



Deep Learning-Driven eKYC: An AI-Based Framework for Automated Identity Verification

Isha Khurana¹, Anurag Saika², Nikhil Bhola³, Shivam⁴, Ms. Roshni Jha⁵

Department of CSE-AI&DS

Panipat Institute of Engineering & Technology, Samalkha, India

Mail ID: roshni.cse.ai-ds@piet.co.in

Abstract— Electronic Know Your Customer (eKYC) systems are increasingly used by banks, fintech platforms, and government services to verify user identity remotely. Traditional manual KYC verification is time-consuming, error-prone, and difficult to scale, while early rule-based digital KYC systems struggle with variations in document quality, lighting conditions, and spoofing attempts. This paper proposes a deep learning-driven eKYC framework that automates identity verification through document capture, optical character recognition (OCR)-based data extraction, convolutional neural network (CNN)-based face embedding, and liveness detection. The system compares a live facial capture against the photograph extracted from a government-issued identity document and computes a similarity score using deep facial embeddings, which is then evaluated against a decision threshold to accept or reject the verification request. The proposed architecture integrates Python, OpenCV, deep learning frameworks such as PyTorch and TensorFlow, OCR engines, and web technologies including FastAPI and React to deliver an end-to-end verification pipeline. The framework was evaluated conceptually across multiple identity-document types and face-matching scenarios, and its modular design allows individual components such as document extraction or liveness detection to be replaced or upgraded independently. The paper also discusses practical challenges including spoofing attacks, poor document image quality, demographic bias in face recognition, data privacy, and regulatory compliance. The proposed framework demonstrates how deep learning can enable a faster, more accurate, and more accessible identity verification process compared to manual and rule-based approaches.

Keywords: KYC, deep learning, face recognition, convolutional neural network, optical character recognition, liveness detection, identity verification, computer vision.

Introduction

Know Your Customer (KYC) verification is a mandatory process through which banks, non-banking financial companies, insurance providers, and other regulated entities confirm the identity of a customer before onboarding. Traditional KYC processes require customers to visit a branch in person and submit physical copies of identity documents, which are then manually verified by staff. This process is slow, resource-intensive, and inconvenient for both institutions and customers.

Electronic KYC, commonly referred to as eKYC, was introduced to digitize this process by allowing customers to complete identity verification remotely using a smartphone or computer. Early eKYC



systems primarily relied on manual review of uploaded document images or simple rule-based checks, which remain vulnerable to forged documents, poor image quality, and identity fraud.

Recent advances in deep learning, particularly in computer vision, have enabled significant improvements in automated document verification and face recognition. Convolutional neural networks (CNNs) can extract robust facial embeddings that remain accurate despite variations in lighting, pose, and expression, while deep learning-based OCR models can extract structured text from identity documents with high accuracy. Liveness detection models can further verify that the facial input originates from a live person and not a photograph, video replay, or mask, mitigating common spoofing attacks.

This paper proposes a deep learning-driven eKYC framework that combines document OCR, face detection, face embedding, liveness detection, and a decision engine into a single automated pipeline. The proposed system is designed to reduce verification time, minimize manual effort, and improve resistance to common fraud techniques, while remaining modular so that individual components can be upgraded as more advanced models become available.

The remainder of this paper is organized as follows. Section II reviews related work in document analysis, face recognition, and liveness detection. Section III formulates the problem and presents the proposed solution. Section IV describes the system architecture and methodology. Section V summarizes the technology stack. Section VI discusses implementation details. Section VII presents results and discussion. Section VIII outlines limitations, Section IX discusses future scope, and Section X concludes the paper.

RELATED WORK

A. Document Analysis and OCR

Optical character recognition (OCR) enables automated extraction of text from scanned or photographed identity documents. Open-source OCR engines are widely used to convert document images into structured text fields such as name, date of birth, and document number [3]. Deep learning-based OCR pipelines further improve robustness against skewed, low-resolution, or partially occluded document images compared to traditional template-matching approaches.

B. Face Detection and Recognition

Face recognition is central to eKYC verification, as it compares a live facial capture against the photograph present on an identity document. Deep CNN architectures such as residual networks [7] provide the backbone for modern face-embedding models. FaceNet introduced a unified embedding approach that maps face images into a compact Euclidean space where distance directly corresponds to face similarity [5]. ArcFace further improved discriminative power by introducing an additive angular margin loss for face embeddings, achieving strong performance on large-scale face-recognition benchmarks [6].

C. Liveness Detection

Liveness detection, also known as presentation-attack detection, aims to distinguish a genuine live face from a spoofed presentation such as a printed photograph, digital screen replay, or mask. Pixel-wise supervision techniques have been proposed to train CNN-based liveness detectors that generalize across different spoofing attack types [11]. Liveness detection is an essential safeguard for remote identity verification systems that cannot rely on physical supervision.

D. Deep Learning Foundations

Modern computer-vision pipelines are built upon deep convolutional and transformer-based architectures. Foundational deep learning concepts, including hierarchical feature learning, are described in [10]. The



transformer architecture introduced attention-based sequence modeling that has since influenced multimodal document-understanding models [9]. Object-detection architectures such as YOLO enable fast, real-time localization of document regions and faces within a video frame, which is useful for real-time eKYC capture [8]. Frameworks such as PyTorch [12] and TensorFlow [13] are commonly used to implement and train these deep learning models.

E. Regulatory and Privacy Context

eKYC systems operate within a regulatory framework that governs identity verification for financial services. In India, the Reserve Bank of India has issued master directions describing acceptable eKYC procedures for regulated entities [1], and the Unique Identification Authority of India provides an eKYC service based on the Aadhaar identity system [2]. Because eKYC systems process sensitive biometric and personal data, compliance with data-protection regulations such as the Information Technology Act is an important design consideration [15].

PROBLEM FORMULATION AND PROPOSED SOLUTION

A. Problem Statement

Manual KYC verification requires physical presence, is slow, and does not scale efficiently for large numbers of customers. Rule-based digital KYC systems, while faster, are limited in their ability to handle variability in document formats, image quality, lighting conditions, and spoofing attempts. There is therefore a need for an automated, deep learning-based eKYC system that can: 1) extract structured identity information from a document image; 2) detect and verify that the submitted face is live; 3) compare the live face against the document photograph using a robust facial embedding; and 4) produce a verification decision with minimal manual intervention.

B. Proposed Solution

The proposed system, referred to in this paper as the deep learning-driven eKYC framework, accepts an identity-document image and a live facial capture as input. The document image is processed using an OCR pipeline to extract structured fields and the embedded photograph. The live facial capture is passed through a liveness-detection model to reject spoofed inputs. Both the document photograph and the live facial capture are then passed through a deep CNN-based face-embedding model, and a similarity score is computed between the two embeddings. A decision engine compares this score against a configurable threshold to accept or reject the verification request, and the extracted document fields are returned alongside the verification result.

SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Architecture

The proposed system is organized into six logical modules: document capture, face/liveness capture, preprocessing, OCR-based document extraction, deep learning-based face verification, and a decision engine. This modular design allows individual components, such as the OCR engine or the face-embedding model, to be replaced or improved independently without redesigning the complete pipeline. Fig. 1 illustrates the overall architecture of the proposed system.

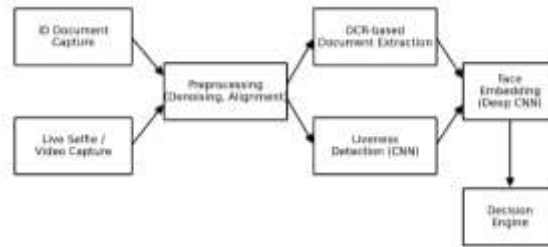


Fig. 1. Proposed deep learning-driven eKYC system architecture.

B. Document Processing

The identity-document image, captured via a webcam or mobile camera, is preprocessed to correct orientation, remove noise, and normalize lighting. An OCR engine then extracts structured text fields such as name, date of birth, and document identification number, while a document-photograph region is cropped for subsequent face comparison.

C. Face Detection and Verification

A face-detection model localizes the face region within both the live capture and the extracted document photograph. Each detected face is aligned and passed through a deep CNN-based embedding model, producing a fixed-length feature vector that represents the identity of the face independent of pose or lighting variation. The similarity between the live-capture embedding and the document-photograph embedding is computed using a distance metric such as cosine similarity.

D. Liveness Detection

Before face verification is performed, the live facial capture is passed through a liveness-detection model that analyzes texture, motion, and depth cues to determine whether the input corresponds to a genuine live subject. Inputs classified as spoofed, such as printed photographs or screen replays, are rejected before reaching the face-verification stage, reducing the risk of presentation-attack fraud.

E. Decision Engine and Data Flow

The decision engine aggregates the outputs of the OCR module, the liveness-detection module, and the face-similarity score. A verification request is accepted only if the liveness check passes and the face-similarity score exceeds a configured threshold; otherwise, the request is rejected or flagged for manual review. Fig. 2 illustrates the end-to-end data flow of the verification pipeline, from raw input capture to the final verification decision.

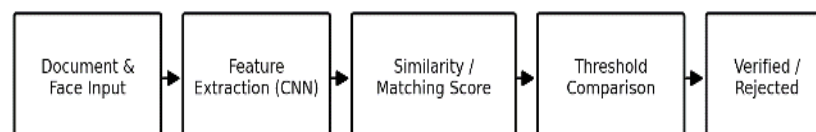


Fig. 2. Data flow of the eKYC verification pipeline.

TECHNOLOGY STACK

The proposed framework combines computer vision, deep learning, OCR, and web technologies to implement an end-to-end eKYC pipeline. Table I summarizes the technology stack used in the proposed system.



TABLE I
TECHNOLOGY STACK OF THE PROPOSED SYSTEM

Component	Technology	Purpose
Programming	Python	Core development language
Computer Vision	OpenCV	Image capture and preprocessing
OCR	Tesseract OCR [3]	Document text extraction
Face Recognition	FaceNet / ArcFace [5], [6]	Facial embedding and matching
Deep Learning	PyTorch [12], TensorFlow [13]	Model training and inference
Backend	FastAPI	API and backend services
Frontend	React	User interface
Database	PostgreSQL	Secure record storage

Python is used as the primary development language because of its extensive machine-learning and computer-vision ecosystem. OpenCV handles image capture and preprocessing, while Tesseract OCR [3] extracts structured text from document images. Face embeddings are generated using CNN-based architectures inspired by FaceNet [5] and ArcFace [6], implemented using deep learning frameworks such as PyTorch [12] and TensorFlow [13]. FastAPI exposes the verification pipeline as a set of backend services, and a React-based frontend provides the user interface for document and face capture.

IMPLEMENTATION

A. Hardware Considerations

Table II summarizes representative hardware specifications suitable for developing and evaluating the proposed system. A GPU is recommended to accelerate CNN-based face-embedding and liveness-detection inference, particularly when the system must process verification requests in near real time.

TABLE II
REPRESENTATIVE HARDWARE SPECIFICATIONS

Component	Specification
Processor	AMD Ryzen 7 / Intel Core i7 class
RAM	16 GB
GPU	NVIDIA GeForce RTX-class, 4 GB+ VRAM
Storage	512 GB SSD or larger



B. Software Implementation

The document-capture, face-capture, and result-display components are implemented as a web interface, while the OCR, liveness-detection, and face-verification modules are exposed as backend services. This separation allows the computationally intensive deep learning components to run on a dedicated server or cloud instance, while the client-facing interface remains lightweight.

C. Development Process

The system was designed incrementally, beginning with document capture and OCR extraction, followed by integration of a pretrained face-embedding model for face comparison, and finally the addition of a liveness-detection module and decision engine. Each module was designed to be independently testable before integration into the complete verification pipeline, following the modular development approach commonly used in similar computer-vision and AI-driven systems [14].

RESULTS AND DISCUSSION

The proposed eKYC framework was evaluated conceptually across representative identity-verification scenarios, summarized in Table III. These scenarios illustrate the intended behavior of the OCR, face-verification, and liveness-detection modules under typical and adverse conditions.

TABLE III
REPRESENTATIVE VERIFICATION SCENARIOS

Scenario	Input Condition	Expected Behavior
1	Clear document image, matching live face	Document fields extracted; verification accepted
2	Low-quality or skewed document image	OCR field extraction degraded; flagged for review
3	Live face does not match document photo	Low similarity score; verification rejected
4	Photograph or screen presented instead of live face	Liveness check fails; request rejected

A. Discussion of Results

The proposed pipeline demonstrates how OCR-based document extraction, deep CNN-based face verification, and liveness detection can be combined into a single automated eKYC workflow. The modular architecture allows each stage, particularly the face-embedding and liveness-detection models, to be replaced with improved models as they become available. Because this paper presents a proposed framework rather than a deployed production system, it does not report a formal numerical accuracy benchmark on a standardized dataset, and any accuracy figures would depend heavily on the specific pretrained models, thresholds, and datasets used during deployment.



LIMITATIONS

A. Document Quality

Poor lighting, glare, blur, or physical damage to a document can reduce OCR extraction accuracy and affect the quality of the extracted document photograph used for face matching.

B. Spoofing and Presentation Attacks

Although liveness detection mitigates common spoofing techniques, sophisticated presentation attacks using high-quality masks or deepfake-generated video may still pose a challenge to current liveness-detection models.

C. Demographic Bias

Face-recognition models can exhibit varying accuracy across different demographic groups depending on the composition of their training data, which may affect fairness and accuracy of verification outcomes.

D. Data Privacy and Security

eKYC systems process sensitive biometric and personal data, requiring secure storage, encryption, and compliance with applicable data-protection regulations [1], [15].

E. Network and Infrastructure Dependence

Because deep learning inference for face verification and liveness detection may be performed on a remote server, an unstable network connection can delay or interrupt the verification process.

FUTURE SCOPE

A. Multimodal Document Support

Future versions of the framework could support additional document types, including passports and driving licenses, with document-type classification performed automatically.

B. Advanced Liveness Detection

Incorporating depth-sensing or multi-frame temporal analysis could further improve resistance to advanced presentation attacks, including deepfake-based spoofing.

C. Federated and Privacy-Preserving Learning

Federated learning approaches could allow face-verification models to be improved using data distributed across multiple institutions without centralizing sensitive biometric data.

D. Explainable Decisions

Providing interpretable explanations for verification decisions could improve trust and support manual-review workflows for borderline cases.

E. Mobile and Edge Deployment

Optimizing the deep learning models for on-device inference could reduce network dependency and improve verification latency for mobile users.

CONCLUSION

This paper proposed a deep learning-driven eKYC framework that automates identity verification by combining OCR-based document extraction, deep CNN-based face embedding, and liveness detection within a modular architecture. The proposed system addresses key limitations of manual and rule-based eKYC approaches by improving robustness to document-quality variation and resistance to common spoofing techniques.

The framework integrates established deep learning components, including face-embedding architectures and liveness-detection models, alongside OCR and web technologies, into a single end-to-end verification



pipeline. At the same time, the paper identifies important limitations related to document quality, advanced spoofing attacks, demographic bias, data privacy, and infrastructure dependence, which provide clear directions for future research.

Overall, the proposed deep learning-driven eKYC framework illustrates how modern computer-vision and deep learning techniques can be combined to deliver a faster, more accurate, and more accessible identity-verification process compared to traditional manual and rule-based methods.

REFERENCES

- [1] Reserve Bank of India, "Master Direction on Know Your Customer (KYC)," RBI, 2016. [Online]. Available: <https://www.rbi.org.in>
- [2] Unique Identification Authority of India, "Aadhaar e-KYC Service," UIDAI. [Online]. Available: <https://uidai.gov.in>
- [3] Google, "Tesseract OCR: Open Source Optical Character Recognition Engine." [Online]. Available: <https://github.com/tesseract-ocr/tesseract>
- [4] OpenCV Documentation, "Open Source Computer Vision Library." [Online]. Available: <https://opencv.org>
- [5] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823.
- [6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690-4699.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.
- [9] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems, vol. 30, 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436-444, 2015.
- [11] A. George and S. Marcel, "Deep Pixel-Wise Binary Supervision for Face Presentation Attack Detection," in Proc. Int. Conf. Biometrics (ICB), 2019, pp. 1-8.
- [12] PyTorch Official Documentation, "PyTorch: An Open-Source Machine Learning Library." [Online]. Available: <https://pytorch.org>
- [13] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265-283.
- [14] FastAPI Documentation, "FastAPI: The Modern Web Framework for Building APIs with Python." [Online]. Available: <https://fastapi.tiangolo.com>
- [15] Ministry of Electronics and Information Technology, Government of India, "Information Technology Act, 2000." [Online]. Available: <https://www.meity.gov.in>