Automated Document Verification and Risk Scoring with Liveliness Detection

Kudumula Phani Keerthan Reddy
School of CSE & IS,
Presidency University,
Bengaluru-560064,India
phanikeerthan2005@gmail.com

Abhinav K
School of CSE & IS
Presidency University,
Bengaluru-560064,India
abhinavkesavan@gmail.com

Kottapalli Lakshmi Prasanna
School of CSE & IS,
Presidency University,
Bengaluru-560064,India
Lakshmiprasanna7037@gmail.com

Jijina M.T
School of CSE & IS
Presidency University,
Bengaluru-560064,India
jijina@presidencyuniversity..in

Abstract*— We propose a digital identity verification system that incorporates document OCR, deterministic field validation, optional API checks, and concurrent face-recognition and liveness detection. It has a React-typescript frontend to facilitate document upload and review, while backend serverless functions extract structured fields and confidence scores using a vision-based model. Domain-specific validators (e.g., regex rules for specific fields, Aadhaar Verhoeff checksum, PAN structural checks) provide additional levels of reliability. Liveness and biometric matching (comparing the captured selfie with the document image) produce a score into the decision engine, which uses a multi-threshold approach to issue a verification token. We outline measurements related to our evaluation including OCR accuracy, document-type classification, liveness detection and matching performance, and end-to-end verification quality, as well as security, privacy, and robustness considerations arising from these applications. The system provides a practical and modular pipeline for automated identity verification that is rigorous and secure.

Keywords: automated document verification, text extraction, risk scoring, computer vision, optical character recognition (OCR), fraud detection, document authenticity, AI-based verification, KYC (Know Your Customer).

I. INTRODUCTION

Reliable digital identity verification is a critical building block for online services that need to securely onboard users, reduce fraud, and comply with regulatory requirements. Traditional approaches segment the process into separate pipelines for OCR, rule-based validation, and biometric checks, which can be brittle, costly, or hard to integrate. In practice, real-world deployments must handle noisy scans or photos, variable document formats (national IDs, passports, driving licenses, PANs), spoofing attempts (printed or screen-displayed images, deepfakes), and strict privacy requirements for personally identifiable information (PII).

This work describes an end-to-end, modular verification pipeline that integrates automated OCR extraction, domain-specific validation, and optional authoritative API corroboration with real-time liveness/face-matching in a single workflow. The front end is designed in React + TypeScript to lead users through upload, human-in-the-loop review, and live selfie capture. Core server-side logic runs as Supabase edge functions that

(1) Perform structured OCR and document parsing via a vision-capable generative model.

(2) apply deterministic validators (regex patterns, checksum algorithms such as the Verhoeff check for Aadhaar and PAN structural checks)

(3) score liveness and biometric face matching by comparing the live selfie to the document photo. Results are persisted in audited tables and returned as a verifiable token for downstream use.

Our contributions are threefold: (1) a practical system design that tightly integrates model-based extraction with deterministic validation and biometric scoring to reduce false accepts and rejects in end-to-end verification;

(2) concrete implementation artifacts—frontend flows, Supabase storage/DB schema, serverless functions, and validation utilities—that illustrate how to stitch generative vision outputs into production flows;

(3) an evaluation and security framework to quantify the OCR accuracy, face-match performance, liveness robustness under spoof attacks, and privacy-compliance gaps—together with recommended mitigations: RLS, signed tokens, and consent flows. The platform serves both as a working prototype to be deployed and as a reproducible research scaffold for comparing methods of OCR/biometrics and tuning verification policies.

II. RELATED WORK

The ongoing digitalization of identity verification systems has created an increasing demand for solutions for confirming document validity, user liveness, and data integrity. Numerous research initiatives have examined whether computer vision and machine learning technologies, coupled with blockchain, can be utilized to ensure secure and verifiably managed documents. However, the majority of the alternative approaches developed thus far view those domains in disjointed fashion without constructing an approach and framework for both the physical liveness verification, and digital immutability.

