Automated ID and Certificate Data Extraction Using Optical Character Recognition (OCR)

*G. Tarshith, *G. Vandana, *G. Bhavana, *D. Chandra Lekha
Department of Computer Science Engineering, GVPCE(A), Visakhapatnam

***Abstract*—Optical Character Recognition (OCR) technology is essential for extracting text from scanned documents, images, and PDFs. Traditional OCR methods struggle with structured data extraction due to format variations and noise. This project presents an OCR-based Data Extraction System that combines Regular Expressions (Regex) and Machine Learning (ML) to enhance accuracy and reliability. Using Tesseract OCR, the system converts scanned text into a machine-readable format, followed by text cleaning to ensure structured output. Regex identifies key attributes, while an ML model predicts missing data when regex fails. Extracted data is structured in JSON and exported to Excel for integration and analysis. Error handling ensures smooth execution, making the system effective for applications in education, banking, and government. By combining rule-based and ML approaches, this solution improves efficiency, scalability, and accuracy in automated document processing.**

Keywords*—Optical Character Recognition (OCR), Data Extraction, Machine Learning (ML), Regular Expressions (Regex), Tesseract OCR, Document Processing, Structured Data Extraction, Text Recognition, Automation, PDF and Image Processing, Pattern Matching, JSON Data Structuring, Error Handling, AI-based Text Extraction, Information Retrieval

I. INTRODUCTION

Optical Character Recognition (OCR) is a technology that converts scanned documents, PDFs, and images into editable, searchable, and machine-readable text. The process involves multiple stages: image acquisition, preprocessing to enhance readability, character recognition using pattern- matching algorithms, and postprocessing to refine extracted text. Traditional OCR methods often struggle with structured data extraction due to variations in document formats, fonts, and noise.

To address these challenges, this project presents an advanced OCR-based system that integrates Regular Expressions (Regex) and Machine Learning (ML) techniques. By leveraging Tesseract OCR along with image processing tools, the system enhances text recognition accuracy and automates structured data extraction. The extracted data is formatted for seamless integration into digital workflows, significantly improving efficiency in document-heavy industries such as education, banking, and government.

II. METHODOLOGY

The Automated ID and Certificate Data Extraction System follows a structured methodology to efficiently extract, process, and organize text from scanned documents, PDFs, and images. The methodology consists of multiple stages, including data acquisition, OCR processing, text preprocessing, attribute extraction, validation, and structured data output.

1) Data Acquisition and Preprocessing

- Document Collection: The system supports PDFs, scanned images, and digital documents containing structured and unstructured text. Various datasets, including ID cards, certificates, and marksheets, are used to train and evaluate the system.
- OCR Text Extraction: PDFs are converted to images using Poppler for better text recognition. Tesseract OCR extracts raw text from images. Quality checks are applied to handle low-confidence OCR outputs and empty extractions.
- Text Cleaning & Normalization Removes extra spaces, newlines, and special characters to improve OCR accuracy. Standardizes text format (e.g., case conversion and noise reduction).Filters out unnecessary text using predefined rules.

2) Attribute Extraction

- Rule-Based (Regex) Extraction: Predefined Regular Expressions (Regex) are used to extract structured attributes from standardized documents. Regex helps in pattern-based text extraction, ensuring structured data retrieval.
- ML-Based Attribute Prediction (Fallback Mechanism): A Random Forest Classifier is trained using labelled text datasets to classify and predict missing attributes.TF-IDF Vectorization converts text into numerical features for ML-based classification. If Regex fails to extract an attribute, the ML model predicts the missing value.
- If both Regex and ML fail, the system returns "Not Found" to maintain clear output handling.

3) Data Validation and Error Handling

- Validation checks ensure that extracted attributes follow expected formats.
- Missing or erroneous values are logged for debugging and correction.
- Encoding issues (e.g., special characters and Unicode errors) are handled to prevent data corruption.

4) Data Storage and Export

Extracted attributes are structured and saved in:

- Excel (.xlsx) format for record-keeping and analysis.
- JSON format for seamless integration with external applications.
- Proper data formatting ensures easy retrieval and automated processing.

