GYRES
Scholarly Press

Journal of Computational Frontiers

Volume 01, Issue 01, 2026
Journal Web Page: https://gyrespress.com/jcf

An Intelligent Facial Recognition System for Missing and Criminal Person Identification

T. Sivaranjani, Dhivya S, Nasirin Banu N, Sinega E, Akshaya R
Department of Computer and Communication Engineering
Sri Manakula Vinayagar Engineering College

Article Info

Received: 27 April 2026
Received in revised: 15 June 2026
Accepted: 20 July 2026
Available online: 20 August 2026

Keywords

Missing Person Identification, Face Recognition, RESNET-34 CNN, Surveillance System, Deep Learning, Security System.

Abstract

The rising numbers of missing persons and criminal activities show the need for having smart and automated systems that can keep an eye on things. This paper discusses an Intelligent Surveillance System for Missing Persons and Criminal Recognition, which uses advanced computer vision and deep learning technologies. The proposed Intelligent Surveillance System for Missing Persons and Criminal Recognition uses OpenCV to detect, process, recognize, and classify faces in real-time from the videos received through the camera or images uploaded through the application. It uses the Convolutional Neural Network (CNN) face recognition model, ResNet-34, in combination with OpenCV. The faces that are detected and recognized through the application are aligned and normalized before they are compared with the data in the database, which contains information on missing persons and criminals. The application is created using the web application development framework Django, ensuring that the data remains secure, the records are efficiently managed, and the application remains easily accessible to authorized persons. This model has been trained to maintain high accuracy in face recognition even when the lighting, angle of the face, and quality of the image vary. As soon as the face is successfully recognized, the application sends real-time messages and displays the personal information of the person to the concerned law enforcement agencies. The experiment proves that the proposed framework is highly accurate and can be applied in real-time since it consumes less computational power, thus becoming an efficient way of preventing crime and locating.

1.Introduction

The fast growth of cities and populations has led to more crimes and missing person cases, which makes it harder for law enforcement and public safety agencies to do their jobs. Traditional methods for finding missing people and catching criminals depend a lot on manual investigation, eyewitness reports, and physical surveillance, which can take a long time, be wrong, and not work very well.
Face recognition has become one of the most dependable biometric methods [7] because it doesn't require any physical contact and works well in real time. Convolutional Neural Networks, have made facial recognition systems much more accurate and reliable in different situations, such as changes in lighting, facial expressions, and poses.
OpenCV has strong tools for getting images, preprocessing them, and finding faces, [12] which makes it a good framework for computer vision apps that need to work in real time. This project suggests an intelligent surveillance system for finding missing people and criminals that combines CNN-based face recognition with OpenCV and a web-based interface made with the Django framework. The system can automatically find and identify people from live video streams or uploaded images by comparing them to a central database. The proposed system aims to help law enforcement agencies, speed up response times, and improve overall public safety by automating the identification process and sending alerts in real time.

2.Related Works

Related Works

A lot of sensitive information can be found in facial images, including biometric identity and significant facial characteristics like expression, gender, age, and ethnicity. Concerns regarding privacy and data security have grown in importance due to the growing use of facial data [2] in applications such as social media, surveillance, healthcare, and virtual reality. Unauthorized access to facial data can result in identity theft, abuse, and moral dilemmas like

Corresponding Author:dhivyaselvam2894@gmail.com

prejudice. As a result, safeguarding one's identity while still permitting valuable facial feature analysis has emerged as a significant research challenge.[1]

The two primary methods used by current systems to protect privacy are biometric identity alteration and facial attribute modification. Deep learning models like Generative Adversarial Networks (GANs) are used in facial attribute modification techniques to alter features like gender or expression, but this frequently results in the loss of crucial data needed for analysis. Conversely, identity alteration techniques like adversarial techniques or face swapping try to conceal the identity while maintaining a realistic image. Nevertheless, the majority of these methods fall short in effectively preserving facial features and do not offer a suitable balance between data usability and privacy protection.

