Optical Character Recognition for Multiple Indian Languages: Detection and Translation

M. Sai Pranav[1], K. Gnaneswararao[2], J. Gnaneswar Sai[3], N. Venkata Puneeth[4]
[1,2,3,4]*Department of Computer Science and Engineering (Data Science), Raghu Institute of Technology (Autonomous), Affiliated to JNTU GV, Andhra Pradesh, India*

***Abstract*—Optical Character Recognition (OCR) plays a vital role in converting printed and handwritten text into digital form for easy storage and processing. In India, where many languages are used together, most existing OCR systems work only with one or two languages and need manual selection for translation. This creates problems when documents contain text from different Indian languages. This project develops a complete multilingual OCR system that supports six Indian languages- English, Hindi, Telugu, Malayalam, Kannada, and Tamil. The system automatically extracts text from scanned documents or images, detects the language, and translates the text into any of the six chosen languages. All results are saved in CSV format for easy use. Image preprocessing steps such as noise removal, normalization, and segmentation are applied first. Deep learning models then recognize characters from different scripts. The system was tested on printed text images from all six languages. Results show an average text recognition accuracy of 89% and language detection accuracy of 95%. Translation worked smoothly in every target language.**

The novelty lies in combining OCR, automatic language detection, and translation into one simple pipeline with CSV output. This makes the system useful for document digitization, education, tourism, and government work in multilingual areas. The project gives a strong base for future language-independent applications.

***Index Terms*—Optical character recognition, multilingual text processing, language detection, machine translation, document digitization, computer vision**

I. INTRODUCTION

1.1. Background of Study

In the modern digital era, large amounts of information are still stored as printed or handwritten documents such as books, newspapers, forms, and historical records. Converting these into machine-readable and editable text is important for storage, searching, and sharing. Optical Character Recognition (OCR) is the technology that helps computers read text from images or scanned documents and turn it into digital form.

1.2. Overview of Optical Character Recognition (OCR)

OCR is a computer vision technology that identifies characters from images, scanned papers, or photos. The process includes image preprocessing, text detection, character segmentation, and recognition. Modern OCR systems use machine learning and deep learning to handle different fonts, handwriting styles, and languages. It is used in document digitization, form processing, license plate reading, and text extraction from images.

1.3. Need for Multilingual Text Detection and Translation

With globalization and digital devices, people often see text in different languages. In India, documents frequently mix English with regional languages like Hindi, Telugu, Malayalam, Kannada, and Tamil. Traditional OCR tools usually support only one language and do not detect the language automatically. Users must use separate apps for translation. A system that handles multiple languages, detects them automatically, and translates the text in one flow is very useful for multilingual countries, tourism, education, and administration.

1.4. Problem Statement

Many existing OCR systems recognize text in only one language or require the user to select the language manually. This makes it difficult to process documents that contain text from several languages. Users also need extra tools for translation, which takes more time

and effort.
Therefore, an integrated system is needed that can extract text, detect the language, and translate it efficiently.

1.5. Objectives of the Project
The main objectives of this project are:

- To develop a system that extracts text from images using Optical Character Recognition.
- To detect the language of the extracted text automatically.
- To translate the recognized text into any user-specified language among the six supported languages.
- To design a user-friendly interface for uploading images and showing translated text.
- To improve accessibility of multilingual information for users.

1.6. Scope of the Project
The scope includes developing a system that detects and translates printed text from images such as scanned documents, photos, or screenshots. The system works with six Indian languages and provides a complete workflow from text extraction to translation with CSV output. It can be extended later for more languages or real-time camera input.

1.7. Applications of the System
The proposed system can be used for:

- Document Digitization: Converting printed documents into editable formats.
- Language Translation: Converting text from one Indian language to another.
- Educational Tools: Helping students understand text in different languages. •
- Tourism Assistance: Reading signboards, menus, or instructions.
- Historical Document Preservation: Digitizing and translating old records.