5) Performance Evaluation

The system is evaluated based on:

- Precision, Recall, and F1-score for OCR and ML- based extraction.
- Accuracy of extracted attributes compared to manually labelled ground truth data.

III. ADVANTAGES

A. Improved Search Capabilities

The system enables efficient indexing and retrieval of extracted data, allowing users to quickly locate specific details within large datasets. This enhances search functionality by enabling keyword searches, filters, and sorting, improving productivity.

B. Efficient Data Retrieval

Extracted data is stored in structured formats such as databases, spreadsheets, or JSON files, making it easier to access and analyse relevant information. Businesses can instantly retrieve customer details from scanned forms or extract financial data from receipts for accounting purposes.

C. Reduced Manual Effort

By automating the data extraction process, the system eliminates the need for manual data entry, reducing errors and enhancing accuracy. This automation streamlines workflows and minimizes human intervention in document processing.

D. Increased Efficiency

Structured data allows seamless integration into various applications. Extracted customer information can be imported into CRM systems, and financial data can be integrated into accounting software, eliminating redundant tasks and optimizing operational efficiency.

E. Data Mining and Analysis

The system enables businesses to analyse trends, patterns, and insights from large datasets. Retailers can evaluate sales data from scanned receipts, while healthcare providers can extract patient records for statistical analysis and research.

F. Machine Learning and AI Integration

The structured data generated by the OCR system can be used to train machine learning models and AI algorithms. This enhances predictive analytics, fraud detection, and customer behaviour analysis, enabling data-driven decision-making.

G. Scalability and Adaptability

Designed to handle large volumes of documents, the system is scalable for organizations of any size. Its adaptability allows it to process various document types, from simple text to complex layouts, ensuring wide applicability. Cloud integration further enhances scalability.

H. Cost Savings

By automating data extraction, the system significantly reduces operational costs. Businesses can save on labour expenses and minimize errors that could lead to financial losses or compliance issues.

I. Enhanced Accessibility

Structured data improves information

accessibility across different departments or systems. Legal teams can quickly retrieve contract details, while marketing teams can analyse survey-based customer feedback, fostering collaboration and informed decision-making.

J. Future-Proofing

The system's structured data approach ensures compatibility with emerging technologies such as AI, IoT, and blockchain. This future-proofs organizations, allowing them to integrate new advancements without modifying their existing data infrastructure.

IV. DATASET AND MODELS

A. Datasets:

A comprehensive dataset has been collected, including names, roll numbers, Aadhaar numbers, addresses, marks, and other relevant details from various sources. The data is meticulously organized and structured into well-defined datasets to ensure clarity and ease of access.

The datasets are compiled and stored in Excel format, allowing for efficient data management, analysis, and sharing. Each column in the Excel sheets corresponds to a specific attribute, ensuring a systematic arrangement. This structured approach facilitates seamless data processing and supports further analysis, reporting, and decision- making for the project.

B. Models and Algorithms

1) Input Processing

Accept input files (PDFs or images): Read the list of attributes to extract.

2) Text Extraction (OCR Processing)

- If the file is a PDF: Convert pages to images using pdf2image.Extract text from each image using Tesseract OCR.
- If the file is an image: Directly extract text using Tesseract OCR.
- If OCR output is empty, return "OCR Failed" error.

3) Text Cleaning & Preprocessing

Normalize text by: Removing extra spaces, line breaks, and special characters. Converting text to uppercase/lowercase for consistency.

4) Attribute Extraction

For each attribute (Name, Aadhaar Number, Registered Number, etc.): Apply Regular Expressions (Regex) for pattern matching. If a match is found, store the extracted value.

5) ML-Based Attribute Classification (Fallback Mechanism)

If Regex extraction fails for any attribute: Convert text into numerical vectors using TF-IDF Vectorization. Pass vectorized text into a trained ML model (Random Forest / SVM). Store the predicted value.

