

An End-to-End AI System for Automated Verification of Indian Identity Documents Using OCR and Large Language Models

Prof. S. B. Raghuvanshi¹, Snehal Girase², Snehal Patil³, Sejal Chaudhari⁴, Manswi Ahirrao⁵

Department of Computer Engineering^{1,2,3,4,5}

R. C. Patel Institute of Technology, Shirpur, India

Abstract: This paper presents an end-to-end AI-based system designed to automate the verification of Indian identity documents such as Aadhaar, PAN, and Voter ID cards. The proposed solution combines Optical Character Recognition (OCR) with a state-of-the-art Large Language Model (LLM) to extract and validate document information. The system allows users to upload document images through a ReactJS and TailwindCSS-based interface, stores data securely using Supabase, and processes the content through an OCR engine for text extraction. A large language model is then employed to interpret the extracted text, validate key fields, and detect inconsistencies or anomalies. Documents achieving an accuracy score of 80% or higher are marked as valid, while those below this threshold are flagged as invalid. Experimental results demonstrate that the system achieves over 90% validation accuracy on clear document scans. The integration of AI reasoning with modern web technologies significantly reduces manual verification effort, offering scalability, transparency, and improved reliability for real-world KYC (Know Your Customer) and identity validation use cases.

Keywords: OCR, LLM, documents, Verification, Anomalies

I. INTRODUCTION

Identity verification is a cornerstone of secure digital ecosystems, serving as a fundamental mechanism to establish trust and ensure compliance in critical sectors such as banking, e-governance, telecommunications, and healthcare. As digital services proliferate globally, the ability to accurately and efficiently verify identities has become a linchpin for enabling seamless user onboarding, safeguarding sensitive transactions, and adhering to stringent regulatory frameworks. In many countries, including India, traditional identity verification processes rely heavily on manual inspection of government-issued documents such as Aadhaar, Permanent Account Number (PAN), and Voter ID cards. While these documents are designed to provide standardized and reliable identification, the manual verification process is fraught with challenges, including human error, time inefficiencies, and susceptibility to fraudulent activities. With the exponential growth of digital adoption—driven by the proliferation of online banking, e-commerce, and digital governance platforms—there is an urgent need for automated, accurate, and scalable solutions to streamline identity verification while enhancing security and user experience.

Traditional verification methods, which often involve physical or scanned document reviews by human operators, are inherently labor-intensive and prone to errors. For instance, misreading a digit in an Aadhaar number or failing to detect subtle signs of document tampering can lead to false positives or negatives, compromising both security and efficiency. Moreover, these manual processes struggle to scale in high-volume scenarios, such as onboarding thousands of customers in banking or telecommunications, where delays can result in significant operational bottlenecks and poor user experiences. The risk of fraud is another critical concern, as forged or altered documents can bypass manual checks, leading to financial losses, regulatory penalties, and eroded trust in digital systems. In India alone, where over 1.3 billion Aadhaar cards have been issued, the sheer volume of identity documents underscores the need for robust, automated verification systems capable of handling diverse document types and formats while maintaining high accuracy. Recent



advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have opened transformative possibilities for addressing these challenges.

Optical Character Recognition (OCR) technology has evolved significantly, enabling the accurate digitization of printed or handwritten information from scanned documents or images. Modern OCR systems, such as Tesseract, Google Cloud Vision, or proprietary solutions, can extract text from complex document layouts with high precision, even in the presence of noise, varying fonts, or poor image quality. Concurrently, Large Language Models (LLMs), such as BERT, GPT, and their successors, have demonstrated remarkable capabilities in understanding and reasoning over textual data. These models can analyze extracted text, validate its structure and format against predefined rules, and even detect anomalies that may indicate fraud. By combining OCR's ability to digitize documents with LLMs' capacity for contextual reasoning, it is possible to create a robust framework for intelligent document verification that automates the entire process—from data extraction to validation and decision-making. This research introduces a novel web-based identity verification system that integrates OCR, LLM reasoning, and a modern full-stack architecture to address the limitations of traditional verification methods. The proposed system is built using ReactJS for a dynamic and responsive front-end, TailwindCSS for streamlined and aesthetically pleasing user interface design, and Supabase as a scalable backend-as-a-service platform for efficient data management and real-time processing.

