A Facial Biometric Architecture for Pension Administration with Explicit Verification States and Auditable Record Linkage

Amiekwe Stephen E

Computer science department, Alex Ekwueme Federal University, Ebonyi State, Nigeria

DOI: https://doi.org/10.51584/IJRIAS.2026.11100042

Received: 21 October 2026; Accepted: 26 October 2026; Published: 04 November 2026

ABSTRACT

A facial recognition component becomes useful to pension administration only when its output is connected to a trusted record and a reviewable decision. This study develops a modular architecture for a MATLAB-based pensioner-verification prototype in Enugu North, Nigeria. The method combines analysis of the acquisition and matching callbacks, a formal state model, a record-linkage schema and an evidence assessment of the existing prototype. Camera capture, image processing, feature visualization and ranked-image retrieval are identifiable in the implementation. A local-feature callback attempts to index 20 returned feature pairs before displaying a verification message; its behavior therefore contains a match-count boundary and an exception-dependent decision. The underlying neural-model test matrix records 102 correct classifications among 105 observations, while a separate application table contains 18 True-coded entries among 20 cases. These recognition results are treated as component evidence rather than measurements of database or pension-payment performance. The architecture introduces explicit capture, matching, review and audit states, with stable person identifiers linking templates and service records. The study contributes a formal integration model and a traceable assessment of implemented versus proposed functions. The result is a technically specified route from a recognition prototype to an accountable verification service, with calibration, transactional testing and a monitored institutional pilot required before operational adoption.

Keywords: biometric information system, pension administration, facial verification, state model, record linkage, software architecture

INTRODUCTION

Pension verification is an information-system task with a biometric component. The service must establish a claimant's relationship to a personnel record, preserve the evidence used and support correction when either the image or the record is uncertain. A classifier alone cannot maintain this relationship. The surrounding software determines whether a candidate match is presented for review, silently accepted or incorrectly stored as an administrative fact.

The Enugu North prototype was developed to support civil-service pension management using facial images and a MATLAB interface. It includes image capture, feature processing and recognition demonstrations. Its architecture also envisages staff details and a centralized database. The engineering question is how to connect these functions without allowing capture errors, nearest-image retrieval or software exceptions to substitute for identity decisions.

Biometric verification tests a claimed identity, whereas gallery identification returns an identity candidate (Jain et al., 2004). A practical service may use both, but its records should make clear which task occurred. This is especially important when different callbacks employ different matching operations. An interface can display a convincing visual match while the code path used for that display differs from the neural model that produced the reported evaluation.

The present paper addresses the integration problem using the original prototype as a case study. Its two contributions are (1) a formal architecture combining verification states, person-to-template linkage and audit

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

events, and (2) a code-grounded assessment that separates visible acquisition and matching behavior from the additional controls needed in a service deployment. The system is evaluated through documented implementation behavior and existing recognition evidence; no unmeasured financial saving or pension-payment improvement is claimed.

LITERATURE REVIEW

Recognition components in an administrative service

Recognition pipelines comprise acquisition, representation and matching, but the output must be interpreted in the context of an application. Wang and Deng (2021) place representation and evaluation protocols within the development of face recognition. FaceNet and ArcFace show how learned facial representations can support matching (Schroff et al., 2015; Deng et al., 2019). Their relevance to architecture is that the matcher should be a replaceable component with a defined interface, rather than an inseparable part of the personnel database.

The case-study pipeline uses Gaussian processing and HOG features. These are established image-processing and representation operations (Getreuer, 2013; Dalal & Triggs, 2005). They do not themselves specify how an administrator should act on a candidate. Similarly, the quality-aware objectives in MagFace and AdaFace motivate measuring input quality, but a low-quality capture should remain distinct from a negative identity decision (Meng et al., 2021; Kim et al., 2022).

Security and traceability

A facial template is sensitive information even when the raw photograph is not displayed. Sun and Liu (2025) examine privacy across data generation, inference and storage, motivating controlled access and traceable handling of template records. Presentation-attack detection is a separate component: a high similarity score does not show that the input came from a live claimant rather than a printed image or replay (Sharma & Selwal, 2023; Yu et al., 2023).

The operational gap is therefore broader than classification accuracy. A service needs an explicit decision boundary, stable record identifiers, processing-status codes and review history. The present architecture makes these responsibilities visible while retaining the original prototype's implemented acquisition and image-processing functions. Table 1 links each research concern to a system requirement.

Table 1 Research basis for the integration requirements

Concern	Literature or technical basis	Architectural consequence
Task definition	Jain et al. (2004)	Store claimed and candidate identities separately
Capture quality	Meng et al. (2021); Kim et al. (2022)	Distinguish unusable images from nonmatches
Ranked retrieval	MathWorks (n.d.-b)	Treat the first returned image as a candidate
Presentation attacks	Sharma and Selwal (2023)	Add independently evaluated liveness controls
Template privacy	Sun and Liu (2025)	Limit access and record template versions

METHODOLOGY

Engineering case-study design

The analysis uses the original system-design description, the MATLAB graphical-interface callbacks, the interface screenshot and the development and application results. The material identifies a facial-image collection in Enugu North over 2018–2021 and a software environment including MATLAB, image-processing

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

facilities, neural-network tools and database technologies. The application architecture is analyzed as a prototype, with proposed operational controls distinguished from visible software functions.

Three methods were used. First, the information flow was decomposed into acquisition, representation, matching, record association and administrative review. Second, the visible callbacks were examined to identify their inputs, successful paths and exception behavior. Third, a state model and normalized record structure were formulated to make the transition from technical evidence to administrative action explicit. This is a design and implementation analysis, not a new recognition experiment.

Figure 1 presents the proposed integration architecture. The acquisition and image-processing stages correspond to the implemented prototype; the separation of review, audit and administrative authorization specifies the surrounding service design. A stable identifier connects the technical result to the personnel record.

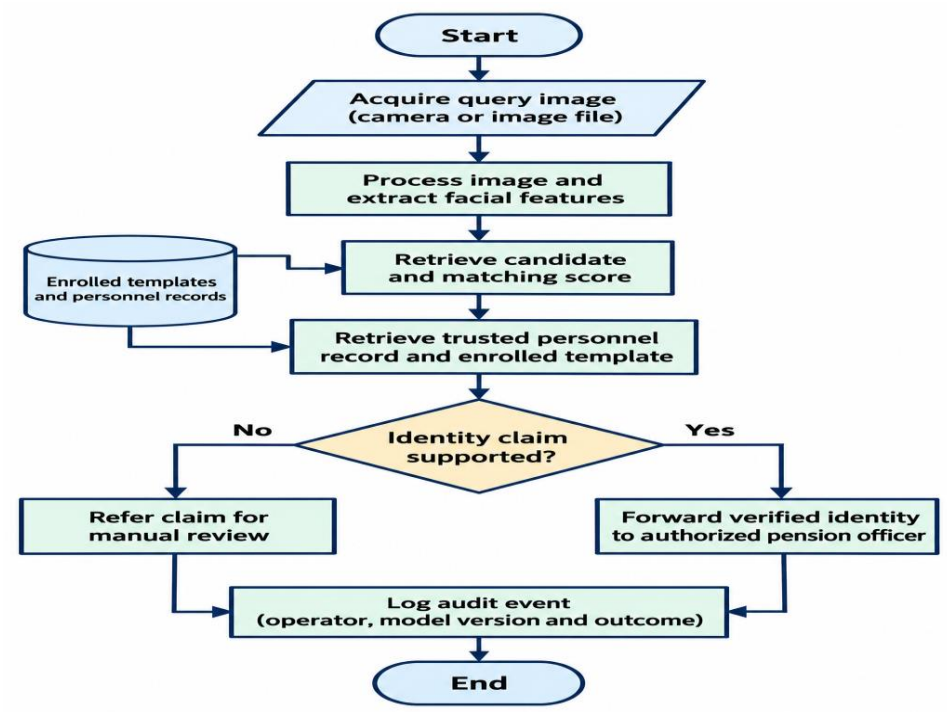


Figure 1. Integration architecture for pension verification. Review, audit and record-control functions are design specifications; their production operation was not measured in the original prototype.

Formal verification state model

Define the verification service as a labelled transition system in Equation (1), where S is the set of states, E is the set of processing events and δ maps a current state and event to a subsequent state.

$$V = (S, E, \delta, s_0), \quad \delta: S \times E \rightarrow S \tag{1}$$

The state set is S = {registered, captured, unusable, processed, candidate, verified, referred, closed}. The initial state s_0 is registered, meaning that an episode has been linked to a claimed personnel identifier. A capture error produces an unusable state; successful processing produces a candidate state. Only a reviewed and supported identity claim enters verified. Figure 2 summarizes the principal decision flow, including recapture and unresolved-claim routes.

$$\delta(\text{captured}, \text{quality_fail}) = \text{unusable}; \tag{2}$$

$$\delta(\text{candidate}, \text{supported_claim}) = \text{verified} \tag{3}$$

Equations (2) and (3) prevent a technical failure from being interpreted as an adverse entitlement decision. In this model, verified means that identity evidence supports the claim; it does not mean that payment has been

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

authorized. The transition policy is proposed as part of the architecture and is not described as an already deployed control.

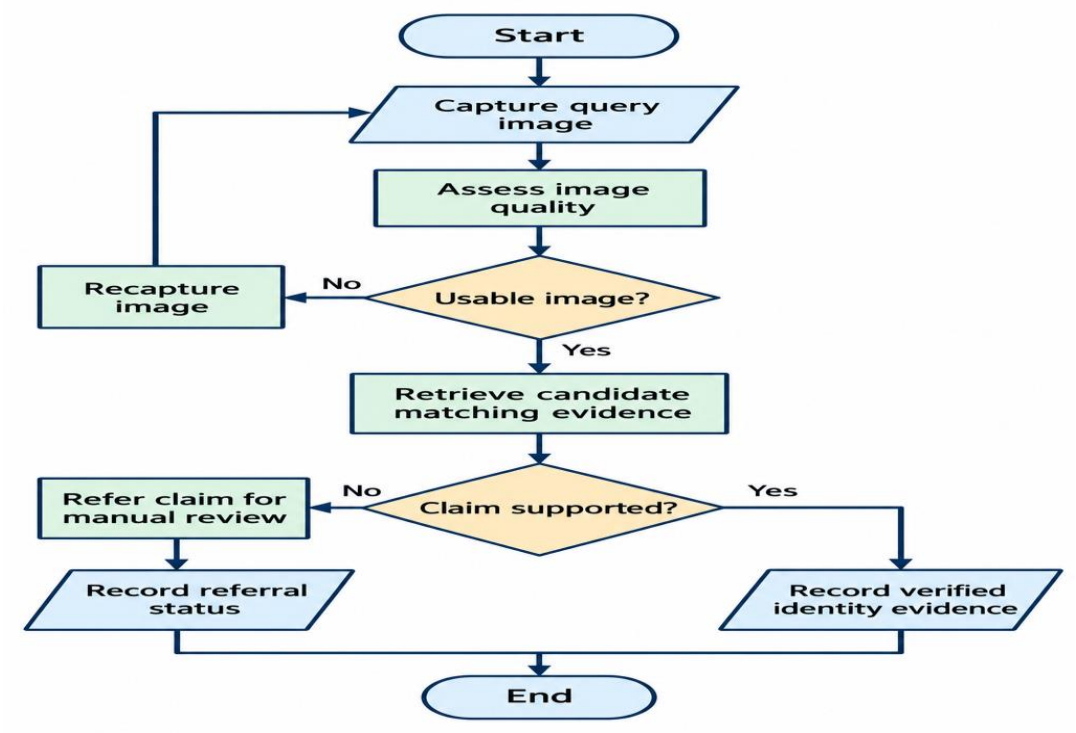


Figure 2. Proposed verification flow with Yes and No branches. Capture quality and support for an identity claim are separate decisions.

Matching interface and administrative action

A matching service can expose a structured output o containing the claimed identifier r, candidate identifier c, score s, quality status q, model version m and event time t. The schema in Equation (4) makes the distinction between technical evidence and the administrative record explicit.

$$o = (r, c, s, q, m, t) \quad (4)$$

For a score-calibrated implementation, a claim-support indicator can be expressed by Equation (5). The quality condition q_ok must hold, the candidate must correspond to the claimed record and the score must reach a threshold τ. This rule formalizes a proposed service policy; the original ranked-image callback does not implement this complete rule.

$$a = 1\{q_{ok} \wedge c = r \wedge s \geq \tau\}; \quad a = 0 \rightarrow \text{review} \quad (5)$$

The administrative action additionally depends on the personnel record and authorized review. Let v denote supported identity evidence, e denote eligibility established through the administrative process, and u denote approval by an authorized officer. Equation (6) expresses the separation of these conditions.

$$\text{payment_authorization} \Rightarrow v \wedge e \wedge u \quad (6)$$

This is a logical requirement for the proposed architecture, not an inferred legal rule or a tested pension-payment mechanism. It prevents recognition success from being used as a substitute for the institution's eligibility and authorization procedures.

Record linkage and audit structure

Figure 3 provides a minimal relational design. Person holds the stable identifier and personnel-record reference. Template holds the enrolled facial representation and model version. VerificationEpisode records a claimed

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

identity and episode status. AuditEvent records the operator, time and technical or review event. One person can have multiple templates and multiple service episodes, and each episode can have multiple audit events.

The separation supports template replacement without changing the person identifier, and it preserves a history of capture retries without recording them as different pensioners. Foreign-key constraints enforce the relationship between a template and a person. Status changes should be committed with their corresponding audit event so that a visible decision and its review record cannot diverge. The schema is a proposed normalized design; the original appendix does not establish a production database transaction implementation.

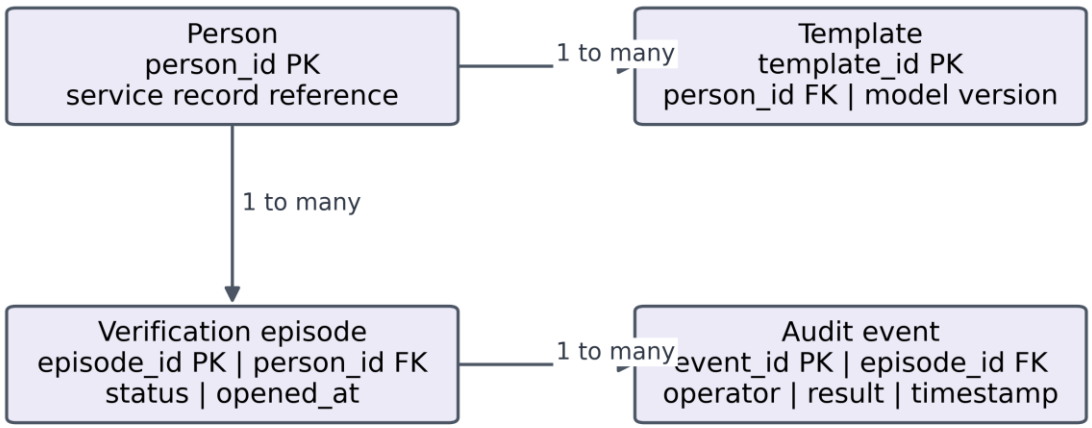


Figure 3. Minimal proposed record structure. PK denotes a primary key and FK denotes a foreign key. Cardinalities specify relationships rather than measured database activity.

Code-path assessment

The acquisition callback creates a face detector and point tracker, reads camera frames and displays tracking output. A separate image-retrieval path constructs an index, invokes retrieveImages and selects the first returned identifier. The documented function returns images in descending similarity order (MathWorks, n.d.-b); the first result is therefore a candidate, not an independently calibrated verification decision.

Another callback detects local features, extracts descriptors and calls matchFeatures, which returns the indices of matching features (MathWorks, n.d.-a). The code then selects indexPairs(1:20, :) within a try/catch block before displaying a verification message. Let M be the number of returned feature pairs. Conditional on no other error, the indexing condition is Equation (7).

$$\text{indexable} = 1\{M \geq 20\} \tag{7}$$

If $M < 20$, indexing the first 20 pairs raises an exception. If $M \geq 20$, indexing can succeed, but the original logic still lacks an explicit independent check of the claimed identity. Exceptions from other operations can also produce the nonverification message. Equation (7) therefore describes a boundary in the visible callback, not a biometric operating threshold.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

RESULTS AND DISCUSSION

Implemented interface and functional evidence

Figure 4 reproduces the interface panel from the original study. It shows an acquisition display, processing controls and a facial-recognition action. The screenshot is evidence of the interface design; the presence of a control does not demonstrate its measured accuracy, runtime or production reliability. No participant photographs are included in this panel.

The strongest directly traceable software functions are camera access, image loading, feature extraction, visualization and ranked-image display. Table 2 separates these functions from the additional service-level responsibilities introduced by the architecture. This distinction helps identify the next implementation work without attributing unobserved database or security behavior to the prototype.

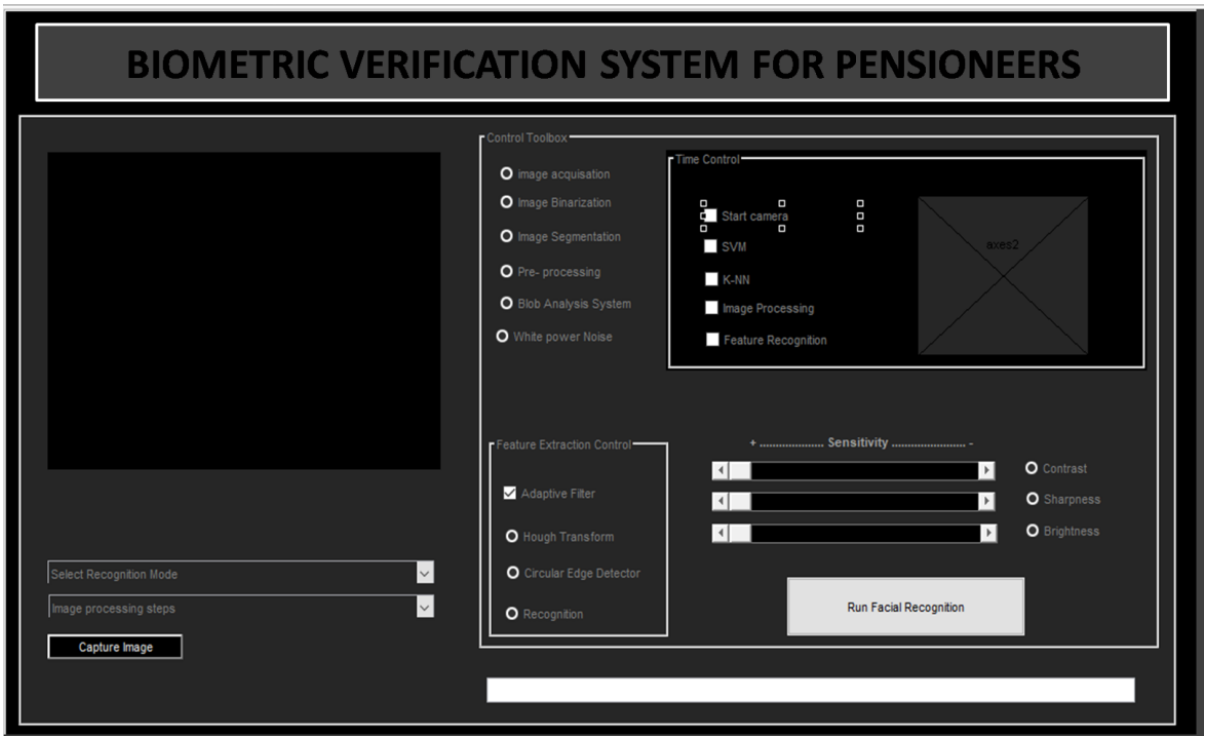


Figure 4. Original prototype interface, cropped to the application panel. Control labels are retained as recorded; no redesigned interface is presented as an experimental result.

Table 2 Traceability of system functions

Function	Implementation evidence	Interpretation
Capture and tracking	Camera, cascade detector and tracker callbacks	Bounded acquisition and display routine
Feature extraction	HOG and local-feature operations	Numerical image representation and visualization
Gallery retrieval	Index construction and first-ranked image	Candidate generation
Local-pair comparison	Selection of the first 20 returned pairs	Index-count condition inside exception handling
Neural classification	Original partition confusion matrices	Component-level two-class performance

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

Database and review	Narrative design and proposed schema	Requires transactional and workflow verification

Match-count boundary and decision semantics

Figure 5 shows the deterministic indexing boundary implied by Equation (7). Counts from 0 through 19 do not support indexing the first 20 pairs, whereas counts of 20 or more do. The graph is a logical analysis of the code condition, not a new image-matching experiment. Table 3 illustrates the boundary with four input counts.

The result has a practical consequence: the software message depends partly on whether an array access succeeds. A message reached through this path cannot be interpreted as a calibrated estimate that a claimant is the enrolled pensioner. Moreover, the catch branch merges insufficient matches with other runtime errors. A revised implementation should report processing failures separately and reserve identity decisions for a score- and claim-based rule.

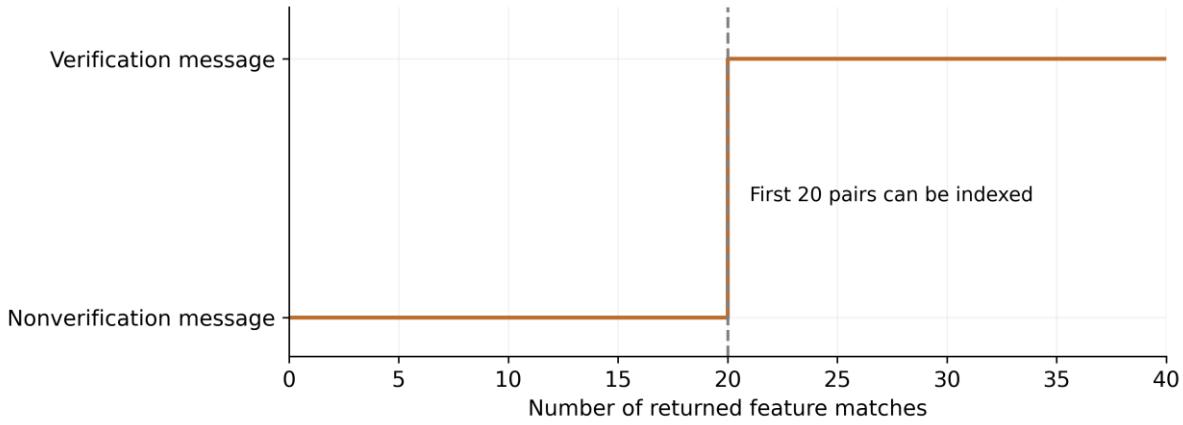


Figure 5. Analytical match-count boundary in the visible local-feature callback, assuming no other runtime error. This is code-path analysis, not a measured biometric response curve.

Table 3 Boundary cases for the original array-indexing condition

Returned pairs M	**First 20 pairs indexable**	**Consequence under the stated assumption**
0	No	Exception path
19	No	Exception path
20	Yes	Success message can be reached
21	Yes	Success message can be reached

Note. Successful indexing does not establish correct identity, score calibration or liveness.

Recognition evidence and service implications

The development test matrix records 65 Class-1-to-Class-1 and 37 Class-2-to-Class-2 assignments, with three off-diagonal errors. This gives 102/105 = 97.14% accuracy for the two-class component experiment. The complete development record contains 699 observations and 677 correct classifications. The separate application table reports 18 True-coded entries from 20 cases. These results support the existence of a recognition demonstration but do not measure the completeness of record linkage, transaction atomicity or payment authorization.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue X October 2026

A pension service should evaluate a complete episode rather than only a successful frame. The episode begins with selection of the claimed record, may include capture retries and ends with confirmation or referral. Measuring these states would reveal whether difficulties arise from acquisition, matching, enrollment linkage or personnel-record inconsistencies. The architecture makes those outcomes observable through status and audit fields.

The model-version field is also important for maintenance. A change to the feature extractor or neural network can alter the representation stored in a template. Compatibility should therefore be checked before a new model is activated, and comparison events should retain the version used. This requirement follows from the proposed data relationships; it is not evidence that the existing database already enforces version compatibility.

Pilot deployment and limitations

A controlled parallel pilot provides a way to evaluate the architecture while maintaining the institution's existing review procedure. The pilot should record service time, recapture count, unresolved claims, record-linkage discrepancies and audit completeness. It should preserve a manual route for claimants who cannot produce a usable image. No reduction in waiting time, staffing cost or improper payment is inferred until those outcomes are measured.

Security evaluation should include printed-image and replay attacks and should assess the handling of templates and administrative accounts. Anti-spoofing capabilities must be evaluated separately from identity matching (Yu et al., 2023). The current study also lacks concurrent-user tests, backup-restoration trials and a complete executable database export. Consequently, the architectural contribution is a formal route to accountable integration, with the visible prototype providing its implementation starting point.

The principal design finding is that explicit states and stable record links resolve an ambiguity that a recognition percentage cannot address: whether the system has observed a usable face, retrieved a candidate, supported a claim or authorized an administrative action. Keeping these events separate allows technical improvements to be evaluated without losing the history of the service decision.

CONCLUSION

The study developed a formal integration architecture for a facial pensioner-verification prototype. Its state model, matching interface and person–template–episode–audit schema distinguish technical processing from identity review and administrative authorization. Code analysis identified a 20-pair indexing boundary and ranked-image retrieval behavior that should not be treated as calibrated verification. The original recognition results provide component evidence, while transactional operation, security and service outcomes require a monitored pilot. The architecture offers a technically explicit basis for that next stage.

Declarations

Data and software availability. The analysis uses aggregate counts and software excerpts from the underlying study. Identifiable facial images and pension records remain controlled by the original data custodian. Reconstructed count tables and analytical expressions are reported in the article.

Ethics and consent. Institutional ethics approval details, permission to use the pensioner records, and participant-consent documentation require confirmation by the original investigators before submission. No identifiable participant photographs are reproduced here.

Funding and competing interests. The authors must confirm the funding statement and any competing interests.

Author contributions. The author list, affiliations, corresponding author and individual research roles require confirmation by the research team.

Use of generative AI. Generative AI assisted language editing, mathematical exposition, analytical scripting and figure preparation. The authors are responsible for verifying the reported evidence, calculations and final manuscript.

REFERENCES

1. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Vol. 1, pp. 886–893). IEEE. https://doi.org/10.1109/CVPR.2005.177
2. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4690–4699). https://openaccess.thecvf.com/content_CVPR 2019/html/Deng_ArcFace Additive Angular_Margin_Loss_for_Deep_Face_Recognition_CVPR_2019_paper.html
3. Getreuer, P. (2013). A survey of Gaussian convolution algorithms. Image Processing On Line, 3, 286–310. https://doi.org/10.5201/ipol.2013.87
4. Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. IEEE Transactions on Circuits and Systems for Video Technology, 14(1), 4–20. https://doi.org/10.1109/TCSVT.2003.818349
5. Kim, M., Jain, A. K., & Liu, X. (2022). AdaFace: Quality adaptive margin for face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18750–18759). https://openaccess.thecvf.com/content/CVPR2022/html/Kim AdaFace Quality Adaptive Margin_for_Face_Recognition_CVPR_2022_paper.html
6. MathWorks. (n.d.-a). matchFeatures: Find matching features. Retrieved October 17, 2026, from https://www.mathworks.com/help/vision/ref/matchfeatures.html
7. MathWorks. (n.d.-b). retrieveImages: Search image set for similar image. Retrieved October 17, 2026, from https://www.mathworks.com/help/vision/ref/retrieveimages.html
8. Meng, Q., Zhao, S., Huang, Z., & Zhou, F. (2021). MagFace: A universal representation for face recognition and quality assessment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14225–14234). https://openaccess.thecvf.com/content/ CVPR2021/ html/Meng_MagFace_A_Universal_Representation_for_Face_Recognition_and_Quality_Assessment_ CVPR_2021_paper.html
9. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 815–823). https://openaccess.thecvf.com/content_cvpr 2015/html/Schroff FaceNet A_Unified_2015_CVPR_paper.html
10. Sharma, D., & Selwal, A. (2023). A survey on face presentation attack detection mechanisms: Hitherto and future perspectives. Multimedia Systems, 29, 1527–1577. https://doi.org/10.1007/s00530-023-01070-5
11. Sun, Z., & Liu, Z. (2025). Ensuring privacy in face recognition: A survey on data generation, inference and storage. Discover Applied Sciences, 7, Article 441. https://doi.org/10.1007/s42452-025-06987-2
12. Wang, M., & Deng, W. (2021). Deep face recognition: A survey. Neurocomputing, 429, 215–244. https://doi.org/10.1016/j.neucom.2020.10.081
13. Yu, Z., Liu, A., Zhao, C., Cheng, K. H. M., Cheng, X., & Zhao, G. (2023). Flexible-modal face anti-spoofing: A benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 6346–6351). https://openaccess.thecvf.com/content/CVPR2023W/ FAS/html/Yu_Flexible-Modal_Face_Anti-Spoofing_A_Benchmark_CVPRW_2023_paper.html