6) Handling Missing Attributes

If both Regex and ML fail, return "Not Found" for that attribute.

7) Data Storage & Output Generation

Save extracted attributes into Excel (XLSX) or JSON format. Log errors (if any) for debugging and future improvements.

V. SYSTEM ARCHITECTURE

The proposed Automated ID and Certificate Data Extraction System follows a modular architecture that ensures efficient processing, validation, and structured data output. The system is divided into multiple components, each handling specific tasks such as OCR processing, text preprocessing, attribute extraction, validation, and data export.

A. System workflow

1) Upload File

- The user uploads one or multiple ID certificates in PDF or image format.

2) Processing Steps

- File Type Validation: Checks if the uploaded file format is valid.
  - Valid → Proceed to attribute selection.
  - Invalid → Notify the user.
- Attribute Selection: The user selects the attributes to extract.
- OCR Processing: Extracts text from images or PDFs.
- Check for Data:
  - Data Found→ Proceed to text processing.
  - Not Found → Notify the user.

3) Text Processing

- Preprocessing: Cleans and formats the extracted text.
- Rule-Based Extraction: Extracts attributes using predefined patterns.
- ML-Based Extraction: Uses a machine learning model to predict missing attributes.
- Generate Output File: Creates an output file

with extracted data.

4) *Download Output*

- The user downloads the final processed file in Excel (XLSX) or JSON format.

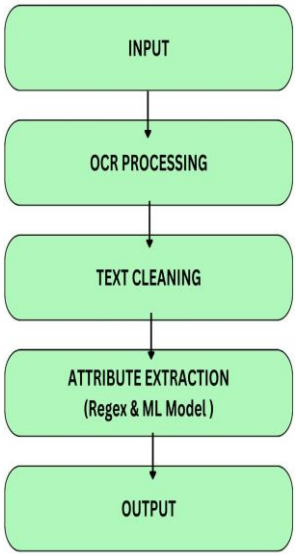

Figure 1: Flow diagram

B. *System Modules*

1) *OCR Processing Module*

- Converts PDFs and images into raw text using

Tesseract OCR.

- Uses Poppler for PDF-to-image conversion (if required).
- Handles error detection for empty or low-confidence OCR results.

2) *Text Preprocessing Module*

- Cleans extracted text by: Removing unnecessary spaces, newlines, and special characters and Normalizing text format (e.g., case conversion, noise removal).
- Improves OCR accuracy by refining raw text output.

3) *Attribute Extraction Module*

- Regex-Based Extraction: Uses predefined patterns (e.g., names, registration numbers, Aadhaar numbers, dates, amounts).
- ML-Based Attribute Prediction (Fallback Mechanism):
  - Uses TF-IDF+ Random Forest Classifier (or another ML model) to classify extracted data when Regex fails.
  - Helps in handling unstructured or non-standardized documents.

4) *Data Validation & Error Handling Module*

- Verifies extracted attributes against expected formats.
- Logs missing attributes and marks them as "Not Found" if both Regex and ML fail.
- Generates error logs for debugging (e.g., invalid formats, incorrect matches).

5) *Data Export Module*

- Saves extracted structured data in Excel (XLSX) or JSON format.
- Handles Unicode errors (e.g., replacing unsupported symbols with safe alternatives).
- Ensures proper data formatting for further processing.

6) *User Interface Module*

- Command Line Interface (CLI) for batch processing.
- Web-based Dashboard for uploading files, viewing results, and downloading structured data.

7) *Logging & Performance Tracking Module*

- Logs errors, OCR failures, and attribute extraction issues for debugging and performance monitoring.