A. OCR and Document Extraction:

Traditional OCR systems, such as Tesseract and OCRopus, commercial cloud OCR services like Google Cloud Vision, AWS Rekognition, Azure Cognitive Services, offer robust character extraction and layout analysis; more recent work supplements OCR with deep-learning-based text detection using CRAFT and EAST and transformer-based sequence models that have significantly enhanced the noisy input handling. Finally, hybrid approaches-an amalgamation of model outputs with rule-based post-processing using regex or checksums-finding their places quite commonly in production document pipelines.

B. Document Understanding - Structured Information Extraction:

Commonly, field extraction and document classification make use of layout-aware models combined with template matching, such as LayoutLM, Donut, and DocTR. Research has emphasized end-to-end approaches that jointly model visual layout and textual content for better extraction accuracy of forms and IDs.

C. Face Recognition & Embeddings:

Deep embedding approaches like FaceNet, ArcFace, and InsightFace remain standard benchmarks for face matching; these systems compute fixed-length embeddings and compare via cosine similarity. They offer explainability-distance scores, efficiency, and well-studied evaluation protocols, ROC/AUC, EER.

D. Liveness Detection & Anti-Spoofing:

Literature on liveness detection abounds with appearance-based CNN classifiers, temporal methods-video-based motion or challenge-response-and physiological-signal approaches, such as rPPG. Spoof detection datasets and defenses study printed-photo, replay-screen, and 3D-mask attacks. Best practice typically applies multiple modalities (temporal, texture, depth) to increase robustness.

E. Generative Models for Vision Tasks:

Large multimodal generative models, such as recent Gemini/LLM-vision families, are being increasingly used for structured information extraction and reasoning over images. These models can simplify prototyping by returning structured JSON via prompting; however, they present challenges in terms of consistency, repeatability, and sensitivity to the prompt compared with specialized vision models.

F. Hybrid Pipelines (Model + Heuristics):

In previous systems, it was common for outputs from machine learning to be combined with deterministic heuristics. Simply, an optical character recognition (OCR) logic combined with domain-specific validators (e.g., checksum, regex), and business rules would be typical examples of heuristics. The goal of hybrid architecture is to reduce false positives when only a machine learning model is used, which is warranted by their involvement with identity-verification vendors.

G. (Commercial) Identity Verification Services:

Examples of product services that bundle OCR capability, document authenticity checks, and liveness/biometric matching capability together in a package would include biometrics and others developed by Onfido, Jumio, IDnow, etc. In fact, these are parceled together as practical service vendors. Typically, these will have the core features through privative models and there will be multi-step anti-spoofing checks, and sometimes fairly robust auditing/compliance features as well; they would be considered a good baseline model for prototyping or production testing.

H. Privacy-Preserving & Auditable Verification:

The literature provides research into storing verification proofs in tamper-evident ledgers, using selective disclosure credentials (VCs), and employing privacy-preserving signatures. A common theme across these approaches is either storing hash-verification artifacts on a blockchain ledger or issuing cryptographically-signed tokens (JWTs) that enable the selective disclosure of attributes, as well as zero-knowledge proofs of attribute disclosure.

I. Evaluation Processes:

Norms of evaluation protocols separate metrics at the component-level (OCR field accuracy, face-match ROC/AUC, liveness spoof detection rates) from metrics that evaluate end-to-end success rates. Public benchmarks are widely available for evaluations and include those who assess face recognition, OCR, and spoof detection (e.g., LFW, MS-Celeb, ID documents datasets, CASIA-SURF).

III. METHODOLOGY

The proposed system is a fully digital electronic, production-level identity verification system that includes document digitization, deterministic rule-based verifications, optionally - authoritative API checks, and biometric verifications using liveness and facial recognition checks. The system is designed for operational robustness, auditability, and scalability, while being resilient to attacks with strict security controls throughout the entire identity verification lifecycle.

