Digital KYC Verification System Using Hybrid Face-Matching & OCR

By Samriddhi Kumari

1. Problem Understanding

Effective identity verification is a critical requirement for banks, financial institutions, and digital service providers. Traditional KYC processes are often slow, manual, and prone to human errors, leading to customer dissatisfaction and operational delays. Additionally, the rise of digital fraud, forged documents, and impersonation attempts has increased the need for stronger, technology-driven verification mechanisms.

The key problems identified are:

- **Manual document verification** causes delays and increases workload.
- **Poor-quality uploads** (blur, low brightness, unclear text) frequently lead to rejections.
- **Lack of automation** results in inconsistent validation outcomes.
- **Selfie-to-document photo mismatch** is a major security risk.
- **High user drop-off rates** occur due to a lengthy and confusing onboarding flow.

A reliable, automated, and AI-assisted digital KYC solution is required to enhance accuracy, reduce fraud, and improve the overall onboarding experience.

2. Technical Approach

The proposed system follows a structured and data-driven methodology to ensure accuracy, robustness, and operational reliability. The solution integrates OCR-based document analysis, OpenCV-powered image quality checks, and multi-metric face similarity matching to validate user identity with minimal manual intervention.

The technical approach includes:

- **OCR Extraction & Document Type Detection** using Tesseract to classify Aadhaar, PAN, Passport, and Voter ID.
- **Image Quality Analysis** (blur, brightness, sharpness) to ensure valid and readable uploads.
- **Multi-metric Face Matching** using SSIM, ORB features, histogram correlation, and edge pattern similarity.
- **API-driven Architecture** using FastAPI/Flask for modular and scalable integration.

- **Validation Pipeline** to ensure consistent processing, error handling, and standardized output (Approved/Rejected).

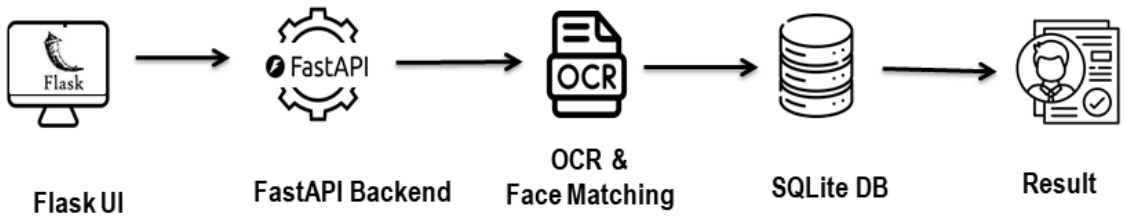

Fig 1: work-flow of the system

Feasibility Assessment:

Technical feasibility:

Table 1: Technology stack

Component	Technology Used	Feasibility Analysis
Backend Framework	FastAPI / Flask	Lightweight, easy to deploy, high performance for APIs. Fully feasible with minimal setup.
Computer Vision	OpenCV	Mature library, cross-platform, supports all required image checks. Highly feasible.
OCR Engine	Tesseract OCR	Open-source, reliable for ID text extraction. Works well with scanned documents.
UI Layer	HTML, CSS, JavaScript	Simple, customizable, requires no heavy frameworks. Feasible for a clean KYC interface.
Database	SQLite	Lightweight, embedded, requires no server. Ideal for storing attempts and logs.
API Testing	Browser	Easily used for testing endpoints.

Table 2: Hardware & System Requirements Feasibility

Requirement	Minimum Specification	Feasibility
Processor	Dual-Core 2.0+ GHz	Easily available on any standard laptop.
RAM	4 GB	Enough for OCR + image processing.
Storage	500 MB	For project files, logs, and temporary images.
GPU	Not required	Increases feasibility—no special hardware needed.
Operating System	Windows / Linux	Cross-platform compatibility ensures feasibility.

Table 3: Software & Resource Requirements

Requirement	Description	Feasibility
Python 3.10–3.12	Required for compatibility with packages	Widely available and easy to install.
pip Packages	FastAPI, uvicorn, OpenCV, pytesseract	All open-source, actively maintained.
IDE	VS Code	Freely available for development.
Tesseract Installation	OCR engine installation	Simple install; fully feasible on Windows/Linux.

Table 4: System Constraints & Considerations

Constraint	Impact	Feasibility Verdict
Image quality dependency	Low-quality uploads may fail OCR or face checks	Manageable with quality filters (blur, brightness).
No deep face recognition	Using similarity metrics instead of embeddings	Acceptable for prototype; scalable later.

Local storage	Temporary file usage	Feasible; can be extended to cloud storage.
Real-time processing	OCR + CV takes 0.5–2 seconds	Feasible within acceptable latency limits.

Economic Feasibility:

The project is economically feasible because it uses open-source tools, requires no licensing cost, and can run on low-cost servers.

Table 5: Economic feasibility

Category	Estimated Cost
Development Tools	₹0 (Open-source)
Developer Time (Student Project)	₹0 (Self-development)
Server hosting (Local or Free-tier cloud)	₹0 – ₹500/month
Hardware (Laptop)	Already available
Maintenance Cost	Very low

Cost–Benefit Overview

- Low development and operational cost due to open-source stack.
- High value addition: automation reduces manual KYC verification cost by ≥70%.
- Scalable for enterprise use, making it economically attractive for fintech and banks.

COCOMO Estimation:

Assumptions:

- Project Size: ~2,000 LOC (Python + HTML/CSS/JS)
- Mode: Organic (small team, simple requirements)

COCOMO Formula

Effort (person-months)

E = 2.4 × (KLOC)^1.05 KLOC = 2 (2000 LOC)
E = 2.4 × (2^1.05)
E ≈ 4.9 person-months

Development Time (in months)

D = 2.5 × (Effort)^0.38
D = 2.5 × (4.9^0.38)
D ≈ 5.6 months

Average Team Size

People = Effort / Time
≈ 0.85 persons
(Meaning: ~1 developer can complete it in 5–6 months.)

3. Innovation & Solution Design

The proposed solution is designed to be lightweight, scalable, and highly user-centric. It employs a structured integration of a Flask-based frontend with a FastAPI backend, enabling rapid processing and modular development. The system utilizes a hybrid face-matching methodology built entirely on OpenCV and statistical similarity metrics. This ensures efficient performance even in low-resource environments. The user interface as shown in fig 2 provides a streamlined workflow that includes document upload, selfie capture, real-time verification, and transparent result reporting. Owing to its simplicity, efficiency, and reliability, the solution is well-suited for deployment in financial institutions, banking systems, and other identity verification processes. Project flow is shown in fig 1.

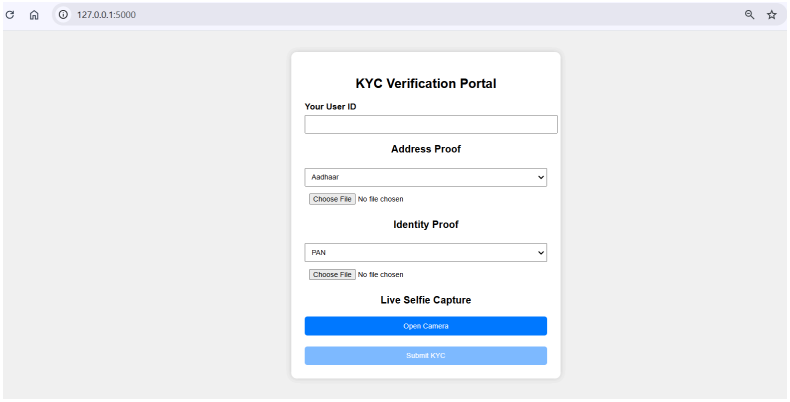

Fig 2: frontend / User interface

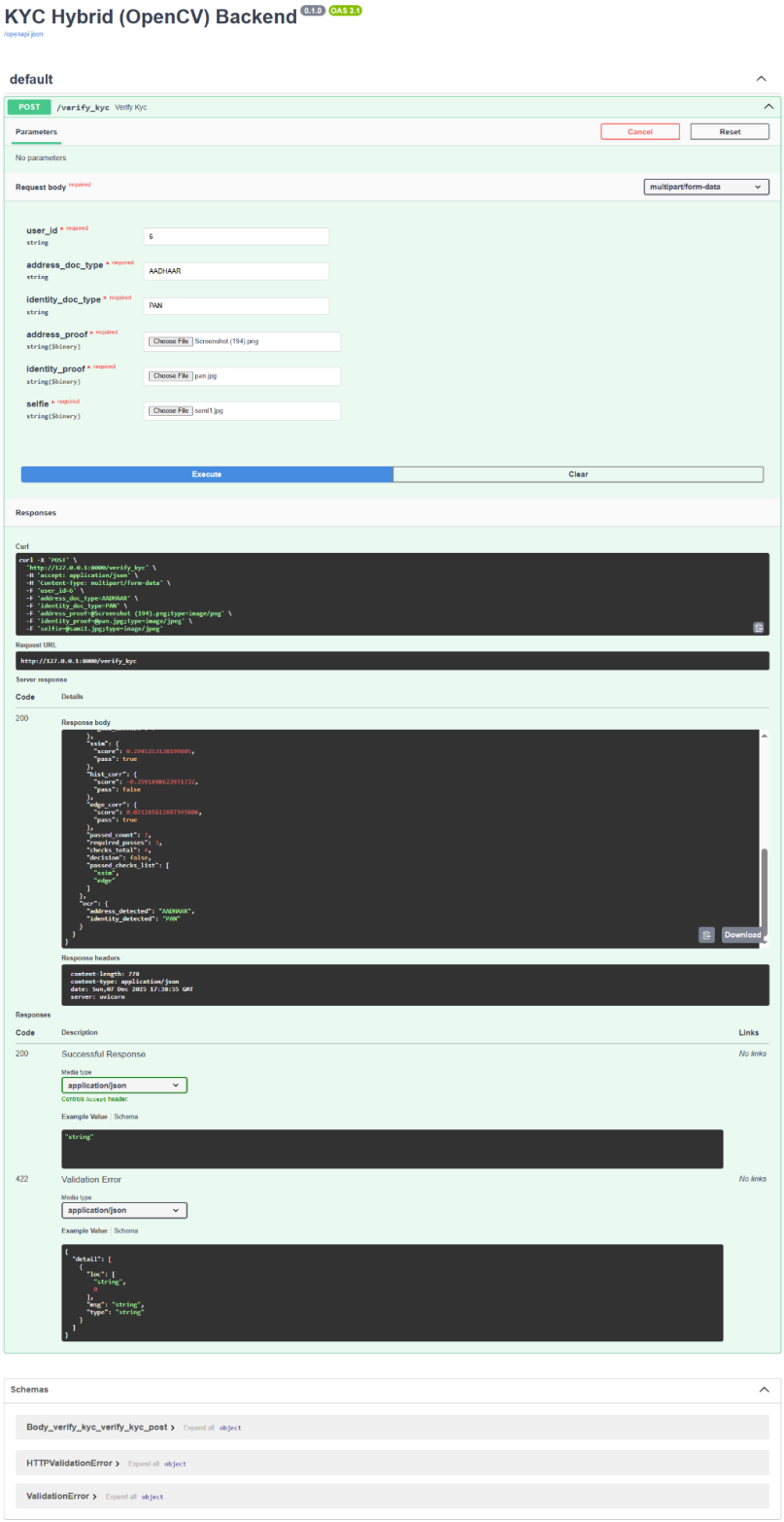

Fig 3: Backend - fastAPI

4. Future Enhancements

- To improve accuracy, usability, and scalability, several enhancements can be implemented in future versions:
- ML-based Face Recognition: Replace OpenCV heuristics with deep-learning models for higher verification accuracy.
- Improved OCR Pipeline: Use advanced OCR engines and noise reduction for better text extraction.
- Cloud Deployment: Host the backend on AWS/GCP with CI/CD automation.
- Role-Based Dashboard: Provide admin analytics, verification history, and user monitoring.
- Multi-language Support: Enable OCR and interface support for multiple Indian languages.

5. Conclusion:

The Digital KYC Verification System successfully automates identity verification using OCR and hybrid face-matching techniques. Its lightweight architecture, fast processing, and user-friendly workflow make it suitable for real-world banking and fintech use. The solution enhances security, reduces manual effort, and demonstrates strong potential for scalable deployment.