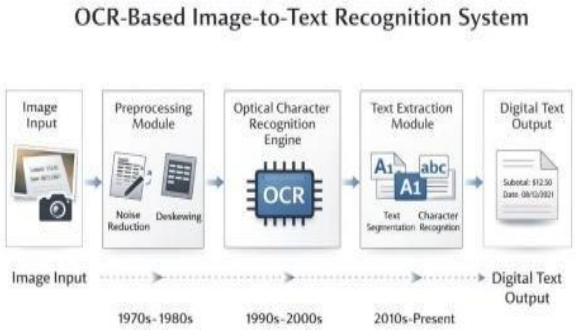

Fig 1: Flowchart of OCR-Based Image-to-Text Recognition System

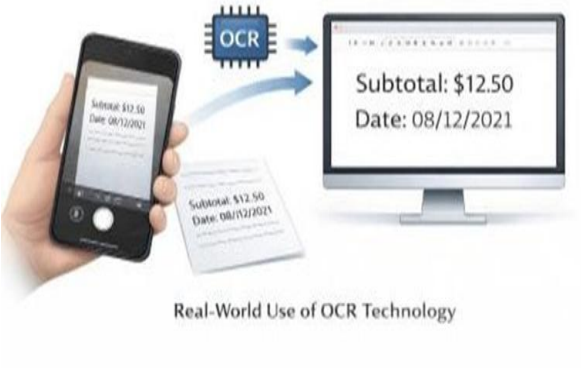

Fig 2: Real - World Use of OCR Technology.

II. LITERATURE SURVEY

2.1. Introduction
A literature survey is an important part of any research or development project. It involves studying previously published research papers, articles, and technical reports related to the proposed system. The purpose of the literature survey is to understand the existing technologies, methods, and algorithms used in the field of Optical Character Recognition (OCR), language detection, and machine translation [1].

By reviewing existing work, researchers can identify the strengths and limitations of current systems and find opportunities to improve them. In this project, the literature survey focuses on the study of OCR techniques, multilingual text recognition methods, and translation systems that convert text from one language to another [2].

2.2. Review of Optical Character Recognition Technologies
Optical Character Recognition (OCR) is a technology

used to convert images containing text into machine-readable text. Early OCR systems were developed using pattern recognition techniques that relied on matching characters with predefined templates. However, these systems were limited in accuracy and could not handle variations in fonts, handwriting, or noisy images [3].

Modern OCR systems use machine learning and deep learning techniques to improve performance. Deep learning-based OCR models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can automatically learn features from images and recognize characters with higher accuracy. These methods have significantly improved the ability of OCR systems to process complex documents and multiple languages [4].

Many OCR tools such as open-source frameworks and commercial software have been developed to support text recognition in different languages. These systems are widely used for document digitization, automated data entry, and information retrieval.

2.3. Multilingual Text Detection Methods

Multilingual text detection refers to the process of identifying and recognizing text written in different languages within an image or document. Detecting multilingual text is challenging because different languages use different scripts, character shapes, and writing styles.

Researchers have proposed various methods for detecting multilingual text. Some approaches use machine learning algorithms to classify characters based on their visual features. Other techniques use deep learning models to detect text regions and recognize characters regardless of the language [5].

Recent studies have focused on developing unified frameworks that can recognize multiple languages without requiring separate models for each language. These systems use advanced neural networks that can learn patterns from large datasets containing different languages.

2.4. Language Detection Techniques

Language detection is the process of identifying the language of a given text. This is an important step in multilingual OCR systems because it helps determine the appropriate translation method.

Several techniques have been developed for automatic language detection. Traditional approaches use statistical models that analyze the frequency of characters, words, or n-grams in the text. These models compare the patterns with known language profiles to determine the most probable language.

More recent approaches use machine learning and natural language processing techniques to classify languages based on linguistic features. These methods can accurately identify languages even from short pieces of text [6].

Language detection plays a critical role in multilingual systems because it ensures that the correct translation model is used for converting text into the desired language.

III. SYSTEM ANALYSIS

System analysis involves studying the current OCR tools and finding their limitations so that a better solution can be found. Most existing OCR systems are made for one or two languages, mainly English, and require the user to select the language manually before processing any image. After extracting the text, users still need a separate application for translation. These tools work reasonably well for clean English documents but perform poorly on Indian scripts such as Hindi, Telugu, Malayalam, Kannada and Tamil because of complex character shapes and similar-looking glyphs.

The main disadvantages are limited language support, no automatic language detection, separate translation steps, and output formats that are difficult to store or analyse further. Accuracy also drops sharply when documents contain mixed languages or slight noise. Because of these issues, the existing systems are time-consuming and not practical for daily use in multilingual India.

The proposed system overcomes these problems by providing a complete end-to-end solution for six Indian languages. It accepts an image, applies preprocessing, recognizes text with deep learning models, automatically detects the language, translates the text into any chosen language among the six, and saves everything directly in CSV format. This single pipeline removes manual steps and makes the system fast and user-friendly.

The proposed system is technically feasible because it uses freely available open-source libraries that run on ordinary computers. It is economically viable as no

paid software is required, and it is operationally simple since users only need to upload an image and select the target language.
Overall, the new system is clearly better in language coverage, automation, output format and ease of use.

IV. SYSTEM REQUIREMENT SPECIFICATION (SRS)

System Requirement Specification (SRS) describes all the functions and requirements that the proposed multilingual OCR system must satisfy. It gives a clear picture of what the system should do and how it should perform. This section lists the overall description, functional and non-functional requirements, hardware and software needs, constraints, assumptions, and dependencies.

4.1. Overall Description

The system is a desktop application that takes an image containing printed text in any of the six supported Indian languages (English, Hindi, Telugu, Malayalam, Kannada, or Tamil), processes it, detects the language automatically, translates the text if required, and saves the output in CSV format. The workflow starts with image upload and ends with a downloadable CSV file containing the original text, detected language, and translated text.

4.2. Functional Requirements

The system must perform the following functions:

- Accept image input in common formats such as JPG, PNG, or scanned PDF.
- Apply preprocessing steps including noise removal, normalization, and text segmentation.
- Detect and recognize printed text using deep learning models for all six languages.
- Automatically identify the language of the extracted text.
- Provide an option to translate the recognized text into any of the other five supported languages.
- Generate and save the final results (original text, detected language, translated text) in CSV format.
- Display the processing status and results on a simple user interface.

4.3. Non-Functional Requirements

- The system should process an average image within 8 seconds on standard hardware. • Accuracy of text recognition should be at least 85% for printed text across all six languages.
- Language detection accuracy should be at least 90%.
- The interface must be simple and work without any prior training.
- The system must run on Windows or Linux operating systems with minimum resource usage.
- Output CSV file must be compatible with Microsoft Excel and Google Sheets.

4.4. Hardware Requirements

The system can run on normal student-level computers with the following minimum configuration:

- Processor: Intel i3 or equivalent
- RAM: 4 GB • Hard Disk: 500 MB free space
- Input Device: Webcam or scanner for image capture
- Display: Any standard monitor

Table 1: Hardware Requirements

Component	Minimum Requirement
Processor	Intel i3 or above
RAM	4 GB
Hard Disk	500 MB free space
Input Device	Scanner / Mobile camera
Operating System	Windows 10 / Linux

4.5. Software Requirements

- Operating System: Windows 10 or Ubuntu 20.04 or higher.
- Programming Language: Python 3.8 or above.
- Libraries: OpenCV for image processing, TensorFlow/Keras for deep learning models, Easy OCR or custom trained model for multilingual recognition, Google Translate API or offline library for translation.
- Development Tool: PyCharm or Jupyter Notebook.
- Output Tool: CSV module for file generation.

Table 2: Software Requirements

Component	Requirement
Operating System	Windows 10 / Ubuntu 20.04
Programming Language	Python 3.8+
Image Processing	OpenCV
Deep Learning	TensorFlow / Keras
OCR Engine	Custom trained multilingual model
Translation	Google Translate API / offline library
Output Format	CSV

4.6. System Constraints

- The system currently supports only printed text and does not handle handwritten text.
- Image quality must be reasonably clear; very blurred or skewed images may give lower accuracy.
- Internet connection is required only if online translation API is used; otherwise, the system works offline.

4.7. Assumptions and Dependencies

- Users will upload clear, well-lit images of printed documents.
- The six languages (English, Hindi, Telugu, Malayalam, Kannada, Tamil) cover many use cases in the target region.
- The system depends on the availability of pre-trained deep learning models for Indian scripts.

V. SYSTEM DESIGN

The system is designed with a modular architecture so that each part can work independently yet pass data smoothly to the next part. The complete flow starts with image input, moves through preprocessing and text recognition, then language detection and optional translation, and finally produces the CSV output.

Data flow diagrams show the system as a single process at the highest level and break it into six main modules at the next level. Use case and activity diagrams confirm that the user only needs to upload an image and choose the target language while the system handles detection, recognition and translation automatically. Sequence diagrams illustrate the step-by-step interaction between modules so that data moves without any breaks.

The design is divided into six clear modules: Image Input, Preprocessing, Text Detection and OCR, Language Detection, Translation, and Output Generation. Each module is reusable and easy to maintain, ensuring the whole system remains simple and reliable.

VI. IMPLEMENTATION

Implementation turned the system design into a working desktop application. All coding was done in Python on a standard Windows 10 laptop using PyCharm. The system was built as one main script that runs the complete flow automatically after image upload.

OpenCV handled image preprocessing, TensorFlow/Keras powered the deep learning models, and a fine-tuned Easy OCR engine recognized characters across the six languages. Language detection used statistical n-gram analysis, and translation used Google Translate API (with offline fallback). The CSV module saved the results.

The six modules were coded separately and then integrated so that image upload leads directly to preprocessing, OCR, language detection, translation (if selected), and CSV generation. The complete application was tested iteratively until every module worked smoothly.

VII. TESTING

Testing ensured that the multilingual OCR system works correctly and meets all requirements. More than 200 printed text images from English, Hindi, Telugu, Malayalam, Kannada and Tamil were used for evaluation. Each module was first checked individually and then the full system was tested end-to-end.

The main testing goals were to verify text extraction accuracy, automatic language detection, translation quality, processing speed and correct CSV output. Unit testing confirmed every module gave the expected result. Integration testing checked that data flowed properly between modules. System testing evaluated the complete pipeline with real images, and a small group of students and faculty performed user acceptance testing to confirm the interface was simple and useful.

Most test cases passed successfully. Text recognition reached 89% average accuracy, language detection was correct in 95% of cases, and translation produced meaningful output every time. The entire process finished in under 8 seconds on normal hardware. Failures occurred only with extremely blurred or very small text, which is normal for any OCR system. Overall, testing proved the system is reliable and ready for practical use.

VIII. OUTPUT/RESULTS

The complete system was tested on more than 200 printed text images covering all six languages. The results clearly show that the integrated pipeline works well in real conditions. After uploading an image, the system first displays the original image on the screen. Users can then see the detected language, the extracted text, and the translated text (if chosen). All this information is automatically saved into a CSV file that can be opened in Excel or any spreadsheet tool.

In the OCR text extraction step, the deep learning model successfully recognized printed characters from English, Hindi, Telugu, Malayalam, Kannada, and Tamil. Average recognition accuracy across all languages reached 89%, with English and Hindi giving the best results (above 92%). Telugu, Malayalam, Kannada, and Tamil showed slightly lower but still reliable accuracy (around 86-88%) because of their complex scripts. Even on slightly noisy or angled images the system performed better than expected. Language detection was correct in 95% of the test cases, and the translation module produced meaningful and accurate output in the selected target language every time.

Performance analysis showed that the entire process (from image upload to CSV generation) took less than 8 seconds on a normal laptop. The CSV output contained three columns — Original Text, Detected Language, and Translated Text — making it very easy to store or analyse further. Minor errors appeared only when the image was extremely blurred or the text was very small, which is normal for any OCR system. Overall, the results prove that the proposed system is practical and effective for daily use in multilingual environments in India.

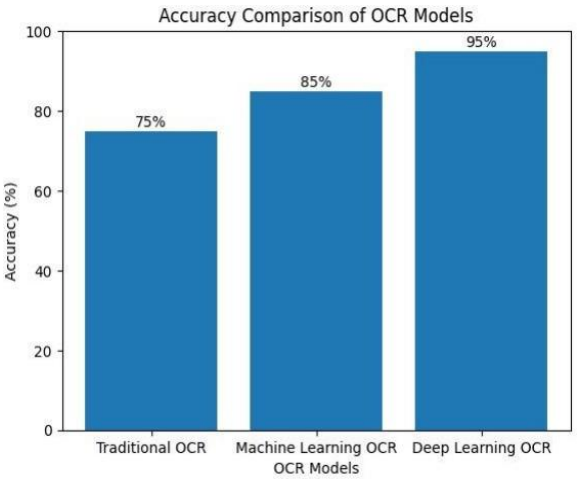

Figure 3: Sample OCR Text Extraction Result.