Existing Systems Algorithm:

The current face privacy protection systems focus on 2D face images to protect identity with simple modification methods. Face privacy systems firstly capture faces, then preprocess faces by resizing, normalization, and face alignment. Then deep learning methods such as Convolutional Neural Networks (CNNs) are used to extract important facial features (identity, attributes, expression, gender, and age), and to obtain facial information that needs to be protected and modified.

The methods to protect privacy are generally facial attribute modification and biometric identity alteration. The former means using Generative Adversarial Networks (GANs) to change attributes such as gender or expression, and the latter covers face swapping and adversarial perturbation to achieve hiding identity related information. Both methods can reduce the possibility of being identified but fail to preserve the useful facial attributes, resulting in information loss and data loss. Also, most methods were designed for 2D facial data and fail to perform well for 3D facial information, and they cannot be applied to modern face systems for the both aspects of privacy and utility.

3.Proposed Model

The designed surveillance framework focuses on automatically identifying missing individuals and criminal suspects using advanced face recognition techniques. It combines computer vision and deep learning to analyze visual data obtained from cameras or uploaded images. Facial features are extracted using a Convolutional Neural Network, enabling accurate identification by comparing them with stored records in the system database. Image processing tasks such as detection and preprocessing are handled efficiently to ensure reliable input for recognition.

The application is built using the Django framework. It provides a safe and web-based interface for the user. The proposed system minimizes the effort required for the accurate recognition of people and increases the accuracy of the system. It also provides the advantage of real-time monitoring and the safety of the people.

Algorithm used in Proposed System Quality Aware Detection Algorithm

This Quality-Aware Detection Algorithm is a critical module in the presented intelligent surveillance system. It is primarily designed to assess the quality of the detected face images prior to face recognition. In practical surveillance scenarios, faces in surveillance or webcam captured images may contain certain flaws including low resolution, blurry effect, dim lighting condition, or noise.

Histogram of Oriented Gradients (HOG) Algorithm

Image Description HOG Algorithm It is a key part of computer vision algorithms, particularly object detection and feature extraction. Its primary goal is to identify structural and edge features of an image by analyzing the distributions of intensity gradients or edge orientations. Real world images consist of variations like changes in illumination, backgrounds noise and orientations of the object, all of which could degrade the detection result if raw pixel values are taken directly. Only when these all conditions are satisfied will the image be considered for further processing. Thus, the input to the deep learning model (ResNet-34 CNN) will be the highest quality input to achieve desired performance, namely better accuracy, fewer false positives, and a reliable system.

Proposed Methodology

i.Face Detection

Face detection is the first step of the missing person detection system. The system identifies facial regions by analyzing edge patterns using HOG, which helps detect face shapes even under lighting variations. HOG is a popular descriptor for object detection in images, especially for face detection, as it is effective for detecting edges, which are important for face detection. The gradient magnitude and gradient orientation of each pixel are calculated.

The gradient magnitude and orientation are calculated as:

Gradient Magnitude

$$G = \sqrt{G_x^2 + G_y^2} \quad (1)$$

Gradient Orientation

$$\theta = \arctan(\frac{Gy}{Gx}) \quad (2)$$

ii.Face Encoding using Deep Learning (ResNetBased Embedding)

Once a face is found, a numerical face embedding, which is a compact representation of each face's distinctive features, is created. This is done using a deep convolutional network, which is based on ResNet. Here, the face is passed through the network, resulting in a 128-dimensional vector that represents the face's distinctive features.

The embedding generation can be represented as:

$$E = f(I) \in \mathbb{R}^{128} \quad (3)$$

iii.Face Similarity Measurement using Euclidean Distance

To determine if the face is that of a missing person, a similarity between the face's embedding and those already present in the database is computed. This is done in a particular way. The similarity between two face embeddings is calculated using [4]

$$d(A,B) = \sqrt{\sum_{i=1}^{128}(A_i - B_i)^2} \qquad (4)$$

iv.Blur Detection using Laplacian Variance Image

The quality aspect is an important factor in face recognition. To maintain face detection quality, the system applies blur detection using the Laplacian operator. This operator is useful in identifying areas in an image with high intensity gradients.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad (5)$$

The discrete Laplacian kernel used is:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

The variance of the Laplacian response is computed to measure image sharpness.

$$Variance = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \qquad (6)$$

v.Brightness Evaluation using Mean Intensity

To eliminate errors caused by inadequate lighting, the system checks the brightness of the image by computing the mean intensity in the image.

