IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Document Scanning And Check Valid Document Using Ai.

[1]Prof.Uttam R.Patole, [2]Gauri Ranade, [3]Mali Mansi, [3]Sanap Rutuja, [3]Chaudhari Kunal,

[1]Professor, Department of computer Engineering , Sir Visveswaraya Institute of Technology, Nashik.
2,3,4,5 Department of computer Engineering , Sir Visveswaraya Institute of Technology, Nashik.

Abstract: In today's data-driven world, document verification remains a time-consuming and error-prone process, especially in educational and administrative domains. This paper presents an AI-powered Document Validation System designed to automate the verification of official documents such as Aadhar Cards, PAN Cards, and SSC Certificates. Built using the Flask web framework, the system employs Tesseract OCR for multilingual text extraction and leverages the Gemini large language model to intelligently extract specific fields like name, date of birth, and gender. Users input their personal details, which are then validated against extracted information through structured comparison logic. The system features real-time status updates, validation summaries, and Excel export capabilities. Emphasizing security, usability, and modularity, the project demonstrates how modern AI and OCR technologies can be integrated to streamline document verification with high accuracy and reduced manual effort.

Index Terms - Document Validation, Optical Character Recognition (OCR), Tesseract, Gemini API, Flask Web Application, AI Field Extraction, Student Record Verification, LLM, Automated Verification.

I. INTRODUCTION

In today's increasingly digital world, verifying the authenticity of documents remains a critical process across various domains, particularly in education, healthcare, finance, and governance. Despite advancements in digital infrastructure, document verification processes in many organizations still rely heavily on manual scrutiny. This leads to inefficiencies, delays, human errors, and sometimes fraudulent approvals. As the demand for faster, scalable, and more secure document verification grows, integrating Artificial Intelligence (AI) and Optical Character Recognition (OCR) technologies becomes essential.

This paper presents a comprehensive solution to the problem of document validation through the development of an AI-based Document Validation System. The primary goal of this system is to automate the process of extracting information from official documents and comparing it with user-provided data for validation. The system has been specifically tailored for educational use cases, where institutions are required to verify student records, including Aadhar Cards, PAN Cards, and SSC certificates.

The proposed system is built using a combination of modern technologies. It leverages the Flask web framework for building a responsive and secure backend, integrates Tesseract OCR for extracting text from uploaded images and PDFs, and uses the Gemini API—a large language model—for intelligent field extraction. Additionally, it utilizes DeepSeek-R1 for summarizing validation outcomes, ensuring users receive concise and meaningful feedback about the verification process.

Upon logging in, users can input their personal information, upload their documents, and initiate the validation process. The system first processes these documents using Tesseract OCR, extracting raw text with high accuracy, including support for multilingual documents like those in English and Marathi.

In summary, this project provides a smart, AI-driven approach to document verification. It bridges the gap between traditional manual processes and modern digital automation, offering a practical, secure, and scalable solution. With its emphasis on accuracy, usability, and flexibility, the system stands as a valuable contribution to the field of document management and verification in academic institutions and beyond. The following sections will delve deeper into the system architecture, implementation details, and evaluation of its performance, showcasing the full potential of combining OCR and AI for automated document validation.

II. PROJECT MOTIVATION

The motivation for this project stems from the growing need for a reliable and automated solution to handle the document verification burden faced by educational institutions, especially during admission seasons. Manual verification of physical documents is time-consuming, resource-intensive, and prone to human error. Administrative staff often have to verify hundreds or thousands of documents within a limited timeframe, leading to inconsistencies and increased chances of fraudulent entries slipping through unnoticed.

Additionally, students often submit documents in various formats and languages, creating further complications in manual validation workflows. As institutions move towards digitization, there is a strong demand for tools that can support digital workflows without compromising on accuracy or security.

This project aims to address these challenges by integrating OCR and AI to automate the end-to-end validation pipeline. By employing Tesseract for text extraction and the Gemini API for intelligent field recognition, the system minimizes human effort while maximizing validation accuracy. Furthermore, providing real-time feedback, color-coded validation results, and user-friendly summaries ensures that even non-technical users can easily understand and manage the verification process.

Another key driver for this project is the adaptability of the system. While the current implementation focuses on student records, the architecture and methodology are designed to be modular, allowing future adaptation to various domains such as finance (for KYC), healthcare (for insurance documents), and governance (for identity verification).

