iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84380

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

AI-Based Missing Person Identification and Tracking System

Dr. Chandrakanth G Pujari[1], Kannika D, H Kavana3, Kavya R[4]
[1]Professor, Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore-560056, Karnataka, India
[2]Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore-560056, Karnataka, India
[3]Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore-560056, Karnataka, India
[4]Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore-560056, Karnataka, India

Abstract: Finding a missing person is a race against the clock, yet in most jurisdictions the process still depends heavily on manual effort: officers dig through paper case files, pass around printed photographs, and lean on personal memory to connect a fresh sighting with an older report. As the number of open cases piles up, this kind of manual cross-checking only gets slower and less dependable, and a promising lead can slip through simply because nobody happened to place two particular photographs side by side. This report looks at a software system built to automate that comparison step, using computer-vision-based face geometry rather than raw image matching. Every photograph that enters the system, whether it comes attached to a missing-person case or a public sighting, is run through a 468-point three-dimensional facial landmark detector, which produces a fixed-length numeric signature for the face. A distance-weighted K-Nearest Neighbour classifier then checks new signatures against the pool of open cases and flags any pair that falls within a configurable similarity threshold, automatically closing the case and emailing the reporting family. The system keeps its two user groups on separate interfaces: a role-gated portal for police officers and administrators, and an open public portal that lets any bystander submit a sighting without creating an account. Case and image data live in a local SQLite store, and an interactive map lets administrators see how open cases are spread across cities. Because the core face-landmarker model only needs to be downloaded once and is then cached, the system keeps working without an active internet connection after initial setup, which matters for stations in areas with patchy connectivity.

I. INTRODUCTION

A disproportionate share of missing-person cases involve children, and these cases are reported in large numbers every day; the stretch of time between a disappearance and a confirmed sighting is often where the case is won or lost. An investigating officer is typically juggling dozens of open files at once, and when a new sighting report comes in, checking it by eye against every existing case photograph simply does not scale. That creates a structural bottleneck: the information needed to close a case may already be sitting in the system, filed by a member of the public, but nobody has yet connected the two records. The system examined in this report goes after that bottleneck directly, replacing manual photo comparison with an automated face-geometry matching pipeline built around an off-the-shelf facial landmark model and a lightweight nearest-neighbour classifier, wrapped in a two-sided web interface that keeps official case management separate from public sighting submission.

The problem is made worse by how varied the underlying photographs tend to be. A missing-person photograph submitted at the time of a first information report might be a formal, identity-card style portrait, whereas a sighting photograph submitted later by a member of the public is often a hurried phone snapshot taken at an angle, in bad lighting, or from a distance. Asking an already overloaded officer to mentally adjust for all of these differences every time a new tip comes in is both slow and prone to error. Automating that normalisation step, so that every photograph is reduced to the same kind of representation no matter how or where it was captured, is not just a convenience; it is what makes the underlying comparison task workable at scale.

The rest of this report is organised as follows. Section 2 sets out the objectives behind the system's design. Section 3 surveys related work on automated face matching and places the present system within that landscape. Section 4 walks through the methodology in detail, including the face-landmark extraction step and the matching algorithm. Section 5 lays out the system's architecture, and Section 6 covers implementation-level details such as the technology stack and database schema. Sections 7 through 10 cover the system's benefits, its evaluation, a broader discussion of the design choices behind it, and its known limitations. Section 11 offers a comparative study against the conventional manual process, Section 12 looks at applications beyond the primary use case, and Sections 13 and 14 close out the report with a conclusion and directions for future work.

II. OBJECTIVES

The key objectives of this study are as follows:

1) Explore a parameter-free, geometry-based approach to face matching that removes the need for officers to manually compare photographs one against another.
2) Describe a two-portal architecture that keeps authenticated case management separate from anonymous public sighting submission.
3) Detail the automated matching pipeline that ties new sightings back to open cases and fires off a notification once a match clears a similarity threshold.
4) Assess how the system handles data and offline operation, with an eye to deployment in low-connectivity or resource-constrained police stations.
5) Identify the system's current limitations and the enhancements it would need before it could move toward field deployment.