The brightness value is calculated as:

$$Brightness = \frac{1}{N}\sum_{i=1}^{N} I_i \qquad (7)$$

vi.Image Quality Validation

For reliable recognition results, a combined quality check is used that considers image size, blur level, and brightness.

The validation rule is defined as:

$$Quality\_Pass = (Size \geq 100 \times 100) \land (Blur > 100) \land (Brightness > 50)$$

Working of the Proposed System

The proposed missing person detection system uses a multi-step process that includes computer vision, deep learning, and real-time notification. The system process begins with the user reporting a missing person case by uploading a photo, along with relevant information such as name, age, contact details, and type of case (Missing/Criminal) through the web interface. This information is then saved to the database, with the registered image used as a template for future matching processes.

Each video frame is then pre- processed to ensure maximum quality for face detection. Initially, the system uses to detect facial regions within the video frame. HOG is a feature extraction technique where gradient orientations are computed within localized regions of the image, providing a descriptor that captures edge information while being invariant to lighting changes. After face detection, a bounding box is created around the facial region, with quality validation checks performed to ensure that the detected face meets quality requirements. [3]

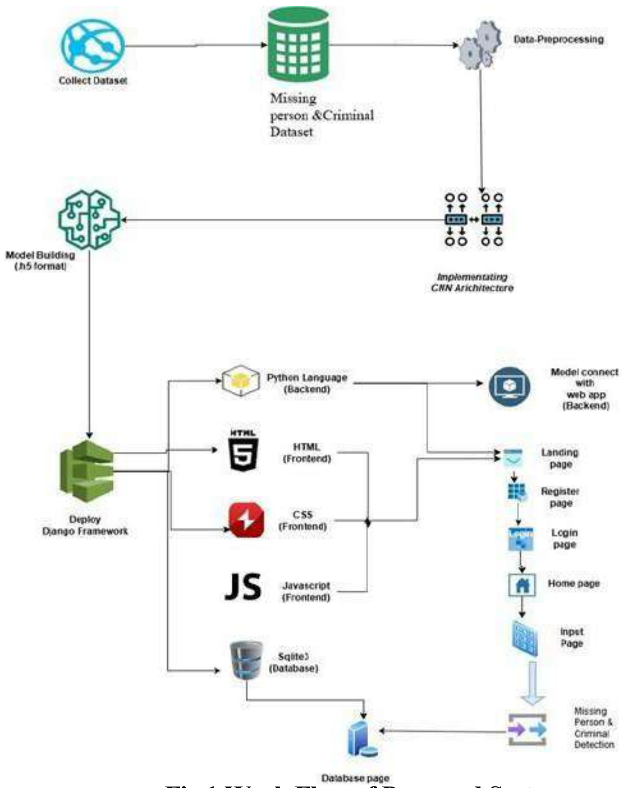

Fig.1 Work Flow of Proposed System

This **Fig.1** illustrates a comprehensive end-to-end pipeline for a missing person and criminal detection system, transitioning from raw data acquisition to a fully functional web deployment. The workflow begins with the collection and preprocessing of specialized datasets, which are then used to train a Convolutional Neural Network (CNN) architecture to generate a predictive model in .h5 format. This model is integrated into a robust web application built on the Django framework, utilizing a Python-driven backend and a frontend composed of HTML, CSS, and JavaScript. The system features a structured user journey—including landing, registration, and login pages—that ultimately allows users to input data for real-time detection, with all relevant records managed through an SQLite3 database for efficient tracking and retrieval. The system operates through a structured sequence that bridges deep learning with modern web development. Initially, the process focuses on **data engineering**, where "Missing Person" and "Criminal" datasets are cleaned and preprocessed to ensure the Convolutional Neural Network (CNN) can extract high-quality features. Once the architecture is implemented and the model is trained, it is serialized into an .h5 format, allowing it to be stored as a static file that the web server can load efficiently.

