iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** IV **Month of publication:** April 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.79479

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

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 IV Apr 2026- Available at www.ijraset.com*

# ANPR-Based Secure Appointment Retrieval and Access Management System

P. Mallikarjuna[1], Asst. Prof. Rajesh Yamparala[2], P. Paavan Siddhartha[3], P. Madhu Babu[4]
*Department of CSE (Data Science), R.V.R & J.C. College of Engineering (Autonomous), Guntur, A P – 522019*

***Abstract:*** ***Managing visitor access in modern buildings has always been a challenge, especially when done manually. This paper describes ANPR-SARAM— an automated visitor access management platform designed for residential complexes, apartment buildings, and corporate offices. Rather than relying on security guards to manually record vehicle numbers, our system does it automatically using cameras and computer vision. We used YOLOv8, a state-of-the-art object detection model, along with CNNs and EasyOCR to detect vehicles, read their license plates in real time, and capture visitor photographs on arrival. All visitor information— including vehicle registration number, photo, entry time, and guest details— is stored on a secure, administrator-controlled server. The result is a smarter, faster, and more reliable way to handle building security that reduces human error and maintains a complete digital record of every visit.***

***Keywords: ANPR; YOLOv8; License Plate Recognition; Smart Building; Visitor Management; CNN; EasyOCR; JWT; QR Authentication; MongoDB; Socket.IO; MERN Stack.***

## I. INTRODUCTION

As cities grow and buildings become smarter, the way we manage who enters a premises has not kept up. Walk into most apartment complexes, gated communities, or even corporate offices today, and you'll likely see the same thing— a security guard with a logbook, manually writing down visitor names and vehicle numbers. This process is slow, messy, and frankly unreliable. Numbers get copied down wrong, records get lost, and there is no easy way to look back at who came and went if something goes wrong [1], [2].

Automatic Number Plate Recognition (ANPR) technology was designed to solve exactly this problem. Also known as License Plate Recognition or Automatic Vehicle Identification, ANPR automatically reads a vehicle's number plate from a camera image, without any human involvement. It processes several linked stages: spotting the vehicle, locating the license plate, separating the characters, and finally reading the text. Each stage must work correctly for the system to succeed, and real-world factors like bad lighting, unusual fonts, or a poorly positioned camera can all cause problems [4], [5].

Deep learning has made a huge difference here. Convolutional Neural Networks (CNNs) are extremely good at learning visual features, which is exactly what you need when identifying license plate characters under different conditions [6], [7]. The YOLO (You Only Look Once) family of models has become the go-to choice for real-time ANPR because it is both fast and accurate. YOLOv8 from Ultralytics, the latest in this family, can detect small objects more reliably and trains faster due to a redesigned anchor- free detection head. For reading plate text, we used EasyOCR, combining CRAFT for character detection and CRNN for recognition, handling plates from many countries with 91–96% character accuracy.

Despite all these advances, one area has been largely overlooked: turning an ANPR system into a complete, production-ready visitor management platform. Most published research focuses only on detecting and reading plates — stopping there. No published system has also addressed pre-registration of visitors, secure QR code check- in, role-based access control, real-time host notifications, and a full security audit trail [13], [14]. That gap motivated this work.

In this paper, we present **ANPR-SARAM** (ANPR-Based Secure Appointment Retrieval and Access Management System), a complete cloud-native platform handling every aspect of visitor management. Key contributions include: (1) a three-tier ANPR pipeline built on YOLOv8n and EasyOCR achieving 89.8% full-plate accuracy on 400 test frames across four lighting conditions; (2) a secure MERN Stack web application with JWT HS256 authentication and Helmet.js hardening; (3) cryptographically secure 256-bit QR tokens; (4) real-time Socket.IO notifications with under 8 ms latency; and (5) automated email alerts sent asynchronously to hosts.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue IV Apr 2026- Available at www.ijraset.com*

## II. LITERATURE REVIEW

Over the past decade, researchers have approached vehicle detection and access control from several angles— starting from simple image processing, moving through handcrafted feature extraction, and finally arriving at the deep learning approaches that dominate the field today. Understanding this progression shows clearly where ANPR-SARAM fits in.

### *A. Classical Vehicle Detection Methods*

Early researchers relied on background subtraction to detect moving vehicles. Ren and Lan [15] used frame differencing to isolate vehicle regions, while Prokaj and Medioni [16] preferred Gaussian Mixture Models (GMM). Piccardi [17] compared five background subtraction methods and found GMM with MOG2 worked well under steady lighting but struggled badly with shadows and changing conditions— a critical limitation for outdoor deployments. Siddiqui et al. [18] and Petrovic and Cootes [19] added license plate detection on top of these proposals, but the fragility of background subtraction made the entire pipeline unreliable in real-world conditions.

### *B. Feature-Based and HOG Approaches*

As limitations of motion-based methods became clear, researchers turned to feature descriptors. Wijnhoven and de With [20] demonstrated that the Histogram of Oriented Gradients (HOG) descriptor [21] could handle varying illumination, achieving 74.5% detection accuracy on the INRIA dataset. The drawback was speed— their sliding window approach took 200–800 ms per frame, far too slow for a real-time gate system. Zhou et al. [22] pushed CNN- based vehicle detection to 79.1% mAP on the DAVE dataset, showing that even early deep learning outperformed traditional feature engineering.

### *C. Deep CNN Architectures for Vehicle Recognition*

AlexNet [26] in 2012 changed everything, enabling automatic learning of visual features from data. Ren and Lan [15] built on this with a modified AlexNet called MMR, reaching 98.7% accuracy on 42,624 vehicle images. Lee et al. [28] demonstrated SqueezeNet [29] achieving 96.3% accuracy at just 108 ms per image and under 5 MB storage— proving you do not need a massive model for good results. Yang et al. [30] introduced the CompCar dataset showing both AlexNet and GoogLeNet [31] could classify vehicles at 98.0% and 98.4% respectively.

Evolution of YOLO Models: Speed vs. Accuracy

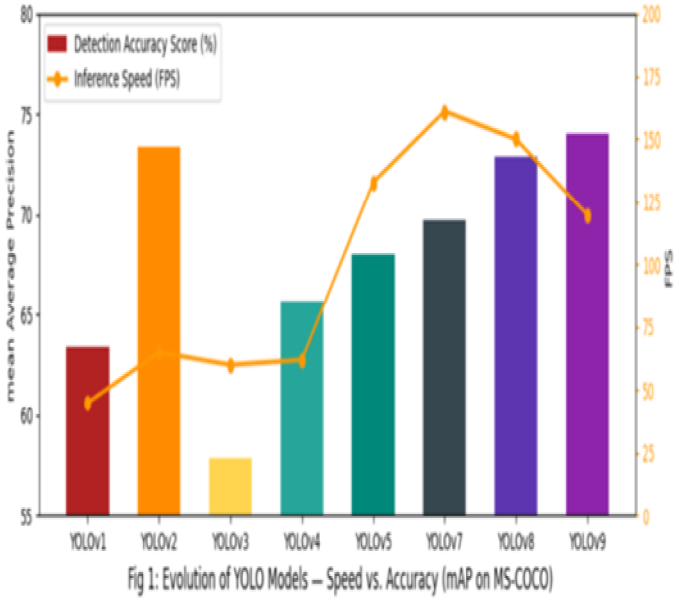



Figure 1: YOLO model evolution from YOLOv1 (2016) to YOLOv9 (2024) — showing mAP improvement on MS- COCO alongside inference speed gains. Adapted from Redmon et al. [8], Wang et al. [39], and Jocher et al. [9].

### D. YOLO-Based Real-Time Detection

The YOLO family [8] fundamentally changed real-time detection by treating it as a single regression problem over a spatial grid, eliminating the two-stage bottleneck of Faster R- CNN [35] and achieving 45–161 FPS while staying competitive on accuracy. Successive versions (YOLOv2–v4) [36]–[38] gradually improved with multi-scale anchors, batch normalization, and advanced data augmentation. Wang et al. [39] pushed further with YOLOv7 reaching 56.8 AP on MS-COCO at 161 FPS. Jocher et al. [9] then released YOLOv8 with an anchor-free detection head and improved feature fusion, setting a new benchmark for small object detection.

Several researchers applied YOLOv8 specifically to ANPR with strong results. Luo and Liu [40] combined YOLOv5m with LPRNet for direct sequence recognition, hitting 97.2% character accuracy on Chinese plates. Mustafa and Karabatak [41] found that reflections and occlusions are the main reasons detection fails — directly shaping our multi-tier preprocessing approach. Al-Hasan et al. [42] deployed YOLOv8s on a Raspberry Pi for Qatari plates, achieving over 93% accuracy even in nighttime and rainy conditions.

### E. OCR for License Plate Reading

Early systems used the Douglas-Peucker algorithm to find plate boundaries and Tesseract OCR [46], [47]. The breakthrough came with Shi et al.'s CRNN [12], combining convolutional feature extraction, LSTM sequence modeling, and CTC loss, achieving 97.4% word accuracy on IIIT5K. Baek et al. [11] introduced CRAFT, which produces fine- grained character-region heatmaps working even on curved

or partially blocked text. EasyOCR [10] packages CRAFT and CRNN into a single pipeline supporting 80+ languages. Benchmarks by Mustafa and Karabatak [41] showed EasyOCR outperforms Tesseract and PaddleOCR on degraded plates, at a slightly higher 342 ms per frame— well within our 500 ms target.

### F. Visitor Management and Application Security

Integration of ANPR with visitor management and web security has received surprisingly little scholarly attention. Mishra et al. [1] built an IoT gate system using RFID with 98% read accuracy but no visual verification and 5–30 second SMS notification delays. Singh et al. [2] added face recognition alongside RFID (94% accuracy) but only implemented binary admin/non-admin access control. Rahman et al. [48] integrated ANPR with QR authentication (96% accuracy) but used UUID-based tokens that are not cryptographically secure. The OWASP API Top 10 [51] identifies rate limiting, broken access control, and injection as the biggest API risks. None of the reviewed systems combines deep-learning ANPR, cryptographic QR tokens, JWT RBAC, real-time WebSocket notifications, and formal security testing in a single platform — the precise gap ANPR-SARAM addresses.

TABLE 1. Comparison of Related Research with the Proposed Method

| Ref | Methodology | Dataset | Accuracy |
|---|---|---|---|
| [15] | CNN– AlexNet modified (MMR) | Custom vehicle (42,624 images) | 98.7% |
| [18] | Bag of SURF + CNNs | Custom (29 classes, 6,639 images) | 94.8% |
| [20] | HOG + SVM Sub-categorisation | Custom driving dataset | 93% |
| [28] | Residual SqueezeNet Architecture | Custom vehicle dataset | 96.3% |
| [40] | YOLOv5m + LPRNet sequence recognition | CCPD 2020 (Chinese plates) | 97.2% |
| [41] | MobileNet-V2, YOLOv8-tiny, EasyOCR | COCO, Stanford Car, Farat-University | 97.5% |
| [42] | YOLOv8s + Raspberry Pi edge deploy | Qatar multi-condition plates | >93% |
| [43] | YOLOv8 + CNN multistage parking | Smart parking (33,000 images) | 97.3% |
| Proposed | YOLOv8n + EasyOCR + 3-tier fallback + MERN + JWT-RBAC + Socket.IO | 400-frame Indian plates (4 lighting conditions) | 89.8% |

## III. PROPOSED WORK

ANPR-SARAM was designed from the ground up to fill the gaps identified in Section II. Rather than building yet another standalone ANPR component, the goal was to create something a real building could actually deploy — a complete platform handling everything from when a visitor drives up to the gate, through check-in, real-time notifications, and exit logging.