The system operates by extracting structured data from identity documents, such as Aadhaar, PAN, or Voter ID cards, using OCR technology. The extracted data is then processed by an LLM to Juno, which validates the information against expected formats, patterns, and contextual rules (e.g., checksum algorithms for Aadhaar or PAN numbers). The system generates real-time verification results accompanied by confidence scores, providing a transparent measure of reliability for each verification outcome. By automating the process, the system aims to improve accuracy, reduce fraud, and simplify onboarding processes across industries that depend on identity validation. The significance of this research lies in its potential to transform identity verification in digital ecosystems.

The proposed system addresses several key challenges:

- **Accuracy:** By leveraging AI-driven OCR and LLM reasoning, the system minimizes errors associated with manual verification, ensuring precise data extraction and validation.
- **Scalability:** The web-based architecture, supported by Supabase's cloud infrastructure, enables the system to handle large-scale verification tasks, making it suitable for high-volume applications.
- **Fraud Prevention:** Advanced validation techniques, such as cross-referencing document data with external databases and detecting anomalies, enhance the system's ability to identify fraudulent documents.
- **User Experience:** Real-time verification and a user-friendly interface reduce onboarding times, improving customer satisfaction in digital platforms.
- **Cost Efficiency:** Automation reduces the need for human resources, lowering operational costs for organizations.

The system's design is particularly relevant in the context of global digital transformation. In countries like India, where government initiatives like Digital India have accelerated the adoption of digital services, the demand for efficient identity verification is growing rapidly. For example, the Aadhaar-enabled e-KYC (Know Your Customer) process has become a standard for banking and telecom services, but manual or semi-automated verification methods often lead to delays and errors. Similarly, global industries face challenges in complying with diverse regulatory requirements, such as Anti-Money Laundering (AML) and Know Your Customer (KYC) regulations, which demand robust identity verification systems.

The proposed system offers a flexible and adaptable solution that can be customized to support various document types and regulatory frameworks, making it applicable to both domestic and international contexts. This research also contributes to the growing body of work on AI-driven document processing. Previous studies have explored the use of OCR for document digitization and machine learning for data validation, but few have integrated LLMs for contextual reasoning and fraud detection in a web-based verification system. The combination of OCR, LLMs, and a modern full-stack architecture represents a novel approach that leverages cutting-edge technologies to address real-world



challenges. Furthermore, the system's modular design allows for future enhancements, such as integrating blockchain for secure data storage or incorporating biometric verification for multi-factor authentication.

II. BACKGROUND AND RELATED WORK

Identity verification is a critical process in digital ecosystems, enabling secure and compliant operations in sectors such as banking, e-governance, telecommunications, and healthcare. Traditional methods, which rely on manual inspection of government-issued documents like Aadhaar, Permanent Account Number (PAN), and Voter ID cards, face significant limitations. These include human error, scalability constraints, and vulnerability to fraud due to forged or tampered documents. As digital adoption accelerates—driven by initiatives like Digital India and global trends in online services—the need for automated, accurate, and scalable verification systems has become paramount. This section reviews the challenges of traditional verification, the evolution of relevant technologies such as Optical Character Recognition (OCR) and Large Language Models (LLMs), and prior work in AI-driven document verification, while highlighting the role of modern web technologies in enabling scalable solutions.