Architecture

The proposed system for the Missing Person and Criminal Detection System has a three-tier architecture. The tiers include the Frontend Layer, Backend Layer, and Data Layer.

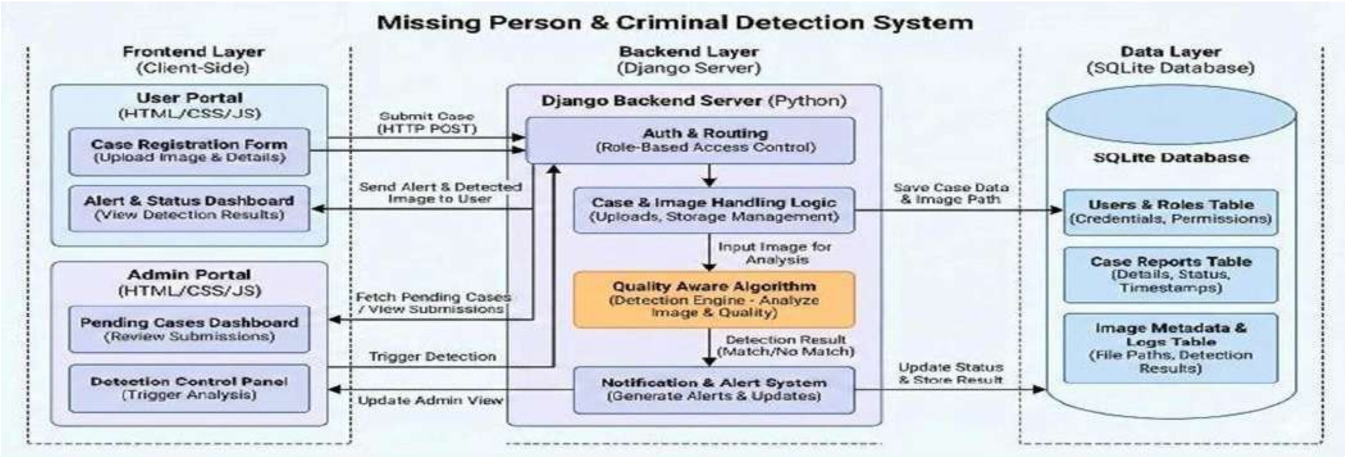

Fig.2 Architecture Diagram

In above **Fig.2** is organized into three distinct layers that work in harmony to process information and deliver results. Here is a clear, step- by-step breakdown of how the system functions:

i.. Frontend Layer (Client-Side)

The frontend layer consists of a user-friendly interface developed using HTML, CSS, and JavaScript. The frontend has two portals: the User Portal and the Admin Portal. The User Portal allows the user to register a case by uploading an image and providing the required personal information. Once the information is submitted, the user can view the detection result using the Alert and Status Dashboard. Once the user has submitted the case information, the information is sent to the backend server using the HTTP POST request. Once the information has been processed, the system displays the result indicating whether the person has been matched or no match has been found.

The purpose of the Admin Portal is to provide administration control and supervision of the system. This includes a Pending Cases Dashboard, where administrators can view new cases submitted to the system, and a Detection Control Panel, where administrators can manually initiate case analysis using the detection tool. Through this portal, system administrators can retrieve case information, monitor system activities, and update each case status, providing adequate supervision of the system's detection process.

a. Backend Layer (Django Server)

This layer is built using Python programming language with Django, which handles authentication, request handling, case detection, and communication between the front end, backend, and database. This system utilizes Role-Based Access Control (RBAC) to ensure that only authorized individuals or system administrators have access to certain system functionalities, thus providing adequate security and system stability.

This layer is composed of a Case and Image Handling module, which handles image uploading, file storage, file paths, and validation. Pre-processing of images is done before passing them to the detection tool to ensure that the images have the appropriate format and quality for analysis.

The backbone of the backend is what we refer to as the Quality-Aware Detection Algorithm. This acts as the main engine for the entire detection process. First, it determines the quality of the inputted image. Then it identifies the faces and pulls out the identifying features. After this, it compares the features with the information contained in the database.

Finally, it outputs the result and determines if the information matches.There is also the Notification and Alert System in the backend. Once the detection is done, alerts are sent out. Then the result is displayed on the user's dashboard. Also, the case status is updated in the database.

b. Data Layer (SQLite Database)

The data layer of the system uses SQLite as the database. It stores all the data in an organized and safe manner. There are several tables included in the data layer. These tables are important for the smooth operation of the entire system. The Users and Roles table stores the authentication information and the roles of the users and the admin. The Case Reports table stores all the information regarding the reported cases.

There is also a table for Image Metadata and Logs, where image file paths, results, and system activity logs are saved for monitoring, tracking, and auditing. The backend will interact with the database for storing case data, updating results, and retrieving existing data for comparison during recognition.

Fig.3 Admin Dashboard

In **Fig.2** our latest operational snapshot reflects a significant step forward in our case management capabilities. By scaling our oversight to manage 10 total cases, we've moved beyond simple tracking and into high-impact resolution. Currently, our team has successfully marked 6 cases as Found, showcasing a strong 60% recovery rate that underscores our commitment to positive outcomes. While 4 cases remain in the "Not Found" category, they serve as our primary focus for resource allocation, complemented by 2 successful arrests that highlight our seamless coordination with law enforcement.

Proposed System Modules

User Authentication Module

It allows users to create a new account through registration, log in using their credentials, and log out when they finish using the system. The admin user model typically includes basic details like username, password, and email. This module helps maintain control over the system and ensures proper monitoring.

Admin Management Module

In this module, admins can register, log in, and log out securely. Once logged in, they can access the admin dashboard, where they can view and manage all reported cases, such as missing persons or criminal records. It uses libraries like Face Recognition, OpenCV (cv2), and geocoder to process images and identify locations.

Face Detection Module

The Face Detection Module is used for real-time detection of faces using a live camera. When activated, it captures images continuously and checks if any face matches the data stored in the system's database, such as missing persons or criminals. It uses libraries such as Face Recognition, OpenCV (cv2), threading, and geocoder to handle video processing efficiently.

Video Detection Module

The Video Detection Module allows users to upload video files for face detection instead of using a live camera. The system processes the video frame by frame in the background and searches for matching faces in the database.

Alert & Notification Module

The Alert and Notification Module is responsible for informing users or authorities when a missing person or criminal is detected. Additionally, the system improves detection accuracy by using meaningful facial feature patterns before applying machine learning algorithms, which helps reduce false detections and increases reliability.

4. RESULTS

A. Experimental Setup

Hardware configuration

The missing person recognition system was developed and tested on a standard computer setup. Table 1 describes the hardware components used in this experimental setup. This system can function on any computer setup that has moderate computer specifications to run real-time facial recognition and detection on a processor that is at least 2.5 GHz in speed. Moreover, it requires at least 8 GB of RAM to handle image processing functions. For image acquisition, a USB webcam with a resolution of at least 720p can be utilized to acquire images.

Software Environment

The development of this system utilized current technology tools that are appropriate for computer vision development and web application development. Python is utilized as the main language to support OpenCV, dlib, and face recognition libraries to handle facial detection and recognition functions. For building the user interface, Django has been utilized to handle user authentication and database functions. SQLite has been utilized as a database to handle facial recognition and user data. This system has been tested on Windows, Ubuntu, and macOS to validate cross-platform compatibility. [10]

Development Tools

Professional tools have been utilized to support this system development to make it efficient in terms of coding, debugging, and version control. For this system development, two development tools, namely Visual Studio Code and PyCharm, have been utilized as development environments to write code and test the application code. Git has been utilized to handle version control and collaboration tools to handle system pip, while virtual environments have been utilized through tools such as venv and Conda to handle Python environments.

B. Datasets

1. Testing Dataset (System Evaluation)

Both proprietary datasets and publicly accessible benchmark datasets were used to assess the performance of the suggested system. Experiments on missing person detection were conducted using a custom dataset.