Taken together, these objectives frame the system less as a finished production tool and more as a proof of concept, meant to show that a lightweight, self-contained matching pipeline can meaningfully cut down the manual workload of connecting a sighting to an open case, without demanding the computational resources or constant internet connectivity that a heavier deep-learning deployment would need.

III. LITERATURE REVIEW

Automated face recognition has a long research history, and the approach taken in the system under review draws on two largely separate threads of that literature: facial landmark detection and classical nearest-neighbour classification.

A. *Facial Landmark Detection*

Lugaresi and colleagues describe MediaPipe as a framework for building cross-platform, real-time perception pipelines, and the Face Landmarker used in this system is one instance of that framework. Rather than producing a single classification label, a landmark detector locates a fixed set of anatomically meaningful points on the face, such as the corners of the eyes, the tip of the nose, and the outline of the jaw. That geometric output suits a matching application well, since it is far lower-dimensional than the original image and is tied explicitly to facial structure rather than to incidental factors like background clutter or clothing.

B. *Classical Face-Recognition Approaches*

Kortli and colleagues survey the broader field of face-recognition systems, drawing a line between holistic approaches that treat the whole face as a single pattern and feature-based approaches that first localise individual facial components before comparing them. The system reviewed here sits in the feature-based camp: rather than learning a representation end-to-end from labelled face pairs, the way a deep metric-learning model would, it relies on a pre-trained, general-purpose landmark detector and treats the resulting geometry as the unit of comparison. That choice gives up some of the accuracy ceiling a purpose-trained face-recognition network could reach, in exchange for a system that needs no dataset of labelled face pairs and no GPU-backed training infrastructure.

C. *Nearest-Neighbour Classification*

Altman's work on nearest-neighbour methods, together with Pedregosa and colleagues' description of the scikit-learn library, forms the classical machine-learning foundation behind the matching step. A K-Nearest Neighbour classifier makes no assumption about how the data is distributed and needs no explicit training phase beyond indexing the examples already on hand, which suits a case pool that keeps changing as cases open and close. The distance-weighted variant used here gives closer neighbours more say in the decision than farther ones do, which fits a matching task where what really matters is the single nearest case, not a majority vote across several neighbours.

D. *Positioning of the Present System*

Compared with the wider survey and prior open-source efforts such as Manku's missing-person matching repository, the system examined here offers a complete, deployable pipeline rather than an isolated matching algorithm. It brings together the landmark-plus-KNN matching core with a role-gated officer portal, a login-free public sighting portal, local data storage, and automated family notification, all built around the operational realities of a resource-limited police station rather than around chasing benchmark accuracy.

IV. METHODOLOGY

At its core, the system reduces every submitted face to a fixed-size numeric representation and treats matching as a nearest-neighbour search problem within that representation space, rather than comparing images pixel by pixel. That keeps the matching step fast and consistent no matter how the two photographs being compared differ in lighting, camera angle, or resolution.

A. Face Landmark Extraction

Every photograph that enters the system, whether attached to a case an officer has registered or a sighting a member of the public has submitted, first passes through Google's MediaPipe Face Landmarker. The detector locates each face in the frame and returns 468 three-dimensional landmark points describing that face's geometry. These points are flattened into a single 1,404-dimensional numeric vector, 468 landmarks each carrying an x, y, and z coordinate, which becomes the stored signature for that face. Because the vector comes purely from facial geometry rather than pixel colour values, it holds up reasonably well against differences in image quality, background, and lighting between the original case photograph and a later sighting photograph.

Before landmark extraction happens, every uploaded image goes through a lightweight pre-processing step that fixes orientation metadata, resizes the image to something the detector can work with, and rejects any frame where no face can be found, returning a clear error to whoever submitted it rather than quietly storing a record that is not usable. When a photograph contains more than one face, the detector's largest bounding box is treated as the primary subject, on the assumption that the person being registered or reported is the most prominent face in the frame.