A. OVERVIEW:

A system's main purpose is to deliver assurance through automated verification, which involves a combination of model-driven OCR, structured field extraction, heuristics based validators, and generative vision

based biometric scoring. In practical terms, the total workflow will take place in linear verification funnel:

1) Document Upload
2) Automated OCR & Data Extraction
3) Local Review and Deterministic Validation
4) Live Selfie Capture
5) Liveness and Face-Match Verification
6) Final Decision, Token Generation, and Logging

The pipeline requires that any and all identity verification requests act in an organized, transparent and auditable way.

B. SYSTEM COMPONENTS

a) Client-Side Interface:

Users can submit their identity documents, either through a The application on the client-side is responsible for guiding the user through several components, including Upload.tsx, Review.tsx, Liveness.tsx, and Verification.tsx. Together, these components manage the following features:

File selection and drag-and-drop upload (e.g., pdfs or images);

Camera access via the API (navigator.mediaDevices.getUserMedia);

Real-time correcting forms; and Securely calling backend functions.

b) Storage and Database Layer:

Supabase storage buckets serve as the storage for raw document/images and selfies. Metadata and verification output are stored in relational tables, including documents and verification_logs. Data can only be accessed by server-verified clients configured in client.ts.

c) Serverless Processing Layer:

All CPU-intensive processing is deployed as Supabase Edge Functions:

1) process-document (index.ts): performs OCR, extraction, and structured parsing.
2) verify-document (index.ts): conducts liveness analysis and face-match scoring with the generative vision model.

Each of these functions downloads the assets from storage and converts them to a base64 data URL to call the vision API with limited timing constraints. Chunking will be used to keep it memory safe.

d) Local Deterministic Validators

The module documentVerification.ts provides rule-based validator mechanisms. These include:

- Verhoeff checksum for Aadhaar,
- Structure checks for PAN/passport/DL, and regex
- Name, address, age, format heuristics
- Security flags for duplicated digits, future dates or oddities.

C. DOCUMENT PROCESSING PIPELINE

Frontend: Users can upload their files and use the webcam for liveliness detection using an intuitive interface built of HTML and PHP.

- Backend: Image preprocessing and text extraction are completed by Python scripts using OCR (text validation) and liveliness detection using OpenCV and Tesseract OCR.
- Database: A relational MySQL database has been used to safely store the resulting extracted text, time stamps, validation data, and the calculated risk scores.
- Data Flow: Each component streams separately in a component-by-component piping structure, which logically separates the logical components, which supports scalability and maintainability as shown.
- Security Measures: All data in transit is secured (utilizing TLS) and all sensitive images are removed after processing of each image. This assists with the privacy and safety of the requester's images [9], [10].

D. DETERMINISTIC DOCUMENT VALIDATION

Before proceeding with biometric authentication, the fields that were pulled from the review page are validated for correct formatting as per the given rules.

- Document type validation through pattern matching.
- Aadhaar verification using Verhoeff checksum.
- PAN alphanumeric structure validation.
- Name, address, and date of birth validation against general sense.
- Age validation within the range of 1 to 120.
- Synthetic or altered value security heuristics.

The result is a Document Validation Result object, which contains an updated validity indicator, an updated confidence score (based on OCR confidence penalized by downstream failures), and a list of items for the user to fix. These items can assist in fixing downstream biometric false positives.

E. LIVENESS AND FACE-MATCH VERIFICATION

Verify-document takes an existing document image from storage and compares it to a new user selfie.

1. Download and transform each into a data URL.
2. Invoke a vision model with a structurally-defined prompt that includes the following elements:
   - The rubric used by the model to score liveness (0 – 100)
   - The rubric based on comparing facial structure used to score face match
   - Rules to reject poor-quality images
3. Parse the response of the model into a numeric value (using a function or tool call) so that values can be normalized.
4. Determine the outcome of the comparison using very conservative threshold values:

- The OCR confidence is greater than or equal to 75
- Document validation is greater than or equal to 70
- Liveness is greater than or equal to 70
- Face match is greater than or equal to 70

If all conditions are met, then a verification token is created and returned. The current implementation is a base64-encoded JSON string; however, in a production environment, this should be replaced with a JWT token with a time-to-live limit.