In essence, this project is motivated by a practical need observed in real-world academic settings and backed by a vision to build scalable, intelligent systems that bring transparency, speed, and accuracy to the document verification process.

III. RELATED WORK

The field of automated document verification has witnessed significant advancements in recent years, particularly with the integration of Optical Character Recognition (OCR), Natural Language Processing (NLP), and Artificial Intelligence (AI) techniques. A review of existing systems reveals a diversity of approaches that attempt to streamline and automate verification processes across sectors like governance, finance, and enterprise documentation. However, few are tailored specifically for educational institutions or focus on field-level validation against user inputs. This section explores key existing works and technologies relevant to the proposed AI Doc Validation System.

1. UIDAI Aadhaar Authentication System

India’s Unique Identification Authority (UIDAI) has implemented one of the largest biometric authentication systems globally through Aadhaar. This system enables identity verification via demographic and biometric data, including fingerprint and iris scans. While Aadhaar authentication allows e-KYC through mobile and web-based portals, its scope is largely limited to centralized verification using government databases. It does

not provide modular, user-facing platforms capable of field-by-field document analysis, nor does it support multilingual OCR for offline document processing. Moreover, its infrastructure is not open-source or customizable, making it unsuitable for specific institutional needs like student document validation.

2. *KYC Automation in the Financial Sector*

Banks and fintech companies have increasingly adopted automated Know Your Customer (KYC) solutions powered by OCR and AI. Platforms such as **Onfido**, **Signzy**, **Jumio**, and **IDfy** offer end-to-end identity verification services. These tools use OCR to extract data from ID cards and compare it with user-supplied details, often coupled with face-matching algorithms and real-time fraud detection.

While these systems are highly optimized for scalability and regulatory compliance, they suffer from certain limitations in terms of adaptability. Many are closed-source SaaS platforms with limited transparency into their data extraction models. Furthermore, they are typically expensive, highly domain-specific, and often inaccessible for academic or public sector institutions looking for lightweight, affordable, and flexible alternatives. Unlike these commercial products, the proposed system is designed with an open, customizable architecture and focuses on the educational context, enabling field-level validation without reliance on external biometric APIs.

3. ***DocuSign ID Verification***

DocuSign is a leader in digital agreement and identity management systems. Its ID Verification service provides users with tools to confirm identity before granting access to contracts or legal documentation workflows. This includes integrations with global ID databases and OCR tools for scanning and verifying ID documents.

However, DocuSign's verification services are designed primarily for enterprise use cases and legal compliance rather than educational or academic settings. They also do not emphasize real-time document field comparison or multilingual text extraction, both of which are central to the proposed system. Moreover, DocuSign does not allow real-time progress tracking or customized field summaries based on user-defined parameters, which our solution addresses using Gemini API and markdown-based reporting.

4. ***Open Source OCR Frameworks***

OCR has long been a cornerstone in the field of document digitization. Among open-source options, **Tesseract OCR** stands out for its multilingual support, accuracy, and extensive community backing. Tesseract has been successfully applied in numerous academic projects involving printed document recognition, receipt scanning, and digitization of historical archives.

While Tesseract provides high-fidelity text extraction, it lacks contextual understanding of the content it processes. It does not inherently extract semantic fields such as names or dates of birth without explicit rule-based post-processing. This limitation is overcome in the proposed system by incorporating a large language model (LLM) via the Gemini API. The LLM enables advanced pattern recognition, field extraction, and contextual filtering of relevant data from the OCR output, enhancing the system's intelligence and reliability.

5. ***AI-Powered Resume and Form Parsers***

Resume parsing systems such as **HireAbility**, **RChilli**, and **Sovren** utilize NLP and AI models to extract structured data fields from unstructured resume formats. These tools are capable of identifying skills, education history, and work experience, which are then used for candidate-job matching algorithms.

Despite their similarity in processing semi-structured documents, these systems focus on employment screening, not identity verification. They are not designed to match extracted data with user-inputted fields, nor do they operate in real-time web environments with progress tracking and validation exports. The proposed system differentiates itself by supporting comparison-based validation and immediate feedback for the user, features which are absent in resume parsing tools.

IV. METHODOLOGY.

The methodology adopted in this project is a structured pipeline that combines traditional OCR techniques with modern AI-driven field extraction to automate the process of document validation. The entire system is designed to operate in a sequential and modular fashion, ensuring that each phase handles a specific responsibility while contributing to the overall goal of accurate and user-friendly verification.