B. Case Registration and Sighting Submission

An officer registers a missing-person case by uploading a photograph through the officer portal; the system detects the face, extracts its landmark vector, and stores it alongside the case's descriptive details, such as the person's name, last-seen location, and the complainant's contact information. Someone from the public who thinks they have spotted the same person submits a photograph, or a short video, through a separate, login-free public portal. When a video comes in, individual frames are sampled and each detected face is treated as its own candidate sighting, so a single video can surface several distinct sightings worth matching.

Frames from video submissions are sampled at a fixed interval rather than every single frame, which keeps the number of candidate sightings from one video manageable while still giving the matching engine several independent looks at the subject's face from slightly different angles and expressions. Sightings that resolve to near-identical landmark vectors from the same video are collapsed into one before joining the pending-sightings pool, so a single video does not end up dominating a later matching run.

C. Matching Algorithm

Matching runs on a K-Nearest Neighbour classifier, specifically scikit-learn's KNeighborsClassifier, using a ball-tree index with distance-based weighting. For every new sighting vector, the classifier searches the pool of open case vectors for the nearest neighbour and returns both the distance to that neighbour and the matching case's identifier. If the distance comes in at or below a configured threshold, the sighting counts as a positive match: the corresponding case is automatically marked Found, and the pairing, sighting, matched case, and distance score, is logged for an administrator to review.

A ball-tree index was chosen over a flat brute-force search because it scales more gracefully as the number of open cases grows, while still returning exact nearest-neighbour results rather than an approximation. Since the landmark vectors run to 1,404 dimensions, the ball-tree's edge over brute-force search shrinks somewhat compared with lower-dimensional data, but it still holds up as the better default for a system whose case pool is expected to keep growing over the life of a deployment.

D. Notification and Case Closure

Once a match is confirmed, the system emails the complainant at the contact address captured during case registration, so the reporting family hears the news without an officer having to track them down and inform them manually. Administrators keep full visibility over every match, edit, or deletion, while officers below admin level can register new cases but only see the cases they personally filed.

The notification email is deliberately careful about what it reveals: it tells the complainant that a possible match has turned up and asks them to contact the station to verify it, rather than attaching the sighting photograph directly. An automated match, after all, is a lead to be confirmed by an officer, not a certainty to act on right away.

E. Access Control and Privacy

The officer and administrator portal sits behind a login screen, with credentials stored as bcrypt password hashes and role assignments, Admin or Officer, set per user. The public sighting portal deliberately skips any login requirement, since the whole point is to make it as easy as possible for a bystander to report what they saw. All case data, images, and derived face vectors stay local, in a SQLite database and an on-disk image folder; nothing goes out to an external service, which keeps the deployment self-contained and shrinks the data-protection surface a police department would otherwise have to manage.

V. SYSTEM ARCHITECTURE

The system is organised as a set of loosely coupled layers connected through a shared local database, which lets the officer-facing and public-facing applications run and scale on their own while still working off the same underlying case data.

1) Dual Portal Layer: Two separate Streamlit applications serve as the entry points into the system. The officer and admin portal handles case registration, case browsing, triggering matching runs, and the map-based reporting view. The public portal is a lighter, mobile-oriented application whose only job is to let a bystander submit a sighting photograph or video.
2) Face Detection and Encoding Layer: Both portals send uploaded media through the same face-detection and landmark-extraction routine, so a case photograph and a sighting photograph get encoded the same way and stay directly comparable no matter which portal they came through.
3) Data Storage Layer: Registered cases and public sighting submissions are persisted through SQLModel into a local SQLite database, with each record holding the face landmark vector as structured JSON alongside descriptive case metadata and a status flag, Found or Not Found.
4) Matching Engine: An administrator triggers a matching run from the dashboard whenever needed. The engine loads every open case and every pending sighting, builds a nearest-neighbour index over the open cases, and scores each pending sighting against it, updating case status and logging any matches whose distance clears the configured threshold.
5) Reporting and Notification Layer: A Folium-based interactive map groups cases by city so administrators can see at a glance where open cases are concentrated, alongside summary counts of found and not-found cases. When the matching engine confirms a match, this same layer fires off the outbound email notification to the case's complainant.