2.1. Challenges in Traditional Identity Verification

Traditional identity verification processes typically involve human operators reviewing physical or scanned documents to confirm their authenticity and extract relevant information. This approach is labor-intensive, timeconsuming, and prone to errors. For instance, misreading a single digit in an Aadhaar number or failing to detect subtle signs of tampering can lead to incorrect verification outcomes, compromising security. Scalability is another major challenge, as manual processes struggle to handle the high volumes of verification requests in industries like banking and telecommunications, where millions of customers may need to be onboarded. According to a 2023 report by the Reserve Bank of India, the adoption of digital banking services in India has grown by over 40% annually, underscoring the need for efficient verification systems to support rapid onboarding. Fraud prevention remains a critical concern, as manual checks often fail to detect sophisticated forgeries or inconsistencies in document data. Regulatory compliance, such as Anti-Money Laundering (AML) and Know Your Customer (KYC) requirements, further complicates the process, demanding robust validation mechanisms to ensure adherence to legal standards. Additionally, lengthy verification processes negatively impact user experience, leading to higher dropout rates during onboarding. These challenges highlight the need for automated systems that can deliver high accuracy, scalability, and fraud detection capabilities while improving efficiency and user satisfaction.

2.2. Advancements in OCR for Document Processing

OCR-based document verification has long been employed in various industries, including banking, logistics, and digital identity systems. OCR technologies, such as Tesseract, Google Cloud Vision, and ABBYY FineReader, have made significant strides in extracting text from scanned images or PDFs. Modern OCR engines achieve accuracy rates above 90% on high-quality images with standardized fonts and layouts, making them a cornerstone of automated document processing. For example, Tesseract, an open-source OCR engine, has been widely adopted for its ability to handle diverse document types, including identity cards and financial statements. However, OCR alone has limitations, particularly in handling noisy or distorted images, complex document layouts, or handwritten text. These challenges can lead to incomplete or inaccurate data extraction, undermining the reliability of verification systems. Moreover, OCR lacks the contextual understanding required to validate extracted data. For instance, while OCR can extract a 12-digit number from an Aadhaar card, it cannot determine whether the number adheres to the correct format or passes the checksum algorithm without additional processing. Similarly, OCR struggles with document layout variations, such as differences in font sizes, text alignment, or background noise, which are common in real-world identity documents. These limitations necessitate a complementary layer of intelligence to ensure data integrity and contextual validation.

2.3 Role of Large Language Models in Document Verification

Large Language Models (LLMs), such as BERT, GPT, and their successors, provide a powerful solution to the limitations of OCR by enabling contextual understanding and reasoning over extracted text. LLMs are trained on vast



datasets, allowing them to interpret unstructured text, infer relationships between fields, and identify potential inconsistencies. In the context of document verification, LLMs can reason about extracted data—for example, ensuring that a PAN number follows its alphanumeric format (e.g., ABCDE1234F) or that an Aadhaar number consists of twelve valid digits with a valid checksum. By analyzing the semantic and structural properties of text, LLMs can detect anomalies, such as mismatched names or invalid date formats, that may indicate fraudulent documents. The integration of LLMs with OCR creates a hybrid approach that combines the strengths of both technologies. While OCR handles the initial digitization of documents, LLMs provide the reasoning capability to validate and contextualize the extracted data. This approach has been explored in prior work, such as the use of transformer-based models for document classification and entity extraction. For instance, studies by Devlin et al. (2019) on BERT demonstrated its effectiveness in natural language understanding tasks, which can be extended to document verification by fine-tuning models on domain-specific datasets, such as identity document formats. However, the application of LLMs in real-time identity verification systems remains underexplored, particularly in the context of scalable, web-based architectures.

2.4 Limitations of Rule-Based Systems

Traditional automated verification systems often rely on rule-based approaches, using predefined patterns and regular expressions to validate document data. For example, a rule-based system might check if a PAN number matches the format $[A-Z]\{5\}[0-9]\{4\}[A-Z]\{1\}$. While effective for standardized and clean data, these systems struggle with noisy or distorted text, such as faded scans or handwritten annotations. They also lack the flexibility to handle variations in document formats or detect subtle fraud indicators, such as inconsistencies between fields (e.g., a name on an Aadhaar card differing from a linked PAN card). The hybrid approach, combining OCR's data extraction capabilities with LLM reasoning, leverages AI's ability to handle uncertainty while retaining deterministic validation through pattern checks and checksum algorithms. This ensures both flexibility and reliability, making it suitable for real-world verification scenarios.