C. *Results*

1) *WELCOME PAGE*

Figure 2: Welcome Page

2) *CHOOSING THE TYPE OF CERTIFICATE*

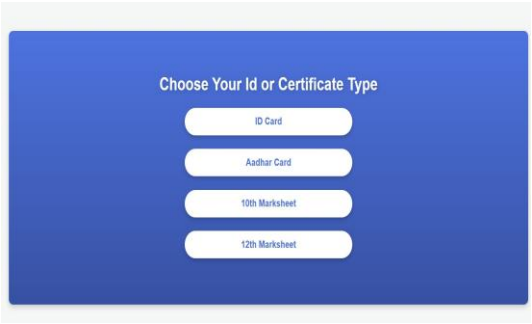

Figure 3: choosing the type of certificate

3) *UPLOADING FILES AND SELECTING ATTRIBUTES*

(single or multiple files can be uploaded)

Figure 4: selecting attributes

4) EXTRACTING ATTRIBUTES IN JSON

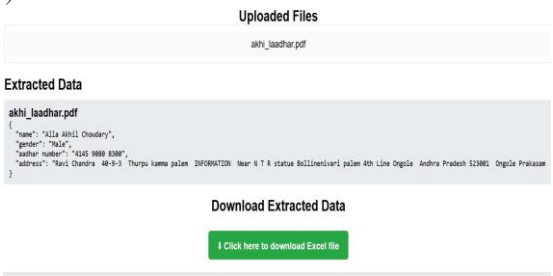

Figure 5: Extracting attributes

5) DOWNLOADING EXTRACTED DATAFILE

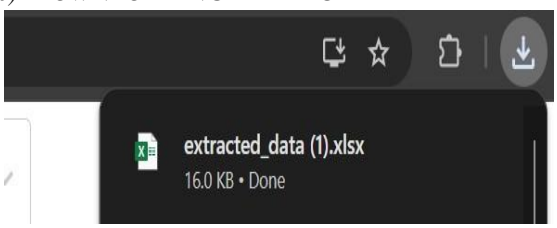

Figure 6: Download extracted datafile

6) EXTRACTED DATA IN EXCEL FILE

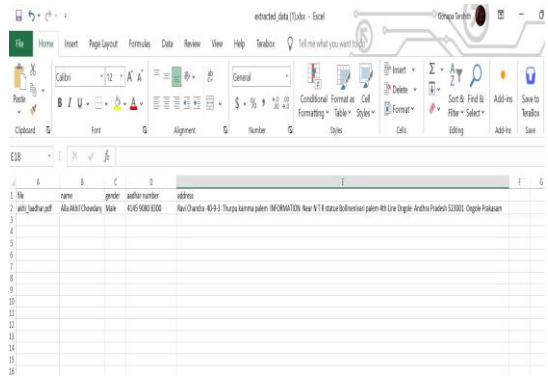

Figure 7: Extracted data in excel

D. Testing

Test case 1: Valid PDF

Expected Output (JSON Format):

```
{
    "file": "id_card.pdf", "attributes": {
                "name": "John Doe",
                "dob":  "1995-06-15",

    "roll no": "123456",
    "mobile":  "9876543210"
  }
}
```

Test case 2: Blurred PDF

Expected Output (JSON Format):

```
{
    "file": "blurred_aadhar.pdf",
    "error": "Text could not be extracted clearly. Please upload a clearer document."
}
```

Test case 3: Multiple PDFs uploaded

Expected Output (JSON Format):

```
[
    {
        "file":
        "id_card.pdf",
        "attributes": {
        "name": "John
        Doe",
          "dob":  "1995-06-15"
        }
    },
    {
}
},
{
"file": "aadhar_card.pdf", "attributes": {
"name": "Amit Sharma", "dob": "1992-04-10"
}
}
]
```

Test case 4: No Attributes Selected

Expected Output (JSON Format):

```
{
"error": "Please select at least one attribute."
}
```

Test case 5: Large PDF uploaded (more than 50 pages)

Expected Output (JSON Format):
{
"error": "File too large. Please upload a document with fewer pages."
}

Test case 6: All Attributes Selected
Expected Output (JSON Format):
{
"file": "id_card.pdf", "attributes": { "name": "John Doe",
"dob": "1995-06-15",
"roll no": "123456",
"blood group": "O+",

"mobile": "9876543210",
"emergency contact": "+91-9876543211"
}
}

VI. CONCLUSION

The OCR-based data extraction system automates document processing by combining Tesseract OCR, regex-based pattern matching, and machine learning. It efficiently extracts structured information from scanned documents, PDFs, and images, reducing manual effort and improving accuracy.