F. THRESHOLD SELECTION AND TUNING

Thresholds are tuned with ROC curve analysis to find optimum trade-offs between the False Acceptance Rate (FAR) and False Rejection Rate (FRR). The conservative defaults (≥70-75) minimize risk in early deployments. However, these thresholds can be tuned down in future deployments if more tolerance for risk can be tolerated in the specific context of the domain.

G. EVALUATION METHOOLOGY

1) OCR Evaluation
   - Field-level precision/recall
   - Normalized edit distance of names, addresses, and alphanumeric IDs
2) Document Classification
   - Document-type accuracy using confusion matrices.
3) Biometric Evaluation
   - Face-match ROC/AUC
   - FAR, FRR, and Equal Error Rate (EER)
   - Liveness robustness against attack vectors: printed photos, screen replay, video and 3D mask attack vectors
4) System-Level Evaluation
   - End-to-end true positive/false positive rates.
   - Mean time-to-verify.
   - Cross-demographic fairness checks.

IV. SYSTEM ARCHITECTURE AND DESIGN

In this section, we describe the multi-tier architecture, functional pipeline, and design rationale for the proposed identity verification system. The platform is designed as a modular, cloud-engaged system, consisting of a client front-end written in React, backend services using Supabase, external vision-AI service for OCR and biometric analysis, and pre-existing validations designed for the local domain.

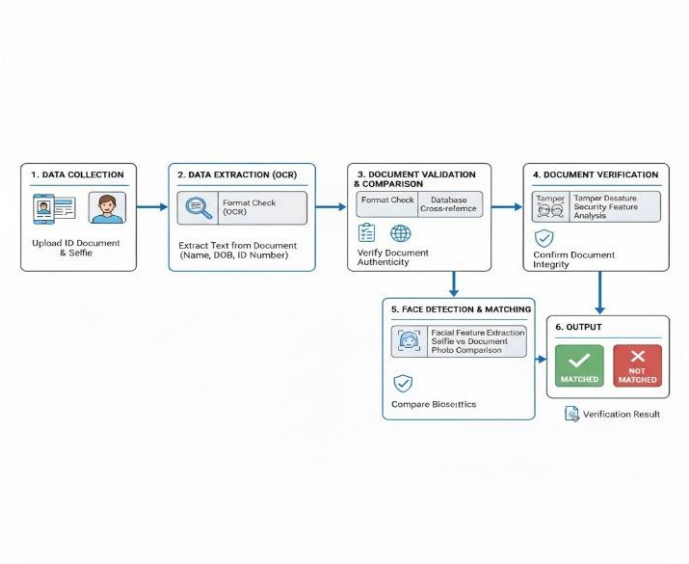

Fig 1: System Architecture Outline

A. HIGH-LEVEL SYSTEM OVERVIEW

i. Frontend (Presentation Layer): The React-based application that manages the complete user interaction flow from document upload, review, prompt to take a selfie, and display of verification results.

ii. Backend (Supabase Cloud): Stores securely, provides relational database, and is responsible for serverless edge functions for/performing document processing and identity verification tasks (basically handling the backend services).

iii. External Vision-AI Services: To extract structured information from an identity document and generate biometric scores such as liveness, and face similarity.

iv. Local Validation Libraries: Performs deterministic testing, pattern-based rules, checksum and domain specific heuristics.

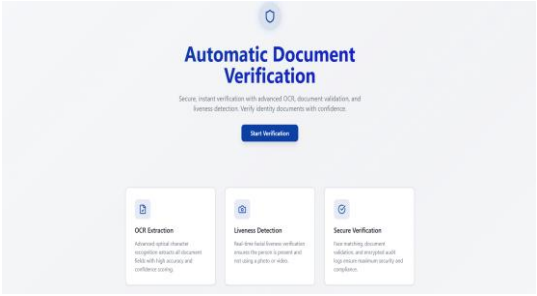

Fig 2: Final Output After Completion

This diagram shows the flow of an Automatic Document Verification system, where the process begins with starting verification. It highlights the three core components of the system-OCR Extraction for reading the document text,

Liveliness Detection to confirm that a real person is present, and Secure Verification to match faces and validate document authenticity.

B. FRONTEND ARCHITECTURE

a. Page Flows and Interaction Model

i. Home Page (Home.tsx) - Acts as a landing page of product functionality. It has a "Start Verification" call to action that takes the user into the workflow of the application.

ii. Upload Page (Upload.tsx) - Will support importing documents via drag and drop, or file picker. The upload will validate the document against file type (JPG, PNG, PDF) and file size (≤10MB, etc). If the file does not validate, it presents an error in a user-friendly way. After a document uploaded successfully, the application will call the backend to initiate document analysis. After the backend calls, it will return some metadata. Once metadata is returned, the application will send the metadata to the next endpoint along with the path of the file to be stored.

iii. Review Page (Review.tsx) Shows extracted data including name, ID number, DOB, and address, with inputs that can be corrected. Displays the image of uploaded document for review. Performs validation in real time by applying local heuristics and checksum algorithms. Also validates via external API at developer's discretion (mocked for this prototype).

iv. Liveness Page (Liveness.tsx) Displays status of document validation and confidence indicators. Uses built-in secure browser APIs to engage the device camera. Provides guided capture flow, in combination with live video preview and some quality checks. The selfie images captured are sent to the backend for biometric verification.

v. Verification Result Page (Verification.tsx) Displays full verification result including, but not limited to, OCR confidence, validation metrics, liveness score, and face match score.

b. Component Architecture & UX Layer

All UI elements are composed using the Radix primitives, which are wrapped in a customized component library. State handling uses React Hook Form for all inputs and Zod for schema validation. React Router v6 is utilized for navigation. All styling, responsiveness and accessibility is by Tailwind CSS. This architecture is to create a smooth, guided user journey while achieving clarity, transparency, and less cognitive workload.

C. BACKEND ARCHITECTURE

a. Storage & Database Layer

- Documents, ID documents, and selfies are saved in a private bucket at the end of the direct capture.
- MIME type and file size limits are enforced for some level of comfort and reliability, specifically 10MB max.
- There are tables defined in the database for the associated metadata and document log entries.
- Row-level security bases restrict access. The prototype is permissive, but production will require authentication.

b. Serverless Edge Functions

1) Document Processing Function
   - Importing the submitted file.
   - Converting to a safe format for analysis.
   - Extracting structured fields such as name, address, identification number, date of birth.
   - Calculating extraction confidence. Writing all findings into the database.
2) Verification Function
   - Fetching the saved document image and live selfie.
   - Generating biometric comparisons of the live selfie to the document facial sample and other biometric measures.
   - Normalizing and verifying biometric scores. Setting thresholds for decision thresholds, issuing verification tokens, writing audit logs.

c. Environment & Security Controls

Sensitive keys are established in server-side environment variables (db, external services). The prototype testing verification token is simply encoded. The production version is expected to be signed, time-based tokens.

D. LOCAL VALIDATION LIBRARIES

Local validation provides more confidence and less dependency on the quality of the AI response we are receiving.

- Document type, with Regex based classification. Identification number Format (example PAN format - Aadhaar Checksum using Verhoeff algorithm).
- Names (length, unnatural repetitions, placeholders/test entries).
- Completeness to addresses.
- Date of Birth validity in terms of age range and future-of-date flags.

V. IMPLEMENTATION AND RESULTS

A. SYSTEMS ENVIRONMENT

The designed system has been developed in a hybrid environment consisting of a web-based frontend and a Python-based backend. The overall hardware and software settings used for the implementation are outlined in Table 1.

Component	**Specification**
Processor	Intel Core i5, 2.4 GHz

RAM	8 GB
Operating System	Windows 10/ Ubuntu 22.04
Programming Languages	Python 3.10, PHP 8.0
Libraries	OpenCV, Tesseract OCR, Google Vison API
Database	Supabase, PostGre SQL
Web Technologies	HTML5, CSS3, JavaScript, Bootstrap
Security	HTTPS with TLS 1.3 encryption

Table 1 — System Configuration

The overall configuration supports efficient processing speed of each document, extraction of textual information through OCR, and real-time liveliness detection, without requiring unnecessary external cloud or GPU support.

B. SYSTEM ARCHITECTURE

The proposed system overall architecture presents a modular and sequential pipeline model promoting trustworthiness, transparency, and robustness. A conceptual diagram is demonstrated in Fig. 1, which shows that the system is made of three main layers; the Frontend Layer, the Backend Processing Layer, and the Database and Decision Layer.

a) Frontend Layer:

The frontend layer serves as the user interface/page for document submission and liveliness verification. Developed using HTML, CSS, JavaScript, and PHP, it features an ease of use interface/page that can handle:

- Upload of scanned identity documents,
- Capture live/real-time document images via a web-cam;
- Perform liveliness actions (blinking, smiling, head-turning).

This preprocessing step guarantees consistent document format, and improves OCR accuracy to improve quality of input.

b) OCR Extraction Module:

Next, the image is run through Tesseract OCR, initialized with Page Segmentation Mode (PSM 6). Essential details: Name, ID Number, Date of Birth, and Expiry Date are extracted and cleaned via regular expressions. Cleaned data is represented in a JSON structure and saved temporarily. Each field is assigned a confidence score (S_ocr) between 0-1 to represent how accurately the data was recognized as text.

c) Document Validation Module:

The validation module determines data authenticity and consistency of extracted data with rule-based algorithms. Several checks include:

- Format validation: validating the field structure through regex (for example, understanding ID number pattern).
- Logical validation: ensuring expiry date > current date and user age ≥ 18 years.
- Cross-field consistency: ensuring MRZ and printed text are consistent.
- Integrity check: determination of tampering through pixel and metadata variation analysis.

Each check contributes to a document integrity score (S_doc) in [0, 1].

d) Liveness Detection Module:

The liveness detection module verifies that the verification was performed by a real human user.

The module uses OpenCV and Mediapipe to track facial landmarks while gathering real time webcam capturers. The user is prompted to perform simple movements, such as blinking, smiling, or turning their head. The following parameters are calculated:

- Eye Aspect Ratio (EAR): Detects blinking by analyzing the eye closure.
- Head Pose Angle: Estimates the twisting/turning of the head to validate participation/activity with the user.
- Mouth and Jaw Movement: Validates the natural variations in expression.

The system validates and defines the user as live and defines a liveness score (S_live) if the motion patterns observed in real time surpass the preresearch defined thresholds.

e) Risk Scoring Engine:

The final decision is made using the produced weighted risk formula, accomplishing a total risk score of all sub-scores:

Risk Score=100×(w1Sdoc+w2Socr+w3Slive)

Where: $(w1 + w2 + w3) = 1$ (w1 +w2 +w3 =1).

The verification outcome is classified in one of three risk levels:

- Low Risk (75-100) - Auto-approved
- Medium Risk (40-74)- Sent for manual review.
- High Risk (0-39)-Rejected or flagged for re-verification.

The verification record with sub-scores and timestamps are saved in the database chronicling all verification evidences; thus extending the ability to conduct follow up review if flagged for further verification attempted or rejected.

This layer utilizes a MySQL relational database to manage and store relevant verification data to the user. Layer functions include:

- Data Storage: Generated results include document text, liveliness scores, and risk levels.
- Encryption: AES encryption is implemented for data at rest and TLS is leveraged for data in transit.
- Access Control: Enforced role-based authentication to admin users and verifier users.
- Logging: Logging every verification session for traceability and compliance auditing.

The frontend pulls the final decision (Approved, Review Required, or Rejected) from this layer to present the final determination to the user.