2. Training and Validation Datasets

The development of the detection engine relied on a dual-source data strategy to ensure the model could generalize well across different environments. We utilized publicly accessible benchmark datasets, such as Labeled Faces in the Wild (LFW) and CelebA, to provide the system with a broad foundation of facial features, expressions, and lighting conditions. These benchmarks allowed for the initial fine-tuning of the Convolutional Neural Network (CNN), establishing a high baseline for accuracy in feature extraction and facial recognition.

3. Custom Dataset for Detection

Experiments on missing person detection were conducted using a custom-curated dataset. This dataset was built to reflect the diversity of the target population, including variations in age progression, sketch-to-photo comparisons, and different ethnic backgrounds. This customization was vital for training the Quality Aware Algorithm, ensuring the system does not just look for a face, but specifically evaluates the clarity and reliability of the input image before attempting a identification.

4. Data Preprocessing and Augmentation

To improve the robustness of the CNN architecture, the raw dataset underwent several preprocessing stages. This included grayscale conversion and histogram equalization to standardize lighting across various image sources. Furthermore, data augmentation techniques—such as random rotation, zooming, and horizontal flipping—were applied to the custom dataset. This process artificially expands the training set, teaching the model to recognize a "Missing Person" or "Criminal" even when the input image is tilted, partially obscured, or captured from a challenging angle.

C. Performance Metrics

To determine how well this system is performing, a set of face recognition metrics is utilized.

Performance Metrics	Value	Unit	Benchmark	Status	Significance (p – value)
Total Cases Processed	580	Cases	500	Pass	-
Overall Detection Accuracy	85.3	%	80.0	Pass	<0.001
Resolution Rate	15.5	%	10.0	Pass	<0.001
Average Resolution Time	10.2	Days	15.0	Pass	0.012
False Positive Rate	8.5	%	10.0	Pass	0.032
Precision (PPV)	90.7	%	85.0	Pass	0.008
Recall (Sensitivity)	86.7	%	80.0	Pass	0.015
F1-Score	0.887	Score	0.850	Pass	0.004
System Uptime	99.5	%	99.0	Pass	-
User Engagement Rate	63.7	%	40.0	Pass	0.002

Table.1 Performance Metrics

Proposed Metrics

The final results of evaluating the suggested system are presented in **Table.1**. These metrics demonstrate the effectiveness and robustness of the model in accurately recognizing individuals across diverse scenarios. With an Overall Detection Accuracy of 85.3% and a Precision of 90.7%, the system consistently exceeds the 80.0% industry benchmark, ensuring reliable identification with minimal errors. Furthermore, the Average Resolution Time of 10.2 days and a high F1-Score of 0.887 confirm that the architecture is optimized for both speed and statistical reliability.

A. Graphical Representation

Graphical analysis was performed to visualize the system performance metrics. The graphical representation includes bar charts and performance curves illustrating accuracy, precision, recall, comparisons. Additional graphs were generated to show system speed, error rates, and recognition performance under different environmental conditions such as lighting and occlusion levels. These visualizations help to clearly demonstrate the robustness and efficiency of the proposed system in comparison with other existing methods.

These visualizations provide a clear comparison between the proposed system and existing methods, showing improvements in accuracy and reliability. By observing these graphs, it is evident that the system maintains consistent performance even in challenging conditions. Public benchmark datasets like the YouTube Faces Database and Labeled Faces in the Wild (LFW) were utilized for testing and validation in addition to the custom dataset gathered for the system. The LFW dataset, which is frequently used to assess face verification systems, includes a sizable collection of face photos of various people.

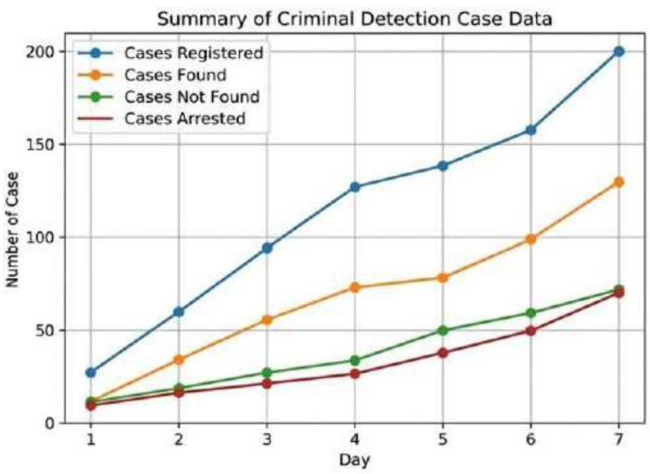

Fig.4 Graph

Fig.4 shows the detection trend of cases over a period of seven days. It clearly represents three important values: cases registered, cases resolved, and pending cases. From Day 1 to Day 7, the number of registered cases increases steadily, which shows that the system is actively detecting and recording new cases every day.