VI. IMPLEMENTATION DETAILS

This section walks through the concrete technology choices behind each architectural layer and how they come together in a working deployment.

A. Technology Stack

Both the officer and public portals are built with Streamlit, chosen for how quickly a data-oriented Python application can be turned into an interactive web interface without writing separate front-end and back-end codebases. Face detection and landmark extraction rely on Google's MediaPipe Face Landmarker task, run locally through its Python bindings. Scikit-learn's KNeighborsClassifier handles the matching engine. Data persistence goes through SQLModel as an object-relational layer over a SQLite database file, and Folium renders the map view. Officer and administrator passwords are hashed with bcrypt, and outbound notifications go out through a standard SMTP email client.

B. Database Schema

The database revolves around two main tables. The Cases table stores one row per missing-person case: the case identifier, the registering officer's identifier, the missing person's name and last-seen location, complainant contact details, the stored face-landmark vector as JSON, a status flag, and a timestamp. The Sightings table stores one row per submitted sighting or extracted video frame: a sighting identifier, the submitted image path, the extracted landmark vector, the submission timestamp, and, once a matching run has processed it, the matched case identifier and the recorded distance score. A separate Users table holds officer and administrator accounts along with their hashed credentials and role assignment.

C. User Interface Design

The officer portal shows a case list with search and filter controls, a case-registration form, a dashboard summarising open, found, and pending-review counts, and the Folium map view. The public portal, by contrast, is kept deliberately minimal: a single upload control for a photograph or video, an optional free-text description field, and a confirmation screen once the submission goes through, reflecting the goal of keeping the barrier to reporting a sighting as low as possible for a bystander who may be in a hurry or unfamiliar with the system.

VII. KEY BENEFITS

As designed, the system offers several practical advantages over a manual case-comparison workflow:

1) Speed: a sighting can be checked against every open case in seconds, instead of requiring an officer to review files one by one.
2) Consistency: every photograph is reduced to the same 1,404-dimensional representation, so matching does not hinge on any one officer's memory or attentiveness.
3) Lower deployment cost: the matching model is a lightweight KNN classifier rather than a large deep-learning system, so it runs fine on modest hardware.
4) Offline operation: once the landmark-detection model is cached locally, the system needs no live internet connection to keep matching cases.
5) Separation of concerns: a no-login public portal lowers the barrier for bystanders reporting a sighting, while the officer portal keeps case management behind role-based access control.

These benefits build on each other rather than standing alone: it is the combination of low computational cost and offline operation that makes the system realistic for a rural or resource-constrained station, and it is the combination of speed and consistency that gives officers a real reason to trust the automated matching step rather than treating it as one more task piled on top of the existing manual process.

VIII. TESTING AND EVALUATION

As an applied engineering project, the system's evaluation so far has focused on functional verification rather than an accuracy benchmark measured against a large labelled dataset of missing-person cases, since no such dataset exists in a form that could ethically be assembled for a student project.

A. Functional Testing

Each portal was tested manually against representative scenarios: registering a case with a clear portrait photograph, submitting a sighting photograph of the same person taken under different lighting, submitting an unrelated person's photograph to confirm no false match gets raised, and submitting a short video to confirm that multiple frames are sampled correctly and treated as independent candidate sightings. Role-based access was checked by confirming that an officer account cannot view or edit cases registered by a different officer, while an administrator account can see and act on all of them.