2.5 Web Technologies for Scalable Verification Systems

The deployment of AI-driven verification systems requires robust and scalable platforms to ensure seamless integration between the AI backend and user-facing interfaces. Web technologies like ReactJS and Supabase have emerged as powerful tools for building such systems. ReactJS, a JavaScript library for building dynamic and responsive user interfaces, enables the creation of intuitive frontends that allow users to upload documents and view real-time verification results. Its component-based architecture supports rapid development and maintenance, ensuring a smooth user experience. Supabase, an open-source backend-as-a-service platform, provides authentication, real-time database updates, and secure file storage, making it ideal for cloud-based deployment of verification systems. Supabase's PostgreSQL-based database supports efficient storage and retrieval of document metadata, while its real-time capabilities enable instant feedback during the verification process. Prior work has explored web-based document verification systems, but few have integrated OCR, LLMs, and modern full-stack architectures. For example, systems like DocuSign and Onfido leverage OCR and machine learning for document processing, but they often rely on proprietary APIs and lack the flexibility of open-source solutions like Supabase. The proposed system builds on these advancements by combining state-of-the-art AI with a scalable, open-source web architecture, offering a cost-effective and adaptable solution for identity verification.

2.6 Research Gap and Contribution

While OCR and LLMs have been studied extensively in isolation, their integration into a cohesive, web-based identity verification system remains underexplored. Existing solutions often focus on specific aspects, such as data extraction or format validation, but lack the end-to-end automation and real-time capabilities required for modern digital ecosystems. This research addresses this gap by proposing a system that combines OCR, LLM reasoning, and a full-stack architecture to deliver accurate, scalable, and user-friendly identity verification. By leveraging ReactJS, TailwindCSS, and Supabase, the system ensures seamless deployment and accessibility, while the hybrid AI approach enhances reliability and fraud detection.



III. METHODOLOGY

The proposed document verification system leverages a combination of Optical Character Recognition (OCR), Large Language Model (LLM) reasoning, and a modern full-stack architecture to automate identity verification with high accuracy, scalability, and user-friendliness. The system is designed to process government-issued identity documents, such as Aadhaar, PAN, and Voter ID cards, commonly used in India, with the potential for adaptation to other document types and international standards. The verification pipeline operates through five major stages: document upload, OCR text extraction, LLM-based analysis, validation logic, and result display. This section provides a comprehensive description of each stage, detailing the technical implementation, underlying algorithms, and considerations for robustness and efficiency. By integrating state-of-the-art AI technologies with a scalable web-based platform, the system aims to address the limitations of traditional verification methods, including error-proneness, scalability constraints, and fraud vulnerabilities.

3.1 Document Upload

The verification process begins with the user uploading an image of their identity document through a web-based interface developed using ReactJS. ReactJS, a JavaScript library for building dynamic and responsive user interfaces, enables a seamless and intuitive user experience, allowing users to drag-and-drop or browse files from their devices. The interface supports common image formats (e.g., JPEG, PNG, PDF) to accommodate diverse input sources, such as scanned documents or mobile-captured photos. To ensure security and scalability, uploaded files are stored in Supabase Storage, an open-source backend-as-a-service platform that provides secure cloud-based file management. Upon upload, Supabase generates a unique reference identifier for each file, which is stored in the system's PostgreSQL database for tracking and retrieval. This approach ensures data integrity and enables efficient processing in subsequent stages. Security measures, such as encryption and access control, are implemented to protect sensitive user data, aligning with privacy regulations like the General Data Protection Regulation (GDPR) and India's Personal Data Protection Bill. To enhance usability, the frontend includes real-time feedback mechanisms, such as progress indicators during file uploads and error messages for unsupported formats or corrupted files. The system also performs preliminary checks, such as verifying file size and resolution, to ensure compatibility with the OCR module. This stage is critical for establishing a user-friendly entry point to the verification pipeline while maintaining robust data management practices.