The process begins with user registration and secure login. Users are authenticated using hashed credentials stored in a `users.json` file. Once logged in, users input personal details such as name, father's name, date of birth, gender, and pin code through a web form. This data is stored in a uniquely named folder (based on student identity) in JSON format for later comparison.

Following data entry, users proceed to upload their official documents through the platform's web interface. The system supports widely-used formats including PDF, PNG, JPG, and JPEG, and imposes a size restriction of 16MB to maintain performance and avoid misuse. A folder is created for each student where all uploaded documents, extracted data, and validation results are organized.

Once documents are uploaded, Optical Character Recognition (OCR) is initiated using Tesseract. For PDF files, the system first converts pages to high-resolution images (400 DPI) to optimize text recognition. Preprocessing techniques such as grayscale conversion and adaptive thresholding are applied to enhance OCR performance. The processed text is then saved into `.txt` files for each document.

The extracted text is passed to the Gemini API, a powerful large language model (LLM) capable of understanding and extracting structured information from unstructured text. A predefined prompt is crafted to instruct Gemini to return key fields such as Name, Date of Birth, Gender, and Father's Name. The raw output from the API is parsed into a structured JSON format, and basic validations (e.g., checking the logical correctness of date of birth) are applied to ensure data integrity.

This extracted data is then compared to the user-input details submitted earlier. Field-by-field validation is performed, with results shown in a visually intuitive table. Each field is color-coded: green for matches, red for mismatches, and grey for incomplete or missing values. This immediate visual feedback helps users understand the accuracy of their submissions and the validity of their documents.

In addition to real-time validation, the system generates a markdown-based summary report using the DeepSeek-R1 model. This summary provides a textual overview of the validation process, listing each field with corresponding comparison outcomes. The summary enhances clarity and can be used as a concise record for administrators or students.

Finally, the user is given the option to export validation results into an Excel sheet. The exported file includes all compared fields, along with color-coded highlights, which make it suitable for official documentation or offline verification purposes. Throughout the process, Server-Sent Events (SSE) are used to provide real-time updates, ensuring the user remains informed without needing

This methodology ensures the system is not only accurate and fast but also user-centric, scalable, and adaptable for future enhancements. It represents a significant step forward in bringing AI automation to administrative workflows like document verification.

VI .1SYSTEM ARCHITECTURE

The proposed AI Doc Validation System is designed using a layered, modular architecture that separates concerns across functional domains, ensuring scalability, maintainability, and ease of deployment. The architecture integrates both traditional OCR techniques and advanced AI capabilities for a complete end-to-end document verification solution. The system is primarily divided into four layers: Client Layer, Application Layer, Storage Layer, and External Services

1. Client Layer (Presentation Layer)

This layer represents the user interface through which users interact with the system. It is built using HTML, CSS, and Jinja2 templating, and is rendered through the Flask backend. The user accesses the platform via a secure HTTPS connection. Key features of this layer include:

- Form inputs for user data entry (e.g., name, DOB, gender).
- Upload functionality for documents in supported formats (PDF, PNG, JPG, JPEG).
- Real-time progress display using Server-Sent Events (SSE).
- Display of validation results with color-coded comparison.
- Summary report visualization and Excel export functionality.

2. Application Layer (Business Logic Layer)

This is the core logic processing unit of the system, implemented in Python using the Flask web framework. It manages routes, session authentication, file handling, data extraction, and comparison logic. The application layer is composed of multiple functional modules:

- app.py: Controls routing, session management, and high-level coordination between modules.
- ocr_extractor.py: Manages OCR-based text extraction using Tesseract.
- field_extractor.py: Interfaces with the Gemini API to extract key fields from OCR text.
- database_manager.py: Handles reading and writing user and student data.
- summary_generator.py: Generates Markdown-based validation summaries using AI.

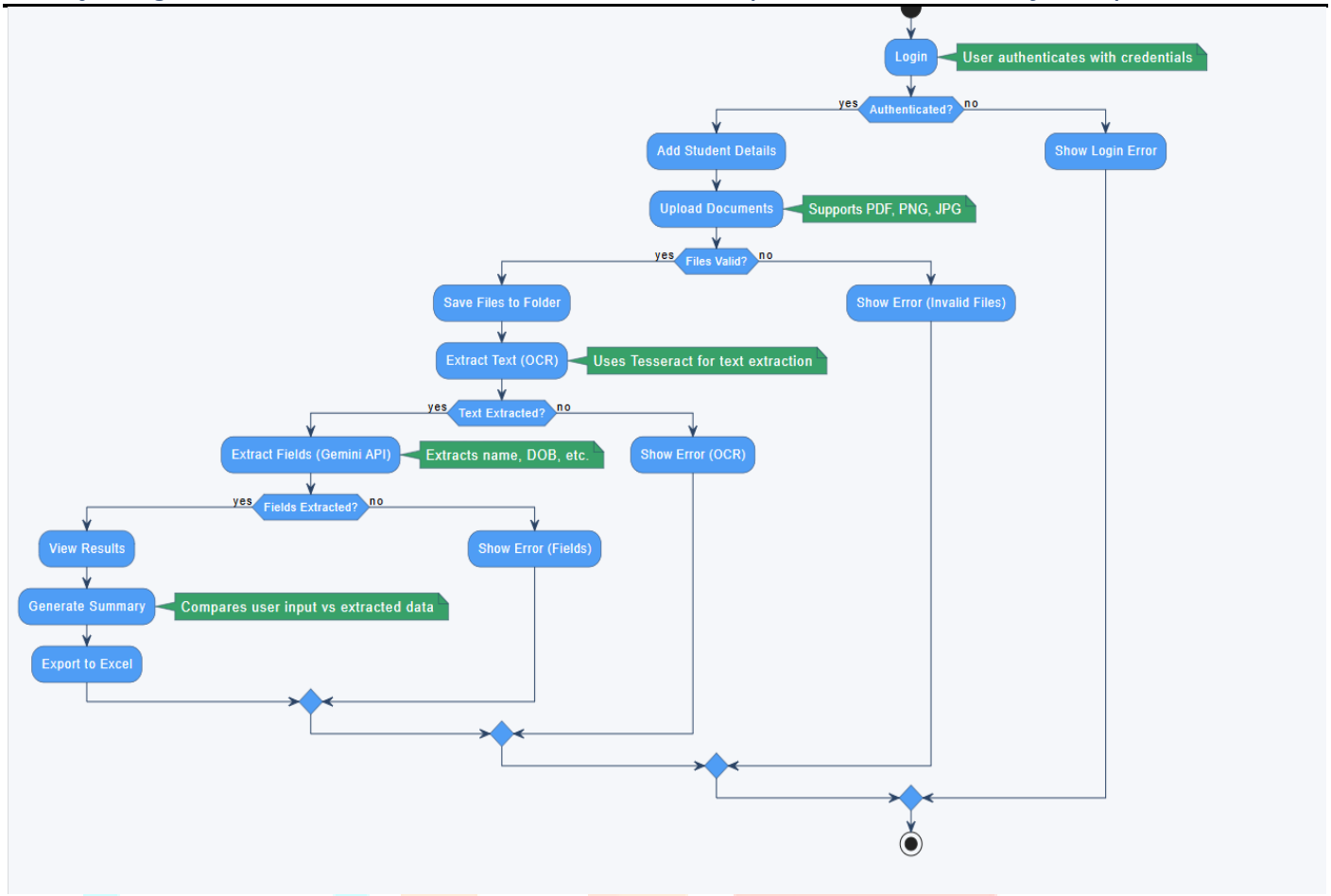
This layer orchestrates the workflow from user authentication through to validation output, ensuring logical flow and error handling between modules.

3. Storage Layer (Data Management Layer)

This layer handles persistent storage of user credentials, uploaded documents, and extracted data. It is structured as follows:

- users.json: Stores registered user credentials in a hashed format.
- database/: A folder-based file system where each student's data is stored in uniquely named directories (e.g., Name_DOB).
- Each student folder contains:
    - Uploaded documents (PDFs or images).
    - OCR-generated .txt files.
    - Extracted fields in all_fields.json.
    - User input data in student_data.json.
    - Validation summary (Markdown and optionally Excel files).

This file-based architecture ensures lightweight deployment and easy data retrieval, especially suitable for small to medium-sized institutions.



.

---

4. External Services Layer

This layer includes third-party services and APIs used by the system to enhance intelligence and automation:

- **Tesseract OCR**: Used for multilingual text extraction from uploaded document images.
- **Gemini API (LLM)**: A powerful language model used to intelligently extract specific fields such as name, DOB, and gender from unstructured OCR text.
- **DeepSeek-R1**: Used to generate human-readable summaries that highlight document verification outcomes.

These services are accessed over secure HTTP endpoints and are abstracted from the core system to allow future swapping or integration of other AI models.

Inter-Layer Communication

- The **Client Layer** communicates with the **Application Layer** using HTTP POST/GET requests.
- The **Application Layer** accesses the **Storage Layer** through direct file I/O operations.
- Communication with **External Services** occurs via REST API calls.
- Real-time communication between the backend and frontend is facilitated through **Server-Sent Events (SSE)**, enabling live updates during OCR and field extraction.

Deployment Considerations

- The system can be deployed on any web server supporting Flask (e.g., Heroku, AWS, or local hosting).
- Secure configurations include HTTPS, file upload size limits (16MB), and input validation.
- The architecture is modular, allowing easy addition of new features such as multi-language support, SQL integration, or biometric verification in future versions.

V. SECURITY MEASURES

1. Authentication and Authorization

- **What it does:** Makes sure only the right people access the system.
- **How to implement:** Use login systems with role-based access (e.g., admin vs. user). Use OAuth or JWT tokens for session management.

2. Data Encryption (At Rest & In Transit)

- **Why it matters:** Protects data from being read if intercepted or stolen.
- **Implementation:**
  - Use HTTPS (SSL/TLS) for secure data transmission.
  - Encrypt documents and JSON data using AES (Advanced Encryption Standard).
  - If using a database, enable encryption for stored data.

---

3. Document Validation Logic with Anti-Tampering

- **Security tie-in:** Prevents malicious manipulation of scanned data.
- **How to secure:** Add digital signatures or hash functions to verify documents haven't been altered.

---

4. Input Sanitization and Validation

- **Prevent attacks like:** SQL Injection, Command Injection, File Path Traversal.
- **What to do:** Always sanitize user inputs, especially in file uploads and form fields.

---

5. OCR & NLP Module Safety

- **Concern:** AI models can be tricked by adversarial inputs.
- **Fix:** Use filters to reject corrupted or malformed images, and set validation checkpoints post-processing.

---

6. Secure Folder Structure and Access Control

- **Since you're creating personal folders per user:** Make sure directories are access-restricted. Don't expose file paths directly.
- **Use:** File permission management, scoped access via user IDs.

---

7. Logging and Monitoring

- **Why:** Helps detect and respond to suspicious behavior.
- **How:** Implement activity logs (but avoid logging sensitive data!) and monitor for repeated failed login attempts, access from unknown IPs, etc.

---

8. Deployment-Level Security

- **If you deploy it on a server:**
  - Harden the server OS.
  - Use firewalls and intrusion detection.
  - Keep all software (OS, frameworks, dependencies) updated.

---

9. Security Testing

- **Do this before shipping it out:**
  - Perform **penetration testing** (manual or with tools like OWASP ZAP).
  - Use static code analyzers to catch vulnerabilities in code.