B. Matching Behaviour Under Varying Conditions

Informal testing against a small set of volunteer photographs suggests the landmark-based signature holds up reasonably well across changes in lighting and camera angle, which fits the geometry-based design rationale, but it is more sensitive to partial occlusion, a face partly covered by a mask, sunglasses, or a hand, and to long gaps between the case photograph and the sighting photograph, since facial geometry can shift with ageing, weight change, or injury. These observations line up with the trade-off already noted in the discussion of algorithm choice and motivate several of the future-work items described later in this report.

C. Performance Characteristics

Since the matching engine only runs on demand rather than continuously, its running time scales with the size of the open-case pool and the number of pending sightings at the time it is triggered, and it stayed well within seconds for the small case pools used during testing.

How the system actually performs at the scale of a real station's caseload, hundreds or thousands of open cases, has not yet been measured, and remains an open question for future deployment testing.

IX. DISCUSSION

Because this is an applied engineering project rather than a benchmarked research model, its evaluation is better framed around the design choices behind it than around a formal accuracy figure measured against a labelled test set.

A. Choice of Matching Algorithm

A distance-weighted K-Nearest Neighbour classifier was chosen over heavier deep-learning face-recognition models because it needs no separate training phase beyond indexing the current set of open cases, retrains almost instantly as cases are added or closed, and is simple enough to audit and reason about when a match gets contested. The trade-off is that KNN-based matching leans heavily on the quality and consistency of the underlying landmark vectors, and can be more sensitive to poor-quality or heavily obscured photographs than a model trained end-to-end for face recognition.

B. Threshold Sensitivity

The distance threshold that decides whether two faces match is a single configurable number, which means the system's precision and recall can be tuned after the fact without retraining anything. Set it low and false positives drop, but genuine matches can slip through when the sighting photograph is poorly lit or taken from an awkward angle; set it higher and more genuine matches get caught, at the cost of more false alerts that an administrator has to manually review and dismiss.

C. Video-Based Sightings

Supporting video uploads, not just still photographs, widens the range of usable evidence a bystander can submit, since a short phone video is often easier to grab in the moment than a clean photograph. Extracting and matching multiple candidate faces per video, however, does add to the volume of candidate sightings the matching engine has to work through on each run.

X. LIMITATIONS

It is worth stating plainly a few limitations of the current system, both to set expectations for anyone considering deployment and to motivate the future-work items described later in this report.

1) No formal accuracy benchmark: without a large, ethically sourced dataset of paired missing-person and sighting photographs, the system's true false-positive and false-negative rates at production scale remain unmeasured.
2) Single-face assumption: the current pipeline treats the most prominent detected face in an image as the subject, which can misfire on group photographs where the missing individual is not the largest face present.
3) Sensitivity to occlusion and ageing: landmark geometry can shift enough with masks, sunglasses, injury, or the passage of time that a genuine match ends up falling outside the configured distance threshold.
4) Single-node deployment: the local-SQLite design that makes offline operation possible also means the system does not, out of the box, support multiple stations querying a shared case pool without extra synchronisation work.
5) Manual threshold configuration: the distance threshold is currently set by an administrator rather than learned from labelled examples, which leaves the burden of balancing precision against recall on a human operator.

None of these limitations take away from the system's core value, cutting down manual comparison workload, but they do mean any automated match should still be treated as a lead requiring officer verification rather than a final determination, and that a production deployment would need the extra safeguards discussed under future work.

XI. COMPARATIVE STUDY

The comparison below sets the proposed system against the conventional, manual case-handling workflow that is still common in many missing-person reporting processes.