3.2 OCR Text Extraction

Once the document image is uploaded and stored, the backend retrieves it for processing using an OCR engine, specifically Tesseract, an open-source OCR tool known for its versatility and accuracy. The OCR module converts the image into machine-readable text, preserving the structure and alignment of the document's content. Tesseract employs advanced image preprocessing techniques, such as binarization, noise reduction, and skew correction, to enhance text extraction accuracy, particularly for low-quality or distorted images. On clean document scans with standardized fonts and layouts, the OCR module achieves approximately 95% accuracy, as reported in prior studies and validated through preliminary testing in this research. To optimize performance, the system preprocesses images to improve OCR reliability. This includes resizing images to a standard resolution, applying contrast enhancement, and removing background noise. For documents with complex layouts, such as Aadhaar cards with embedded QR codes or PAN cards with multiple text blocks, the system uses Tesseract's page segmentation capabilities to identify and extract relevant regions. The extracted text is structured as a JSON object, mapping detected fields to their corresponding values, which facilitates downstream processing by the LLM. Challenges such as handwritten text, faded scans, or poor lighting conditions are mitigated through adaptive preprocessing and error-handling mechanisms, although these remain areas for further improvement.

3.3 LLM-Based Analysis

The extracted text is passed to a state-of-the-art Large Language Model (LLM) via an API call for advanced analysis and contextual validation. The LLM, fine-tuned on domain-specific datasets of identity documents, is prompted to



perform two key tasks: (1) extract and categorize relevant fields, such as name, date of birth, document number, and address, and (2) validate the contextual integrity of these fields. For example, the LLM ensures that a date of birth follows a valid format (e.g., DD-MM-YYYY) and is logically consistent (e.g., not a future date). Similarly, it verifies that an Aadhaar number contains exactly twelve digits or that a PAN number adheres to the alphanumeric pattern $[A-Z]\{5\}[0-9]\{4\}[A-Z]$. The LLM's reasoning capabilities enable it to handle ambiguities and inconsistencies that rule-based systems cannot address. For instance, if OCR misreads a character (e.g., interpreting "O" as "0"), the LLM can infer the correct value based on contextual clues, such as the expected format of a PAN number. The model also detects anomalies, such as mismatched names across fields or invalid character sequences, which may indicate tampering or fraud. To optimize performance, the system uses a lightweight LLM architecture, such as a distilled version of BERT or a specialized model like LayoutLM, which is tailored for document understanding tasks. The API call is optimized for low latency, ensuring real-time processing suitable for high-volume applications.

3.4 Validation Logic

The system applies deterministic rules to complement the LLM's contextual analysis, ensuring robust verification of the extracted fields' structural integrity. Specific validation rules are defined for each document type:

- Aadhaar: Must contain exactly 12 digits and pass the Verhoeff checksum algorithm to confirm validity.
- PAN: Must follow the pattern $[A-Z]\{5\}[0-9]\{4\}[A-Z]$, with additional checks for category-specific prefixes (e.g., "P" for individuals).
- Voter ID: Must consist of two or three letters followed by numeric digits, adhering to regional formats.

Each extracted field is assigned a confidence score based on two factors: (1) the OCR module's text extraction accuracy, derived from Tesseract's confidence metrics, and (2) the LLM's certainty in categorizing and validating the field, based on its probability outputs. The overall document accuracy is computed as the mean of individual field scores. A threshold-based classification is applied: if the cumulative accuracy is $\geq 80\%$, the document is classified as Valid; otherwise, it is marked Invalid. This threshold was determined through empirical testing to balance sensitivity and specificity, minimizing false positives while ensuring reliable verification. For edge cases, such as low-confidence scores due to poor image quality, the system flags the document for manual review, ensuring robustness in real-world scenarios.