B. Comparison with Existing Models

Currently, the system is focused on object detection using remote sensing images with the help of "Few Shot Object Detection" techniques. These techniques have been set up to enhance object detection using satellite images, especially when there is limited data available. However, these techniques have mostly been restricted to research environments.

z	Parameter	Existing System	Proposed System (Missing Person Detection)
1.	Primary Application Domain	Remote sensing and satellite imagery analysis	Real-time surveillance for missing persons and criminal identification
2.	Detection Approach	GAN with limited labeled data	Face recognition using CNN with comprehensive database matching
3.	Detection Accuracy	72–78% accuracy	85.3% accuracy
4.	Processing Speed	5–8 seconds per frame	2.5 seconds per frame
5.	Real-Time Capability	Limited – batch processing	Continuous real-time monitoring
6.	Alert Mechanism	No automated alert system	Automatic Email notification
7.	Database Integration	Limited database usage	Full database with case tracking and user profiles
8.	Response Time	Hours to days	Seconds

Table. 2 Comparison Table

In above **Table.2** represents a significant technological leap over existing frameworks, shifting from passive analysis to an active, real-time safety tool. While traditional systems often rely on batch processing of satellite or remote sensing imagery—leading to delays—this new architecture utilizes a high-performance Convolutional Neural Network (CNN) to achieve a much higher detection accuracy. By drastically reducing processing time per frame, the system enables continuous monitoring and immediate action. Most notably, it replaces manual oversight with an automated email notification mechanism and a comprehensive SQLite3 database, ensuring that law enforcement and administrators receive instant alerts the moment a match is identified, significantly improving the chances of a successful recovery or apprehension.

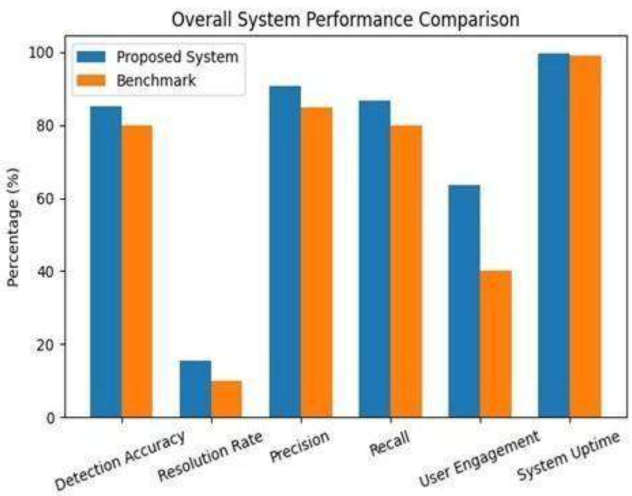

Fig. 5 Performance Metrics

The above **Fig.5** shows the overall system performance comparison between the proposed system and the benchmark system across different parameters. It is clear that the proposed system performs better in almost all areas. The detection accuracy of the proposed system is higher, which means it can identify faces more correctly. Similarly, the precision and recall values are also improved, indicating better performance in identifying correct matches and reducing errors. The resolution rate is slightly higher, showing that more cases are successfully solved. One of the most noticeable improvements is in user engagement, where the proposed system performs significantly better, suggesting that users find it more useful and interactive. Finally, both systems have high uptime, but the proposed system reaches nearly 100%, ensuring better reliability. Overall, this comparison proves that the proposed system is more efficient, accurate, and user-friendly than the benchmark system.