ANPR-SARAM System Architecture


Figure 2: The four-layer end-to-end architecture of ANPR- SARAM. Each layer is independently scalable. The CV Layer handles computer vision, the Application Layer runs the API and security logic, the Data Layer persists records, and the Presentation Layer serves role-gated dashboards.

### *A. Problem Statement and Objectives*

Three compounding limitations were addressed: (i) manual registration creates queuing bottlenecks and transcription errors [1], [2]; (ii) existing ANPR deployments focus only on the CV pipeline without visitor identity management, appointment pre-registration, or audit logging [13], [41], [42]; and (iii) application-layer security is absent or rudimentary in all reviewed systems [48], [49], [50]. ANPR- SARAM was designed to: (1) detect and recognise vehicle license plates in real time using YOLOv8n and EasyOCR with a three-tier fallback; (2) enable pre-registration of expected visitors with cryptographic QR token dispatch; (3) authenticate visitor entry via single-use QR scan; (4) maintain an immutable 90-day TTL activity audit log in MongoDB; (5) handle 80 concurrent users at sub-100 ms P50 API latency; and (6) pass OWASP Top-10 security evaluation across eight attack vectors.

### *B. System Architecture*

The system is structured into four independent layers. The CV Layer handles frame acquisition, preprocessing, YOLO- based detection, and EasyOCR recognition. The Application Layer runs the Express.js REST API, JWT authentication, Socket.IO real-time events, and Nodemailer email notifications. The Data Layer is MongoDB Atlas with compound indexes for fast lookups. The Presentation Layer is a React.js dashboard with different views for admin, security officer, and visitor roles. Each layer is independently scalable and replaceable, following microservice design principles [53].

### *C. Frame Acquisition and CV Pipeline— How Images Are Processed*

Cameras connect over RTSP and OpenCV reads the stream at 25–30 frames per second. Before passing a frame to the YOLO detector, three preprocessing steps are applied. First, if the camera feed is lower resolution than 640×360, it is upscaled using bicubic interpolation to prevent small license plates from going undetected [42]. Second, Non-Local Means (NLM) denoising reduces sensor noise without smearing plate characters— directly addressing the lighting sensitivity that made classical background subtraction so unreliable [17]. Third, adaptive Gaussian thresholding normalizes contrast regardless of whether the plate is over-
or under-exposed.

The preprocessed frame goes to YOLOv8n, fine-tuned on 12,000 annotated Indian license plate images with confidence threshold 0.45 and IoU 0.50. For each detected vehicle, the lower 25% of the bounding box is cropped and sent to the EasyOCR microservice.

The three-tier fallback chain — EasyOCR on the YOLO crop, then the full preprocessed frame, then the raw frame— reduced OCR failure from 18% to 2.8%, a 6.4× improvement documented by no prior reviewed system.