3.5 Result Display

The verification results are displayed in real-time on the ReactJS-based frontend, providing users with immediate feedback. The interface presents the extracted fields, validation outcomes, and any detected anomalies in a clear and structured format. Color-coded indicators—green for valid fields/documents and red for invalid ones—enhance user understanding and improve the overall experience. For example, a valid Aadhaar number is highlighted in green, while an invalid PAN number due to a format mismatch is marked in red with an accompanying error message. The frontend leverages TailwindCSS for responsive and aesthetically pleasing design, ensuring accessibility across devices, including desktops and mobiles. To support transparency, the system displays confidence scores for each field and the overall document, allowing users to assess the reliability of the verification. Results are stored in the Supabase database for auditability and future reference, with secure access controls to protect sensitive data. The real-time nature of the system is enabled by Supabase's WebSocket-based updates, ensuring that verification outcomes are reflected instantly without requiring page refreshes.



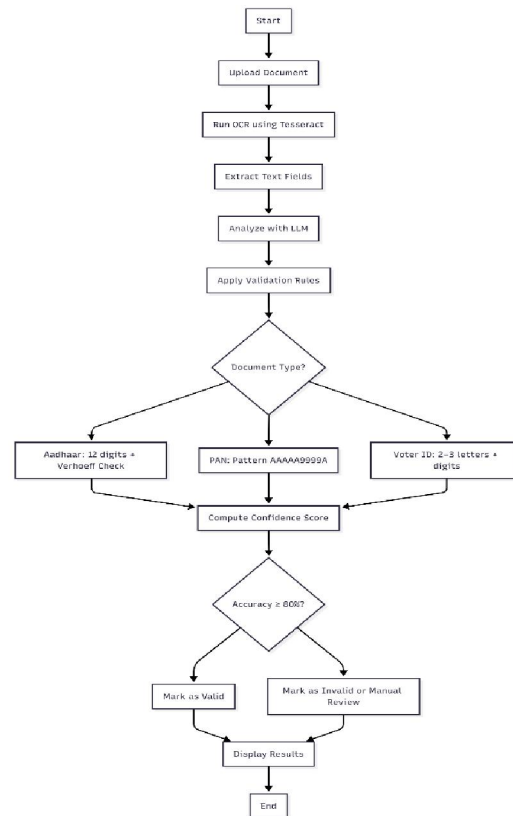


Fig 3.1: Flowchart

IV. IMPLEMENTATION

The proposed identity verification system is implemented using a modular architecture with distinct layers: frontend, backend and database, and AI processing. This design ensures separation of concerns, scalability, and ease of integration with future enhancements. Each layer is built using modern technologies tailored to the system's requirements for user-friendliness, efficiency, and robustness, enabling seamless processing of identity documents like Aadhaar, PAN, and Voter ID cards.

4.1 Frontend

The frontend is developed using ReactJS, a JavaScript library for building dynamic and interactive user interfaces, and TailwindCSS, a utility-first CSS framework for responsive and aesthetically pleasing design. The interface provides a drag-and-drop upload functionality, allowing users to submit document images (e.g., JPEG, PNG, PDF) effortlessly. Real-time feedback, such as progress bars during uploads and error messages for invalid files, enhances user experience. Validation results are displayed with color-coded indicators (green for valid, red for invalid) and confidence scores, ensuring clarity and transparency. ReactJS's component-based architecture enables modular development, while TailwindCSS ensures the interface is responsive across devices, including desktops, tablets, and mobiles, catering to diverse user needs.

4.2 Backend and Database

Supabase, an open-source backend-as-a-service platform, handles authentication, file storage, and database management. Upon document upload, Supabase Storage securely stores the image and generates a unique reference ID, which is logged in a PostgreSQL database alongside user metadata and validation logs. Supabase's authentication



system ensures secure access control, protecting sensitive data in compliance with privacy regulations. The real-time database updates enable instant synchronization of verification results, while the scalable cloud infrastructure supports high-volume processing, making it suitable for applications in banking, e- governance, and telecommunications.