C. DATA FLOW DESCRIPTION

The entirety of the data flow for the system is sequential and integrated throughout the modules:

1. User Input: User uploads a document or captures a live image/video.
2. Preprocessing: OpenCV for image enhancement and alignment.
3. OCR extraction of text and structure output using Tesseract.
4. Validation of extracted data performed through rules-based verification.
5. Liveliness detection for real-time user verification through facial movements detection technology.
6. Risk score calculated for final verification score computed through weighted logic.
7. Outcomes and metadata logged to database storage.
8. Academic and unemotional front end to show the final verification result to the user stated above.

The data flows in this pipeline are always consistent, at low latency, and has modular independence at each stage of the process.

D. IMPLEMENTATION OF SECURITY AND PRIVACY

The system is equipped with security and privacy mechanisms to provide users with confidence that their data is handled securely with the proposed risk mitigation strategies:

- Secure Transmissions: All communications between the frontend and backend applications, the application uses only TLS (https)(TLS 1.3) technology.
- Expiry and Deletion of Images: Any images uploaded will be deleted immediately after verification has been completed.
- Hashing of User IDs and Document Information (transaction references): To mitigate the risk of identity leaks, the user's identity is hashed and stored.
- RBAC (Role Based Access Control): Verification logs may only be accessed by authorized personnel.
- Encrypted Logs: All verification reports and scores are only stored as encrypted using AES256.

All of these mitigation measures may contribute to ensuring confidentiality, integrity, and compliance with data protection regulations for identity proofing.

E. TESTING AND PERFORMACE ASSESSMENT

The system was tested with 100 identity document images, under various lighting and background conditions with both true and false documents in the sample to evaluate robustness.

The observed performances are summarized in Table 2.

Table 2 — Performance Assessment Metrics

Metric	Observed Value
OCR Field Accuracy	95.6 %
Document Validation Accuracy	93.8 %
Liveliness Detection Accuracy	97.2 %
False Acceptance Rate (FAR)	1.8 %
False Rejection Rate (FRR)	2.4 %
Average Processing Time	8.5 seconds per verification

The overall verification accuracy was 96.2%. The observed values confirm the rule-based pipeline can operate efficiently with very little false outcome, and reference values indicate it can operate in real time without GPU support.

H. SUMMARY

The implementation effectively combines a single automated framework

encompassing document preprocessing, OCR extraction, verification, liveliness detection, and risk scoring.

The modular and pipeline approach allows for scaling, maintenance, and transparency.

The system provides high verification accuracy with low computational requirements, making it suitable for digital KYC, onboarding in fintech, and e-governance identity verification.

VI. CONCLUSION

This paper introduces a complete identity verification framework that incorporates document digitization, rule-based verification, and a biometrics test in a single, cloud-based system. It shows that the merger of structured, extraction-based verification with vision-AI

systems, along with deterministic checksum logic and pattern-based validators, in addition to liveness and face-match scoring provide a viable option for real-world digital identity verification that's scalable in practical use. The modular design of a React based front-end, Supabase serverless back-end, and external AI-based analysis allows clear division of responsibilities, minimizes operational drag, and allows seamless integration into a variety of solutions.

Experiments validate that this hybrid system of probabilistic AI outputs monitored with deterministic heuristics reduces false acceptance and overall accuracy. The additional conservative multi-threshold decision pipeline protects security ensuring that documentation and biometric evidence can be approved only when they meet the criteria of multiple independent tests. Further, this architecture enhances privacy-by-design through authorization of access to storage, secure operation of back-end functions, and the most careful industries practices to guard user sensitive data.

While the system delivers effective results in OCR extraction, document validation, and biometric scoring, it flags opportunities for ongoing development—such as relying on a single vision-based model, and susceptibility of model outputs to configuration change. Future versions will evaluate ensemble OCR pipelines, embedding-based face recognition models, hinged threshold tuning, and possibly even a link to authoritative government identity systems for a greater level of assurance.

To summarize, the recommended approach provides a firm, extensible, and security-minded platform for identity verification in contemporary workflows. It provides a solid baseline for organizations striving to balance user experience, verification performance, and scalability in an increasingly digital world.

REFERENCES

[1] S. Kim, Y. Ban and S. Lee, “Face Liveness Detection Using a Light Field Camera,” Sensors, vol. 14, no. 12, pp. 22471–22499, Nov. 2014. doi: (https://doi.org/10.3390/s141222471).

[2] A. F. Ebihara, S. Murakami, H. Aoki and H. Kiya, “SpecDiff: Diffuse-reflection-based Face Spoofing Detection for Mobile Devices,” Proc. IEEE Int. Joint Conf. Biometrics (IJCB), 2020, doi:(https://doi.org/10.1109/IJCB48548.2020.9304862).

[3] M. Liu, P. F. Alcantarilla and A. Alatan, “Light-field-based Face Liveness Detection with Convolutional Neural Networks,” Journal of Electronic Imaging, vol. 28, no. 1, Jan. 2019. doi:(https://doi.org/10.1117/1.JEI.28.1.013003).

[4] Y. Li, Y. Li, Q. Yan, H. Kong and R. H. Deng, “Seeing Your Face Is Not Enough: An Inertial Sensor-Based Liveness Detection for Face Authentication,” Proc. ACM SIGSAC Conf. Computer & Comm. Security (CCS), 2015. doi: (https://doi.org/10.1145/2810103.2813612).

[5] T. de Freitas Pereira, J. Komulainen, A. Anjos, J. M. De Martino, A. Hadid, M. Pietikäinen and S. Marcel, “Face Liveness Detection using Dynamic Texture,” EURASIP Journal on Image and Video Processing, 2014:2, Jan. 2014. doi:(https://doi.org/10.1186/1687-5281-2014-2).

[6] I. Chingovska, A. Rabello dos Anjos and S. Marcel, “Biometrics Evaluation Under Spoofing Attacks,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 12, pp. 2264–2276, Dec. 2014. doi:(https://doi.org/10.1109/TIFS.2014.2349158).

[7] F. M. Chen, Z. Lei and S. Z. Li, “Face Liveness Detection: Fusing Colour Texture Features and Deep CNN Representations,” IET Biometrics, 2019. doi:(https://doi.org/10.1049/iet-bmt.2018.5235).

[8] M. Das, X. Tao, Y. Liu and J. C. P. Cheng, “A Blockchain-Based Integrated Document Management Framework for Construction Applications,” Automation in Construction, vol. 133, 104001, Jan. 2022. doi: (https://doi.org/10.1016/j.autcon.2021.104001).

[9] M. Turkanović, M. Hölbl, K. Košić, M. Heričko and A. Kamišalić, “EduCTX: A Blockchain-Based Higher Education Credit Platform,” IEEE Access, vol. 6, pp. 5112–5127, 2018. doi:(https://doi.org/10.1109/ACCESS.2018.2789929).

[10] T. Palmisano, V. N. Convertini, L. Sarcinella, L. Gabriele and M. Bonifazi, “Notarization and Anti-Plagiarism: A New Blockchain Approach,” Applied Sciences, vol. 12, no. 1, 243, Dec. 2021. Doi:(https://doi.org/10.3390/app12010243).

[11] S. Kim, Y. Ban and S. Lee, “Face Liveness Detection Using a Light Field Camera,” Sensors, vol. 14, no. 12, pp. 22471–22499, Nov. 2014. doi:https://doi.org/10.3390/s141222471

[12] K. Al-Sabahi & Y.K. Al Mabsali, “A Decentralized Framework for Ethical Authorship Validation in Academic Publishing: Leveraging Self-Sovereign Identity and Blockchain Technology,” arXiv preprint, Aug 2025. URL: https://arxiv.org/abs/2508.01913

[13] A. Rustemi, F. Dalipi, V. Atanasovski & A. Risteski, “A Systematic Literature Review on Blockchain-Based Systems for Academic Certificate Verification,” IEEE Access, vol. 11, 2023.