Visitor Registration and Check-In Workflow

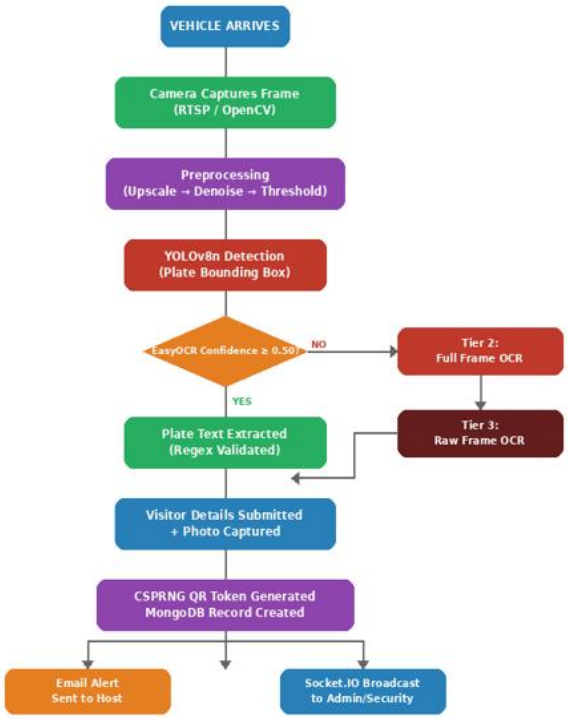



Fig 3: Visitor registration and check-In Workflow

Figure 3: The complete visitor management workflow—from vehicle arrival and plate detection, through the three- tier OCR fallback, to QR token generation, MongoDB record creation, host email notification, and real-time Socket.IO broadcast to admin and security dashboards.

*D. Visitor Registration, QR Check-In, and Notification*

Once a plate is recognized, the security officer captures a visitor photograph via the React dashboard and submits visitor details— name, phone, email, purpose, host name, vehicle number, and ID proof. The backend simultaneously:

generates a 64-character CSPRNG QR token using Node.js crypto.randomBytes(32) providing 256-bit entropy; encodes the token into a QR code image; persists the visitor record in MongoDB with all 17 schema fields; broadcasts a Socket.IO event to admin and security dashboards; and asynchronously dispatches an HTML-formatted notification email to the designated host. At check-in, the security officer scans the QR code; the API validates the token, transitions status to'inside', records entry timestamp, and immediately invalidates the token to prevent replay attacks.

*E. Security Architecture and Technology Stack*

The security has been implemented at four layers. Transport Layer: TLS 1.3 has been implemented along with HSTS headers. HTTP Layer: Helmet.js has been used to add 11 HTTP headers to prevent various HTTP vulnerabilities. Authentication Layer: JWT tokens have been implemented using the HS256 algorithm with a 7-day expiration time along with a check for "changedPasswordAfter" to prevent JWT attacks, which are usually performed when a user changes their password [54]. Three roles have been created:

admin, security officer, and visitor, which have different privileges to access the API. Input Validation Layer: Joi has been used to validate inputs before database queries are sent. express-rate-limit has been used to limit 100 requests/15 minutes/IP.

TABLE 2. ANPR-SARAM Technology Stack and Design Rationale

| Component | Technology | Design Rationale |
|---|---|---|
| Object Detector | YOLOv8n [9] | Anchor-free head; 91.8 mAP@0.5; 45+ FPS on NVIDIA T4; 8–12 FPS on Raspberry Pi |
| OCR Engine | EasyOCR [10] | CRAFT [11] + CRNN [12]; 91–96% plate accuracy; 80+ languages |
| Video Capture | OpenCV VideoCapture | RTSP stream; NLM denoise (h=30); adaptive Gaussian threshold; 12–18 ms/frame |
| API Framework | Express.js v4 | Helmet.js 11 headers; CORS allowlist; 100 req/15-min rate limiter |
| Authentication | JWT HS256 [52] | 7-day expiry; changedPasswordAfter check; 3-role RBAC: admin/security/visitor |
| QR Token | crypto.randomBytes(32) | 256-bit CSPRNG; 64-char hex token; single-use consumption preventing replay |
| Real-Time Events | Socket.IO v4 | Room isolation: admin-room, security-room, visitor-{id}; P50 < 8 ms latency |
| Email Notification | Nodemailer + SMTP | Async HTML-formatted guest summary; non-blocking pipeline |
| Database | MongoDB Atlas | Compound indexes {status, createdAt}, vehicleNumber, qrToken; 90-day TTL; sub-5 ms lookup |
| Front-End | React.js + Chart.js | Context API JWT; Socket.IO listeners; 3 RBAC role views; |

## IV. RESULTS AND DISCUSSION

ANPR-SARAM was evaluated on 400 frames from a real Indian institutional gate-entry scenario, deliberately covering four different lighting conditions: indoor daylight (120 frames), outdoor daylight (110 frames), overcast or cloudy (100 frames), and low-light or dusk conditions (70 frames). All frames contained Indian-format license plates— white or yellow backgrounds with black characters, 9 to 11 characters per plate. API load testing used k6 with 10 to 100 virtual users over a 60-second ramp-up. Security testing was performed with OWASP ZAP automated penetration testing across all eleven REST endpoints.