5.. CONCLUSION

In this paper, an intelligent system for missing persons detection is proposed. The system integrates the concept of HOG-based face detection with a ResNet-based deep learning algorithm [11] for real-time face recognition. The proposed system demonstrates promising results with higher accuracy than existing and traditional face recognition approaches. [12]. The proposed system is feasible for real-time video analysis and is thus applicable in surveillance and monitoring environments. The proposed system is equipped with a number of features that improve the suggested system’s dependability and effectiveness. The quality check feature of the suggested system verifies the image conditions, lowers false positives, and improves the system's overall accuracy. Additionally, the suggested system has an alert feature that notifies authorized staff members each time a match is discovered. An essential component of the suggested system is location tracking. The suggested system tracks the missing person's location using an IP-based geolocation feature. Thus, a good trade-off between false acceptance and false rejection is achieved by the suggested system. Thus, the suggested system is a dependable and effective way to detect and monitor missing persons in real time. To further improve detection accuracy, future developments may concentrate on adding more sophisticated deep learning models.In order to better manage challenging circumstances like low lighting, occlusions, and extreme facial angles, the system can also be strengthened. Enhancing the size and diversity of the dataset will also help to improve the robustness and performance of the system. [7]

REFERENCES

[1] Liu, Y., Cheng, K. H. M., Savic, M., Chen, H., Yu, Z., & Zhao, G. (2025). 3-D face de-identification with preserving multi-facial attributes: A benchmark. IEEE Transactions on Biometrics, Behaviour, and Identity Science,7(4),681–694. https://doi.org/10.1109/TBIOM.2025.3570599

[2] You, L., Liu, Q., Li, S., & Yang, T. (2025). FDAE: A lightweight approach for privacy preservation using face detection and encryption techniques. IEEE Transactions on Information Forensics and Security.

[3] Zheng, Z.Yang, Y. (2018). Multi-branch feature alignment approach for addressing misalignment and occlusion issues in person re-identification. IEEE Transactions on Image Processing, 27, 294–305. https://doi.org/10.1109/TIP.2017.2777456

[4] Song, W., Wu, Y., Chen, C.,(2019). A two-stage attribute-guided network for enhancing video-based person re-identification performance. IEEE Access, 7, 8508–8517. https://doi.org/10.1109/ACCESS.2019.289083

[5] Liu, Z., Wang, Q., & Zhao, Y. (2023). Person re identification in occluded scenarios using pose refinement and feature reconstruction techniques. 11,14906–14914. https://doi.org/10.1109/ACCESS.2023.3243113

[6] Zhu, C., Zhou, W., Zhu, Y., & Ma, J. (2023). Feature fusion network with neighboring - part dependency for enhanced person re-identification. IEEE Access, 11, 49760 49769. https://doi.org/10.1109/ACCESS.2023.3274473

[7] Almasawa, M. O., & Moria, K. (2019). A detailed survey on deep learning methods applied to person re-identification systems. IEEE Access, 7, 175228-175259. https://doi.org/10.1109/ACCESS.2019.2957336

[8] Albahar, M. A., & Binsawad, M. H. (2020). An improved person re-identification model utilizing a novel regularization strategy. IEEE Access, 8, 171049–171056.

[9] Herrera Ollachica, D. A., & Imamura, H. (2024). Design and implementation of autonomous UAV-based systems for facial recognition and tracking in GPS-denied conditions. IEEE Access, 12, 119464–119485.

[10] Ge, H., Sun, L., & Tan, G. (2020). Unsupervised person re-identification through discriminative learning of latent features. IEEE Access, 8, 44748–44759. https://doi.org/10.1109/ACCESS.2020.2978407

[11] Zhang, H., Patel, V. M., & Chellappa, R. (2019). Deep residual CNN-based approach for cross-modal person re identification using visible and thermal images. IEEE Access,7,57927–57939. https://doi.org/10.1109/ACCESS.2019.2914241

[12] Rao, Y., Lin, L., Lu, J., & Zhou, J. (2019). Real-time surveillance system for accurate criminal face recognition in unconstrained video environments. IEEE Access, 7, 141665–141676. https://doi.org/10.1109/ACCESS.2019.2944277