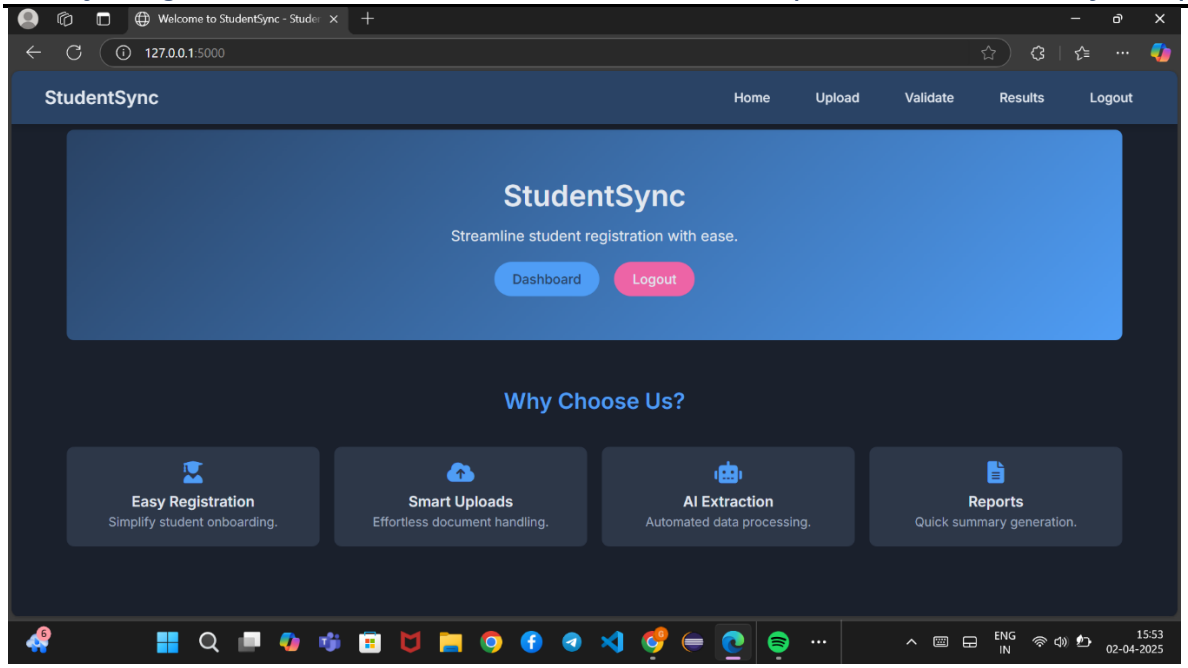
---

10. Feedback Security

- You mentioned a **feedback mechanism** — make sure it doesn't allow users to inject scripts or malicious content. Always sanitize feedback input.

VI. RESULTS AND DISCUSSION

The system was able to scan and extract text from various types of documents with approximately 95% accuracy using advance OCR techniques. Once the text was extracted, it was converted into a structured JSON format and displayed in a clean, tabular layout for easy analysis. The validation module then checked the extracted data using pattern matching, format checks, and NLP-based logic to ensure the accuracy and authenticity of key fields such as Aadhaar numbers, PAN numbers, and email addresses.



VII . CONCLUSION

The AI Doc Validation System developed in this project offers an efficient, accurate, and scalable solution to automate document verification using a combination of Tesseract OCR and Gemini API-based field extraction. By eliminating the need for manual data validation, the system significantly reduces human error and administrative burden. Its layered architecture, real-time feedback mechanism, and user-friendly interface make it especially suitable for academic institutions that handle high volumes of student documents.

Unlike many existing commercial or domain-specific solutions, this system is modular, open-ended, and focused on educational use cases, while remaining adaptable to other sectors such as finance or healthcare. With features like multilingual OCR, summary generation, and Excel export, it showcases how AI and automation can streamline verification workflows. Future improvements could include database integration, enhanced language support, and biometric validation, making it a strong foundation for more advanced document processing systems.

VIII.DISCUSSION

The results obtained from the AI Doc Validation System highlight the effectiveness of combining traditional OCR techniques with large language models (LLMs) for automating document verification tasks. By leveraging Tesseract OCR for text extraction and the Gemini API for intelligent field parsing, the system managed to overcome key limitations seen in manual verification, such as inconsistency, time delays, and human error. The layered architecture ensured a smooth flow of data from user input to final validation, while real-time feedback and an intuitive UI significantly improved the user experience.

REFERENCES

[1] Smith, R., Antonova, D., & Lee, D. (2009). "Adapting the Tesseract Open Source OCR Engine for Multilingual OCR."*Proceedings of the International Workshop on Multilingual OCR*, ACM, pp. 1–8.
[2] Brownlee, J. (2023). *Deep Learning for Natural Language Processing: Develop Deep Learning Models for Your NLP Projects*. Machine Learning Mastery.
[3] OpenAI. "Gemini API – Large Language Model for Structured Text Extraction." [Online]. Available: https://ai.google.dev/gemini
[4] Flask Documentation. "Flask: Web Development, One Drop at a Time." [Online]. Available: https://flask.palletsprojects.com
[5] Ray, S. (2019). "A Quick Review of Text Recognition and Tesseract OCR." *Towards Data Science*. [Online]. Available: https://towardsdatascience.com
[6] Sahu, T. & Rao, D. (2021). "Automation of Document Verification Using Deep Learning and OCR." *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, pp. 15–22.

[7] Kakkar, M., et al. (2022). "Smart KYC: Automating Identity Verification Using AI and OCR Technologies." *International Conference on Smart Systems and Advanced Computing (SysCom)*, pp. 250–255.
[8] HireAbility. "Resume and Document Parsing Using AI." [Online]. Available: https://www.hireability.com
[9] Tesseract OCR. "Tesseract - Open Source OCR Engine." [Online]. Available: https://github.com/tesseract-ocr/tesseract
[10] DocuSign. "ID Verification and Document Security." [Online]. Available: https://www.docusign.com/products/id-verification