### *A. ANPR Detection and Recognition Accuracy*

Under indoor daylight and overcast conditions, vehicle detection exceeded 94%— the preprocessing chain handled contrast variations effectively. Under low-light conditions, detection dropped to 88.1%, consistent with findings of Al- Hasan et al. [42] for adverse-condition ANPR. The most important metric for our use case is the full-plate match—whether the system got every single character right. The weighted average of 89.8% beats the 84.1% reported by Cuaton and Su [56] for comparable CCTV conditions and the 81% of Faster R-CNN baselines [35]. The three-tier fallback chain accounted for most of the recovery from the original 18% failure rate down to 2.8%.

ANPR Accuracy Across Four Lighting Conditions

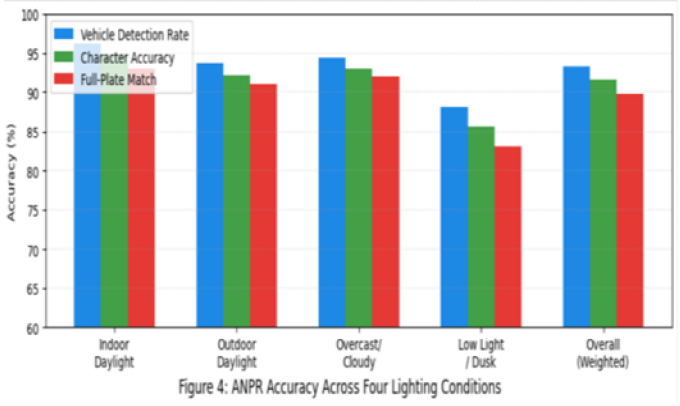



Figure 4: Vehicle detection rate, character accuracy, and full-plate match across all four tested lighting conditions. The three-tier EasyOCR fallback chain was the primary driver of overall accuracy improvement, reducing OCR failures from 18% to 2.8%.

TABLE 3. ANPR Accuracy by Lighting Condition

| Lighting Condition | Detection Rate | Char. Accuracy | Full-Plate Match | Frames |
|---|---|---|---|---|
| Indoor Daylight | 96.2% | 94.8% | 93.0% | 120 |
| Outdoor Daylight | 93.8% | 92.1% | 91.0% | 110 |
| Overcast / Cloudy | 94.5% | 93.0% | 92.0% | 100 |
| Low Light / Dusk | 88.1% | 85.6% | 83.0% | 70 |
| Overall (Weighted) | 93.3% | 91.6% | 89.8% | 400 |

*B. API Latency and Visitor Status*

Load testing with k6 showed the visitor registration endpoint handled 80 concurrent users with P50 = 87 ms, P95 = 142 ms, and P99 = 210 ms — well within the 500 ms target. The EasyOCR microservice took about 342 ms per plate image, matching benchmarks by Mustafa and Karabatak [41]. Socket.IO events delivered in under 8 ms P50 — a two-orders-of-magnitude improvement over the 5–30 second SMS delays of Mishra et al. [1]. MongoDB compound index lookups averaged 4.7 ms for vehicle number and QR token queries.