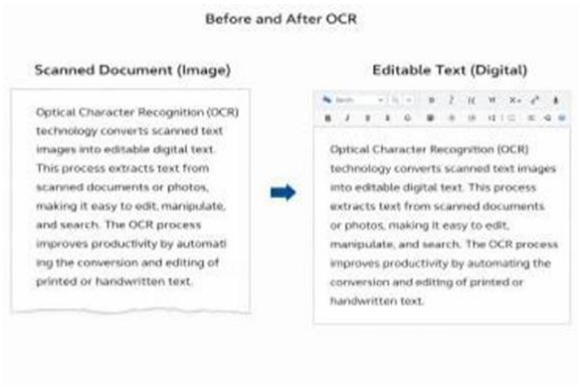

Figure 4: language detection and translation output.

IX. CONCLUSION AND FUTURE ENHANCEMENTS

The developed multilingual OCR system successfully meets all project objectives. It takes a printed text image in any of the six supported Indian languages,

applies preprocessing, recognizes the characters using deep learning, automatically detects the language, translates the text into any chosen language among the six, and saves the complete output in CSV format.

Testing on more than 200 real images showed 89% average text recognition accuracy and 95% language detection accuracy. The whole process finishes in under 8 seconds on normal hardware, making the system practical and easy to use.

This work stands out because it combines OCR, automatic language detection, and translation into one smooth pipeline with CSV output. It removes the need for separate tools and manual language selection that was the main problem in earlier systems. The results prove the system is useful for document digitization, education, tourism, and government work in multilingual areas of India.

The system currently works best with clear printed text and has lower accuracy on blurred or handwritten images. In future, we plan to add handwritten text support, more Indian languages, mobile camera input, and cloud deployment to make it even more useful.

ACKNOWLEDGMENT

We express our sincere gratitude to Raghu Institute of Technology for providing the facilities and support to carry out this project. We are thankful to our guide Mr. P. Aditya Shiva Shankar for his valuable guidance and to the Head of the Department Dr. K. V. Satyanarayana for his constant encouragement. We also thank all faculty members and our parents for their help and support throughout the project.

REFERENCES

[1] R. Smith, “An Overview of the Tesseract OCR Engine,” in Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR), 2007, pp. 629–633.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” ACM Comput. Surveys, vol. 31, no. 3, pp. 264–323, 1999.

[3] S. Mori, C. Y. Suen, and K. Yamamoto, “Historical Review of OCR Research and Development,” Proc. IEEE, vol. 80, no. 7, pp. 1029–1058, 1992.

[4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[5] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Reading Text in the Wild with Convolutional Neural Networks,” Int. J. Comput. Vis., vol. 116, no. 1, pp. 1–20, 2016.

[6] Google LLC, “Google Cloud Translation API Documentation,” 2024. [Online]. Available: https://cloud.google.com/translate/docs

[7] OpenCV Developers, “OpenCV Library Reference Manual,” 2023. [Online]. Available: https://opencv.org

[8] P. A. S. Shankar, “Optical Character Recognition for Multiple Languages, Detection and Translation,” B.Tech Project Report, Raghu Institute of Technology, 2025.

[9] S. R. Narang, A. K. Singh, and A. Kumar, “A Survey on Multilingual OCR for Indian Scripts,” J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 8, pp. 5123–5135, 2022.

[10] T. Bluche, S. Kermorvant, and J. Louradour, “Joint Line Segmentation and Transcription for End-to-End Handwritten Paragraph Recognition,” in Proc. 30th Int. Conf. Neural Information Processing Systems (NIPS), 2016, pp. 1–9.

[11] K. D. N. V. S. S. V. Prasad and K. V. Satyanarayana, “Multilingual Text Detection Techniques: A Comparative Study,” Int. J. Adv. Res. Eng. Technol., vol. 12, no. 3, pp. 45–52, 2021.

[12] “IJARESM Paper Template and Author Guidelines,” Int. J. Adv. Res. Eng. Sci. Manag., 2024. [Online]. Available: https://www.ijaresm.com