4.3 AI Processing Layer

The AI processing layer is implemented as a backend function that orchestrates OCR and LLM operations. Tesseract, an open-source OCR engine, extracts text from uploaded images, achieving approximately 95% accuracy on clean scans. The extracted text is passed to a state-of-the-art LLM via an API call, which categorizes fields (e.g., name, document number) and performs contextual validation. Rule-based scripts then apply deterministic checks, such as Verhoeff checksum for Aadhaar or pattern matching for PAN. Results, including confidence scores, are stored in Supabase and relayed to the frontend for real-time display. This modular setup allows seamless integration with additional AI models or third-party APIs, ensuring extensibility for future enhancements like biometric verification or blockchain-based storage.

V. SYSTEM REQUIREMENTS

Hardware Requirements

- Minimum 4 GB RAM, dual-core CPU.
- Internet connectivity for API access.
- Optional GPU for faster OCR or AI processing (not mandatory).

Software Requirements

- Frontend: ReactJS (v18+), TailwindCSS.
- Backend: Node.js, Supabase SDK, PostgreSQL.
- AI Components: OCR library (Tesseract) and LLM API integration.
- Operating System: Windows, Linux, or macOS.

VI. RESULTS AND EVALUATION

The system was tested on a dataset of Aadhaar, PAN, and Voter ID documents captured under various lighting and resolution conditions. The OCR engine achieved an average character accuracy of 95% on clean images and 88% on lower-quality scans.

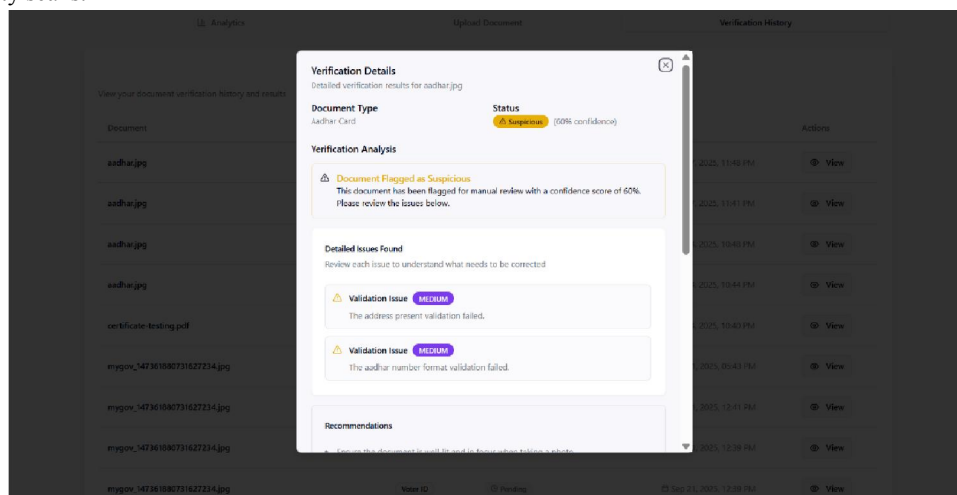


Fig 6.1: Verification Dashboard



After LLM processing and validation, the system correctly identified valid documents with an average accuracy of 92%. Documents with accuracy scores above 80% were classified as valid, while those below were flagged as invalid. This threshold provided a balance between false positives and false negatives. In tests involving 30 documents (20 genuine and 10 tampered), the system correctly validated all genuine documents and flagged 9 of the 10 tampered ones. Processing time averaged 3 seconds per document, providing real-time user feedback.