Table 1: Comparison with Conventional Manual Case Handling

Parameter	Manual Process	Proposed System
Photo Comparison	Officer reviews files and photographs by eye	Automated face-landmark distance matching
Public Reporting	In person or by phone to a station	Login-free web portal, photo or video upload
Case-to-Sighting Linking	Depends on officer memory and manual cross-referencing	Automated nearest-neighbour matching on demand
Family Notification	Manual phone calls once a match is found	Automatic email alert on confirmed match
Case Visibility	Paper files, station-specific	Live dashboard and city-level map for all open cases
Scalability	Degrades as open-case count grows	Matching time grows gracefully with ball-tree indexing
Connectivity Requirement	None, but limited by phone or in-person availability	None after initial model download and caching

XII. APPLICATIONS

The underlying pipeline is not tied to a single police station; it could be adapted to a range of settings where matching a new photograph against a known set of faces would be useful.

1) Police Stations and Missing-Persons Units: As the primary intended use case, the system gives officers a way to register cases quickly and have new sightings checked automatically, instead of manually revisiting every open file each time a new report comes in.
2) NGOs and Child-Welfare Organisations: Organisations running their own missing-child helplines or shelters could run a similar pipeline to cross-check intake photographs against their own records, or against cases shared with them by law enforcement.
3) Disaster Response and Family Reunification: During large-scale evacuations or disasters, a similar photo-matching flow could help reunite separated family members by matching photographs submitted at shelters or relief camps against a shared registry.
4) Border and Transit Checkpoints: With the right legal safeguards in place, a comparable matching pipeline could flag when a photograph captured at a checkpoint corresponds to an open missing-person case.

XIII. CONCLUSION

This report has looked at an AI-assisted missing-person identification system that replaces manual photo comparison with an automated pipeline built on 468-point facial landmark extraction and distance-weighted K-Nearest Neighbour matching. By splitting case management into a role-gated officer and administrator portal and pairing it with a login-free public sighting portal, the system lowers the barrier for both official case handling and public participation in reporting sightings. Relying on a lightweight classifier rather than a large deep-learning model keeps it inexpensive to run and, once the landmark-detection model is cached, able to keep operating without an active internet connection. Overall, the system shows how a fairly simple, well-chosen machine learning pipeline, combined with careful attention to the two very different groups of people it serves, can meaningfully cut the manual workload behind connecting a new sighting to an existing missing-person case. That said, the limitations discussed in this report make clear that, in its present form, the system is best understood as a proof of concept rather than a finished production tool. What it demonstrates is that a lightweight, self-contained, offline-capable matching pipeline is a workable design point for resource-constrained deployments, even where a larger, purpose-trained deep-learning system might eventually reach higher raw accuracy.