The system follows a multi-step approach:

1. OCR Extraction – Tesseract extracts raw text from images.
2. Text Processing – The extracted text is cleaned and normalized.
3. Pattern Matching – Regular expressions (Regex) detect structured data.
4. Machine Learning – If regex fails, a model (Random Forest or SVM) predicts attributes.

This hybrid approach ensures accuracy even with OCR errors or varying document formats. The system exports structured data in Excel (XLSX) and JSON, making it easy to integrate into enterprise applications for automation and compliance.

Future improvements include deep learning for better text recognition, NLP for smarter entity extraction, and support for multiple languages. With its scalable and efficient design, this system has applications in finance, healthcare, legal, and government sectors.

VII. FUTURE SCOPE

The OCR-based data extraction system has great potential for future improvements. Key areas of enhancement include:

1. Improved OCR Accuracy

Using deep learning models like Google Vision OCR or AWS Textract for better text recognition. Applying CNNs and Transformers (TrOCR, Donut) to handle handwritten and low-quality text.

2. Smarter Information Extraction

Using NLP models (BERT, spaCy) to extract meaningful data. Improving entity recognition and classification for better accuracy.

3. Multi-Language & Handwritten Text Support

Supporting multiple languages with advanced OCR engines (Google Vision, Paddle OCR). Enhancing recognition of handwritten text.

4. AI-Powered Automation:

Integrating with Robotic Process Automation (RPA) for automatic document handling.Connecting with business systems like ERP and document management software.

5. Cloud & API Integration

Developing a cloud-based OCR API for mobile and web apps. Using serverless computing (AWS Lambda, Google Cloud Functions) for efficiency.

6. Self-Learning Models

Implementing adaptive learning to improve accuracy based on user feedback. Using reinforcement learning to refine OCR predictions over time.

7. Real-Time Mobile Processing

Creating a mobile app for document scanning and instant data extraction. Adding real-time text editing features like Google Lens.

8. Enhanced Security & Privacy

Enabling on-device OCR to protect sensitive data. Using encryption techniques to ensure secure document processing.

By leveraging AI, cloud computing, and automation, this system can become a highly scalable and intelligent document processing solution for industries like finance, healthcare, legal, and government sectors.

VIII. ACKNOWLEDGMENT

We express our heartfelt gratitude to Gayatri Vidya Parishad College of Engineering (Autonomous) for providing a conducive

environment for research and study. We sincerely thank my coordinator, DR. P.PRAPOORNA ROJA for her invaluable guidance, continuous support, and insightful feedback throughout this study. Her encouragement and expertise have been instrumental in the successful completion of this research.

REFERENCES

[1]. Herl'any, Slovakia,"Key-Value Pair Searching System via Tesseract OCR and Post Processing", IEEE 19th World Symposium on Applied Machine Intelligence and Informatics, January 21–23 DOI: https://sci-hub.se/https://ieeexplore.ieee.org/document/9378680,

[2]. SanjivK.Bhatia, "Regular Expressions", University of Missouri – St. Louis St. Louis, MO 63121, DOI: https://www.umsl.edu/cmpsci/about/People/Faculty/Sa njivBhatia/unIX.Pdf

[3]. Zheng Huang; Kai Chen; Jianhua He; Xiang Bai; Dimosthenis Karatzas; Shijian Lu , "ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction" 5th May, 2019, DOI: https://ieeexplore.ieee.org/abstract/document/8977955

[4]. Ray Smith, "An Overview of the Tesseract OCR Engine",2008, DOI: https://sci-hub.se/https://ieeexplore.ieee.org/document/4376991

[5]. G¨oksel B˙IR˙IC˙IK∗, Banu D˙IR˙I, Ahmet Coskun SONMEZ, "Abstract feature extraction for text classification" 03.02.2011, DOI: https://sci- hub.se/10.3906/elk-1102-1015