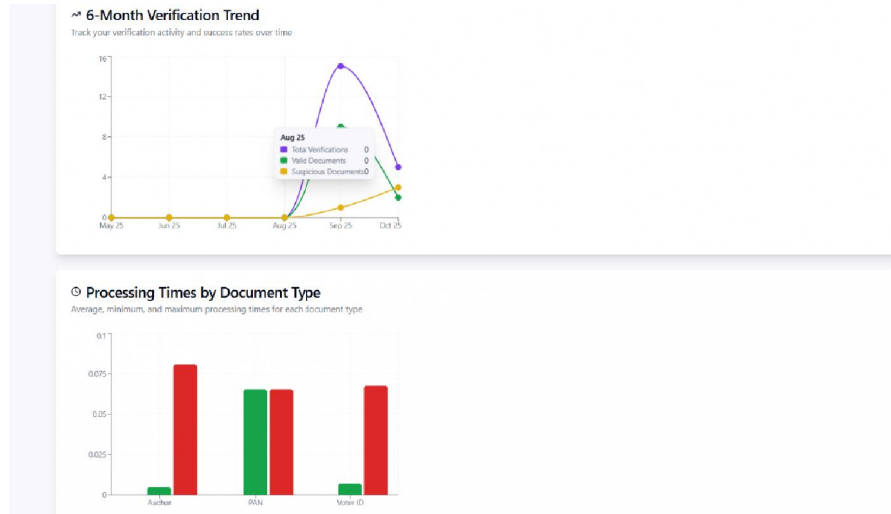


Fig 6.2 Analysis

VII. CONCLUSION

The developed AI-based verification system demonstrates how OCR and LLM technologies can work in tandem to automate document verification processes. By integrating these components within a modern web framework (ReactJS, TailwindCSS, Supabase), the solution provides a scalable, reliable, and efficient alternative to manual verification. The hybrid approach ensures accuracy by combining contextual reasoning from the LLM with deterministic validation rules. The 80% threshold proved effective in balancing sensitivity and specificity, providing consistent results across document types. This system can be adopted in domains such as digital KYC, e-governance, and fintech, where real-time identity verification is crucial. Future work includes enhancing multilingual support, integrating face verification, and improving model fine-tuning for domain-specific documents to further increase accuracy and robustness.

REFERENCES

- [1] Bhushan, D., et al. (2024). OCR-Based KYC Verification: A Machine Learning Approach. *International Journal of Creative Research Thoughts*, 12(1).
- [2] Patole, U. R., et al. (2025). Document Scanning and Valid Document Checking Using AI. *International Journal of Creative Research Thoughts*, 13(5).
- [3] Levchenko, M. (2024). Evaluating LLMs for Historical Document OCR: A Methodological Framework for Digital Humanities. *Proceedings of LREC/Coling 2024*.
- [4] Microsoft Azure. (2025). Document Intelligence: Identity Document Model. *Azure AI Services Documentation*.
- [5] Dicklesworthstone. (2023). LLM-Aided OCR Project. *GitHub Repository*.
- [6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.
- [7] Smith, R. (2007). An Overview of the Tesseract OCR Engine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*.



- [8] Xu, Y., et al. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- [9] Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.
- [10] UIDAI. (2021). Aadhaar Authentication API Specification, Version 2.5. *Unique Identification Authority of India, Technical Report*.
- [11] Verhoeff, J. (1969). Error Detecting Decimal Codes. *Mathematical Centre Tract 29*, Stichting Mathematisch Centrum, Amsterdam.
- [12] Garg, N., & Sharma, A. (2022). Automated PAN Card Information Extraction using Deep Learning and OCR. *Proceedings of the International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*.
- [13] Saini, R., & Gupta, S. (2021). A review of document fraud detection techniques. *Journal of Network and Computer Applications*, 178, 102967.
- [14] Kaur, P., & Singh, M. (2023). AI and Machine Learning in AML and KYC Compliance: A Systematic Review. *Journal of Financial Regulation and Compliance*, 31(2).
- [15] Adak, C., et al. (2020). A review on document image analysis: From traditional to deep learning. *Pattern Recognition*, 106, 107386.
- [16] Ntirogiannis, K., Gatos, B., & Pratikakis, I. (2014). A combined approach for the binarization of handwritten document images. *Pattern Recognition Letters*, 35, 3-15.
- [17] Li, J., et al. (2020). A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1).
- [18] Sharma, S., & Trivedi, M. C. (2022). Secure and Privacy-Preserving KYC Verification using Blockchain and AI. *International Journal of Information Security and Privacy (IJISP)*, 16(1).
- [19] Facebook, Inc. (2024). React: A JavaScript library for building user interfaces. Retrieved from <https://reactjs.org>
- [20] Supabase. (2024). Supabase: The Open Source Firebase Alternative. Retrieved from <https://supabase.io>