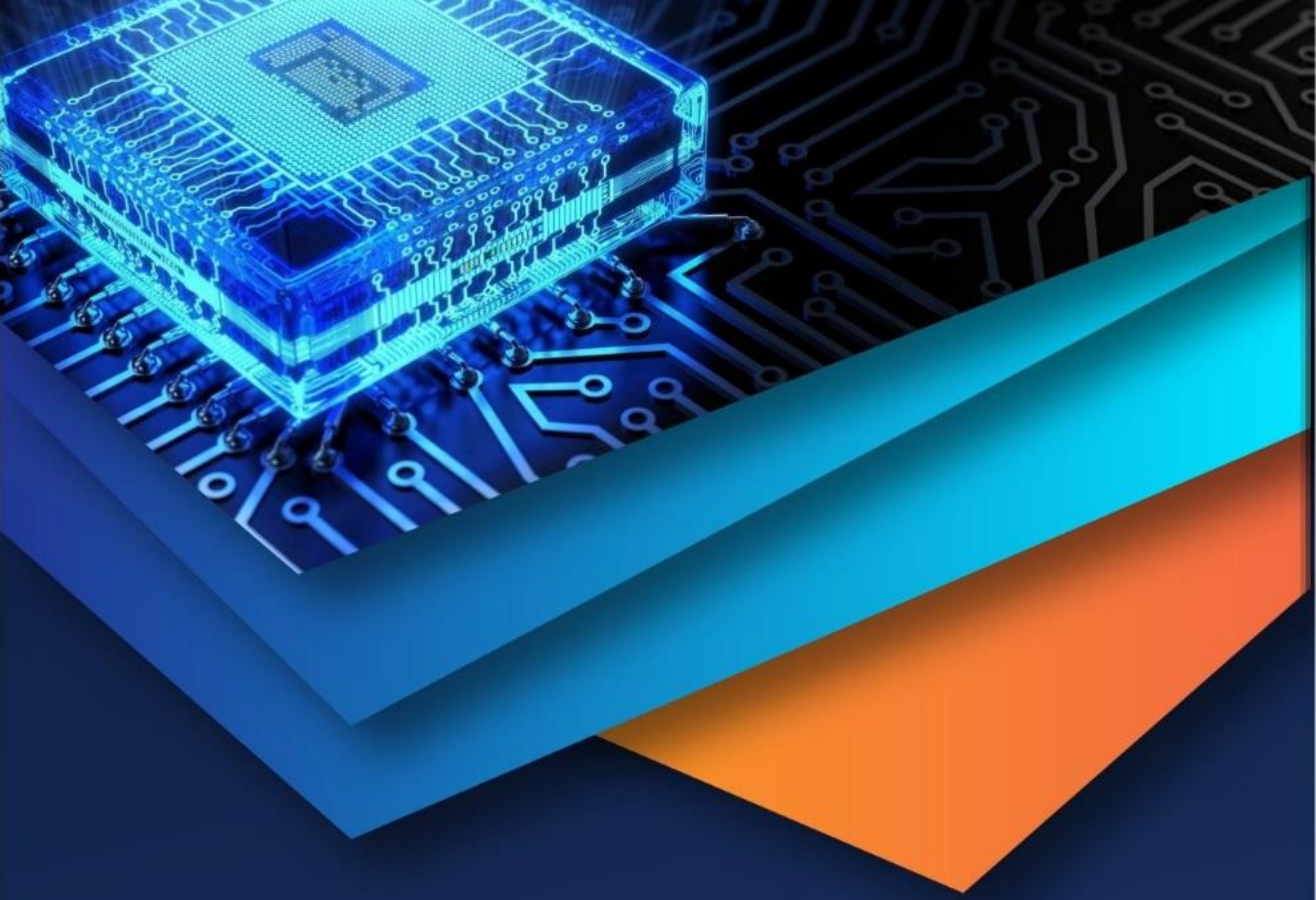
XIV. FUTURE WORK

A few directions could take the system beyond its current form:

1) Deep-Learning Face Embeddings: replacing or supplementing the landmark-geometry vector with embeddings from a trained face-recognition network, which could improve robustness to occlusion, ageing, and extreme pose variation.
2) Real-Time Surveillance Integration: extending the matching pipeline to run against live CCTV or checkpoint camera feeds, rather than only user-submitted photographs and videos.
3) Cross-Jurisdiction Data Sharing: letting multiple stations or agencies query a shared case pool while keeping local data ownership and access control intact.
4) SMS and Mobile Push Alerts: adding SMS or push notification channels alongside email, since not every complainant checks their inbox regularly.
5) Automated Threshold Tuning: learning a distance threshold from labelled match and non-match examples over time, instead of relying on a single fixed, manually configured value.
6) Multi-Lingual Public Portal: localising the public sighting portal so language is not a barrier to reporting a sighting.
7) Formal Accuracy Benchmarking: putting together an ethically sourced evaluation dataset to measure false-positive and false-negative rates before any production deployment.
8) Multi-Face Handling: extending the pipeline to consider every detected face in a submitted image rather than just the most prominent one, to better handle group photographs.

REFERENCES

[1] C. Lugaresi, J. Tang, H. Nash, et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv:1906.08172, 2019.
[2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
[3] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," The American Statistician, vol. 46, no. 3, pp. 175-185, 1992.
[4] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face Recognition Systems: A Survey," Sensors, vol. 20, no. 2, Art. no. 342, 2020.
[5] G. Manku, "Finding Missing Person Using AI" (open-source project repository), GitHub, 2024.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY