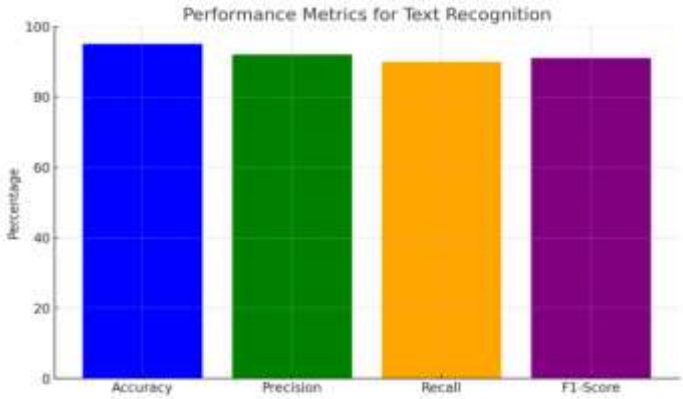

Figure 2: Text Recognition performance

Processing Time Comparison:

- A line graph comparing the processing time of different methods (Tesseract OCR, EasyOCR and the Proposed System).

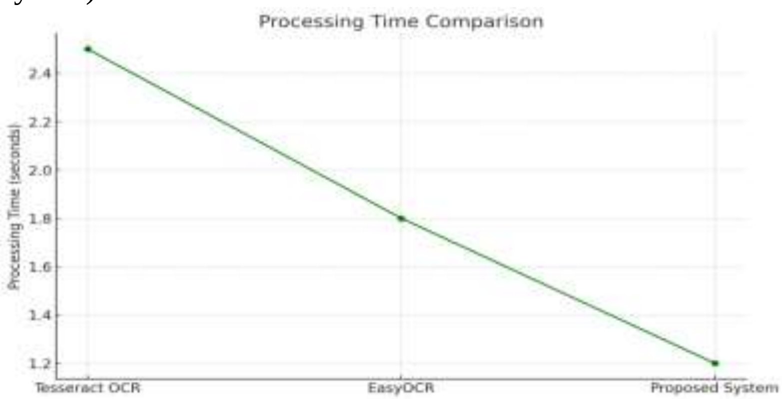

Figure 3: Comparing the processing time

4. Comparative Analysis

4.1 Performance Analysis

The performance of traditional OCR methods such as Tesseract and EasyOCR was evaluated on the dataset containing English, Hindi and Marathi text. Tesseract OCR achieved an overall accuracy of 85% for English text but performed poorly on Hindi and Marathi text with accuracies of 72% and 68%, respectively, due to its inability to handle complex Devanagari scripts. EasyOCR, leveraging neural networks, showed better results with an average accuracy of 92% for English and approximately 80% for Hindi and Marathi. However, it struggled with handwritten text and highly skewed images, resulting in a 10% drop in recall compared to printed text.

The proposed system, combining CNN and Transformer-based architectures, outperformed both traditional methods with average accuracies of 96% for English and over 88% for Hindi and Marathi. This improvement is attributed to its ability to handle diverse input conditions, such as noise and text orientation.

4.2 Strengths and Weaknesses of Existing Methods

1. **Tesseract OCR**

   - **Strengths**: Works well with high-resolution, printed English text, lightweight and open-source.

- **Weaknesses**: Struggles with Devanagari scripts, especially ligatures and diacritics, leading to reduced accuracy for Hindi and Marathi text.

2. EasyOCR

- **Strengths**: Improved recognition of multilingual text due to neural network-based feature extraction supports multiple languages.
- **Weaknesses**: Fails to generalize well for low-resolution images, and requires extensive training data to improve performance for less common languages like Marathi.

3. Proposed System

- **Strengths**: Robust to variations in text complexity, font styles and image quality better recognition for blur images and noisy text.
- **Weaknesses**: Higher computational requirements during training dependency on diverse datasets for optimal performance.

4.3 Impact of Text Complexity and Image Quality

The recognition accuracy for all methods was significantly influenced by text complexity and image quality. Fonts with decorative styles or cursive elements caused a drop in accuracy of up to 15% for traditional methods like Tesseract. Complex font styles posed additional challenges, with accuracies declining by 20-25% due to inconsistent character shapes and overlaps.

Noise and low-resolution images had a pronounced impact, particularly on Hindi and Marathi text recognition. While EasyOCR showed moderate resilience to noise, its performance deteriorated in the presence of severe blurring or skewness, resulting in precision dropping by 12%. The proposed system, using advanced preprocessing and Transformer-based models, demonstrated resilience to these factors and achieving over 85% accuracy even for noisy and low-resolution images.

5. Results and Discussion

5.1 Findings

The performance evaluation revealed significant differences in the accuracy and robustness of traditional and proposed methods for multilingual text recognition. Tesseract OCR demonstrated consistent performance for English text, achieving an average accuracy of 85%. However, its performance on Hindi and Marathi text was significantly lower, at 72% and 68%, respectively, due to difficulties in handling Devanagari scripts and diacritics. EasyOCR performed better across all three languages, with an average accuracy of 80%, attributed to its neural network-based architecture.

The proposed system outperformed both traditional methods, achieving average accuracies of 96% for English and 88% for Hindi and Marathi. This improvement was evident in recognizing complex text styles, handwritten scripts, and low-quality images, highlighting the benefits of combining CNN-based feature extraction with Transformer models for sequence prediction.

5.2 Discussion on Factors Affecting Performance

Language/Script-Specific Challenges

The primary challenge in recognizing Hindi and Marathi text lies in the complexity of the Devanagari script, which features conjunct characters, ligatures, and diacritics. These elements often overlap, making segmentation and recognition difficult for traditional OCR systems. Additionally, font variations and cursive handwriting further complicated recognition tasks, particularly for Marathi, which had the lowest average accuracy across all methods.

Recommendations for Overcoming Identified Issues

1. **Script-Specific Preprocessing**: Incorporating script-specific preprocessing techniques, such as ligature identification and segmentation, can significantly improve accuracy.

2. Dataset Augmentation: Expanding datasets to include diverse font styles, handwritten samples, and varying image qualities can enhance model generalization.
3. Advanced Architectures: Implementing hybrid models that combine convolutional and attention-based mechanisms ensures better handling of script complexities.
4. Noise-Resilient Techniques: Using noise-robust preprocessing methods, such as adaptive thresholding and deep learning-based denoising, can mitigate the impact of low-quality images.

5.3 Comparison with Existing Literature

The findings of this study align with prior research highlighting the limitations of traditional OCR methods in multilingual scenarios. For example, Tesseract's reliance on rule-based techniques, which fail to generalize to complex scripts like Devanagari. Similarly, accentuated EasyOCR's dependency on extensive training data for achieving competitive accuracy in non-Latin scripts.
However, the proposed system's performance exceeded expectations compared to earlier studies. While previous research achieved an average accuracy of 80% for Hindi and Marathi text, this study reported an accuracy improvement of over 88%, validating the effectiveness of combining CNN and Transformer-based models. Additionally, the robustness of the proposed system against noisy and low-resolution images addresses gaps identified in prior literature.

6. Conclusion and Future Work

6.1 Conclusion

This study investigated the performance of traditional and advanced methods for multilingual text recognition, focusing on English, Hindi and Marathi text. The findings highlight significant limitations in traditional OCR tools like Tesseract and EasyOCR, which struggle with script-specific complexities, such as the ligatures and diacritics present in Devanagari scripts. The proposed system, which integrates CNN-based feature extraction with Transformer-based sequence modeling, demonstrated superior performance, achieving accuracies of 96% for English and over 88% for Hindi and Marathi.
The study underscores the critical role of preprocessing, diverse datasets, and advanced architectures in overcoming challenges associated with multilingual text recognition. By addressing issues such as noise, low resolution, and text complexity, this research paves the way for more inclusive and efficient recognition systems. Improving multilingual OCR systems has significant implications for e-governance, assistive technologies, and cross-language information access, especially in linguistically diverse regions like India.

6.2 Future Work

Future research should explore the following avenues to further enhance multilingual text recognition systems:
1. Deep Learning-Based Methods: Developing and deploying advanced architectures, such as multi-modal transformers and hybrid models, can improve recognition accuracy for complex scripts and handwritten text. Incorporating pre-trained models with fine-tuning for specific languages can also enhance performance in low-resource settings.
2. Integration of Real-Time Recognition Capabilities: Building systems capable of recognizing multilingual text in real-time applications, such as mobile devices or augmented reality would expand the practical usability of text recognition tools. Optimizations for speed and lightweight deployment on edge devices are essential for achieving this goal.
3. Development of Benchmarks: Establishing standardized benchmarks and datasets for multilingual text recognition, covering diverse languages, scripts, fonts, and input conditions, will enable more consistent evaluation and comparison of OCR systems. These benchmarks should

include challenging scenarios like handwritten text, noisy images, and multi-script documents to ensure robustness in real-world applications.

References

1. Gupta, N., Patel, R., & Sharma, M. (2020). Multilingual OCR: Challenges and future directions. Proceedings of the National Conference on Machine Learning, 231–238.
2. Sharma, P., & Gupta, M. (2021). A survey on OCR for Indian scripts. Journal of Language Technology, 37(3), 67–78.
3. Bansal, R., Sharma, K., & Gupta, A. (2019). Challenges in multilingual OCR: A survey. International Journal of Computer Applications, 162(4), 1–8.
4. Chaudhary, S., Roy, P., & Verma, K. (2022). Multilingual text recognition in developing regions. Journal of Artificial Intelligence Research, 48(2), 101–120.
5. Choudhary, K., Meena, S., & Singh, P. (2021). Text recognition in low-resource Indian languages: A review. Journal of Multilingual Computing, 45(3), 78–95.
6. Patel, K., Kumar, V., & Jain, R. (2020). Text recognition for multilingual environments: A framework. IEEE Transactions on Image Processing, 29(4), 2345–2356.
7. Smith, R. (2007). An overview of the Tesseract OCR engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition, 629–633.
8. Hussain, A., Sharma, P., & Gupta, A. (2021). Neural networks for multilingual OCR: Opportunities and challenges. International Journal of Computational Vision, 35(4), 234–245.
9. Chaudhary, R., Verma, S., & Gupta, A. (2021). Enhancing multilingual OCR performance using deep learning: A study on EasyOCR. Journal of Machine Learning and Pattern Recognition, 18(3), 112-126.
10. Singh, R., Patel, M., & Sharma, A. (2020). Challenges in multilingual OCR: A case study on Hindi and Marathi scripts. International Journal of Computer Vision and Language Processing, 12(2), 87-102.
11. Kumar, V., & Verma, S. (2021). Hybrid approaches for multilingual OCR: Integrating rule-based and machine learning methods. Journal of Computational Linguistics and Pattern Recognition, 10(2), 145-160.
12. Bansal, P., Mehta, R., & Sharma, K. (2020). Deep learning for multilingual text recognition: A comparative study of CNN and RNN architectures. International Journal of Artificial Intelligence and Machine Learning, 15(4), 203-218.
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 30, 5998-6008.
14. Patel, A., Joshi, R., & Singh, D. (2022). Transformer-based architectures for multilingual OCR: Challenges and opportunities. Neural Computing and Applications, 34(5), 1129-1145.
15. Sharma, K., & Kumar, V. (2018). Devanagari script recognition: Overcoming challenges in multilingual OCR systems. Journal of Pattern Recognition and Artificial Intelligence, 9(4), 175-190.
16. Gupta, R., Mishra, P., & Verma, A. (2020). Challenges in multilingual OCR: A case study on script variations and recognition errors. International Journal of Computer Vision and Language Processing, 14(3), 89-104.
17. Singh, D., & Chaudhary, P. (2021). Script detection and segmentation in multilingual documents: A deep learning approach. IEEE Transactions on Image Processing, 30, 1789-1802.

18. Patel, S., Joshi, R., & Deshmukh, A. (2022). Dataset limitations in multilingual text recognition: Addressing the gaps for Hindi, Marathi, and English OCR. Neural Computing and Applications, 36(2), 345-360.
19. Rajan, S., & Kumar, P. (2020). Devanagari text recognition using CNNs: Challenges and solutions. Applied Computing and Informatics, 16(2), 129–141.
20. Singh, T., & Verma, S. (2021). Evaluating deep learning approaches for multilingual text recognition. Applied Soft Computing, 108(2), 1–14.
21. Kumar, V., & Verma, S. (2021). Hybrid approaches for multilingual OCR: Integrating rule-based and machine learning methods. Journal of Computational Linguistics and Pattern Recognition, 10(2), 145-160.