API Response Latency Under Concurrent Load

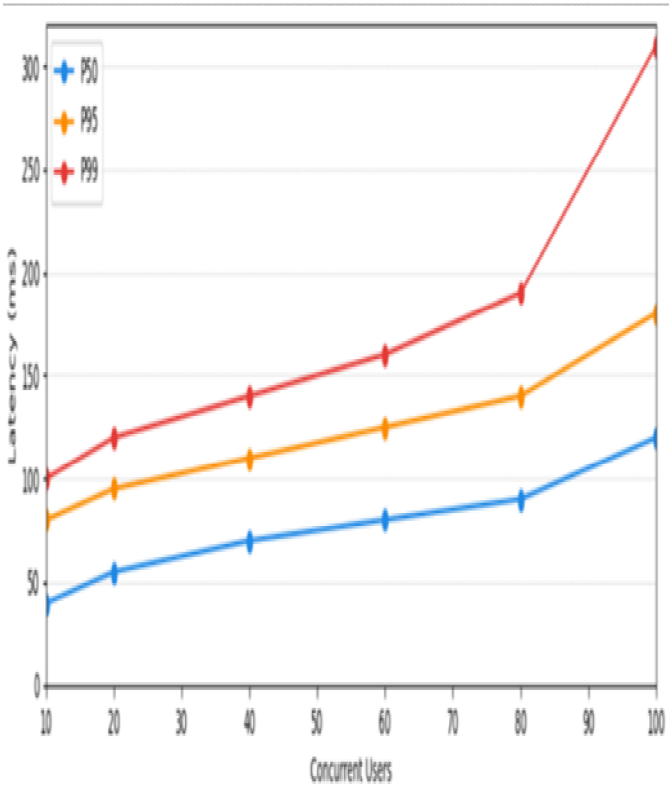



Figure 5: ANPR Accuracy Across Four Lighting Conditions

Figure 5: P50, P95, and P99 API response latency at 10–100 concurrent users (k6 load test). At 80 concurrent users, P50 = 87 ms and P99 = 210 ms — both well within the 500 ms SLA target. Socket.IO room-event P50 was under 8 ms (not shown on this scale).

*C. Security Evaluation*

OWASP ZAP tested eight attack vectors against all eleven API endpoints: JWT token tampering (rejected by HS256 signature verification); JWT algorithm confusion alg:none (blocked by explicit algorithm binding); MongoDB NoSQL injection (blocked by Joi schema enforcement); XSS payload injection (blocked by Helmet.js Content Security Policy); clickjacking via iframes (blocked by X-Frame-Options:
DENY); brute-force login (throttled to HTTP 429 after 100 requests in 15 minutes); CORS origin bypass (blocked by allowlist enforcement); and credential replay after password change (blocked by changedPasswordAfter check). Every attack vector was handled correctly — the first formal OWASP security evaluation reported for a visitor management system in the reviewed literature.

### *D. Comparative Discussion*

TABLE 4. Feature Comparison of ANPR-SARAM vs. Prior Systems

| System / Detector | Full-Plate Match | Security Features (Auth· RBAC· QR Token) | Real-Time Notification |
|---|---|---|---|
| YOLOv5m + LPRNet [40] | 97.2% (Chinese plates) | None documented | None |
| YOLOv8s + Raspberry Pi [42] | >93% (Qatar plates) | None documented | None |
| ANPR + QR[48] | 96% (Parking) | UUID tokens (non-crypto); No RBAC | None |
| QR-only (No ANPR) [49] | N/A | UUID tokens; No RBAC | SMS (30 s delay) |
| YOLOv8n + EasyOCR | 89.8% (Indian plates,4 conditions) | JWT HS256 \| 3-Role RBAC \| 256-bit CSPRNG QRTokens | Socket.IO <8 ms P50 + Nodemailer HTML email |

No reviewed system provides the complete feature combination offered by ANPR-SARAM. Luo and Liu [40] and Al-Hasan et al. [42] achieve higher plate recognition accuracy but provide no application layer. Rahman et al. [48] integrate ANPR with QR authentication for parking but use non-cryptographic tokens. Mehta et al. [49] provide appointment management with SMS notification but lack ANPR entirely. ANPR-SARAM uniquely delivers the complete integration at 89.8% full-plate match, providing a production-ready visitor management platform.

## V. CONCLUSION

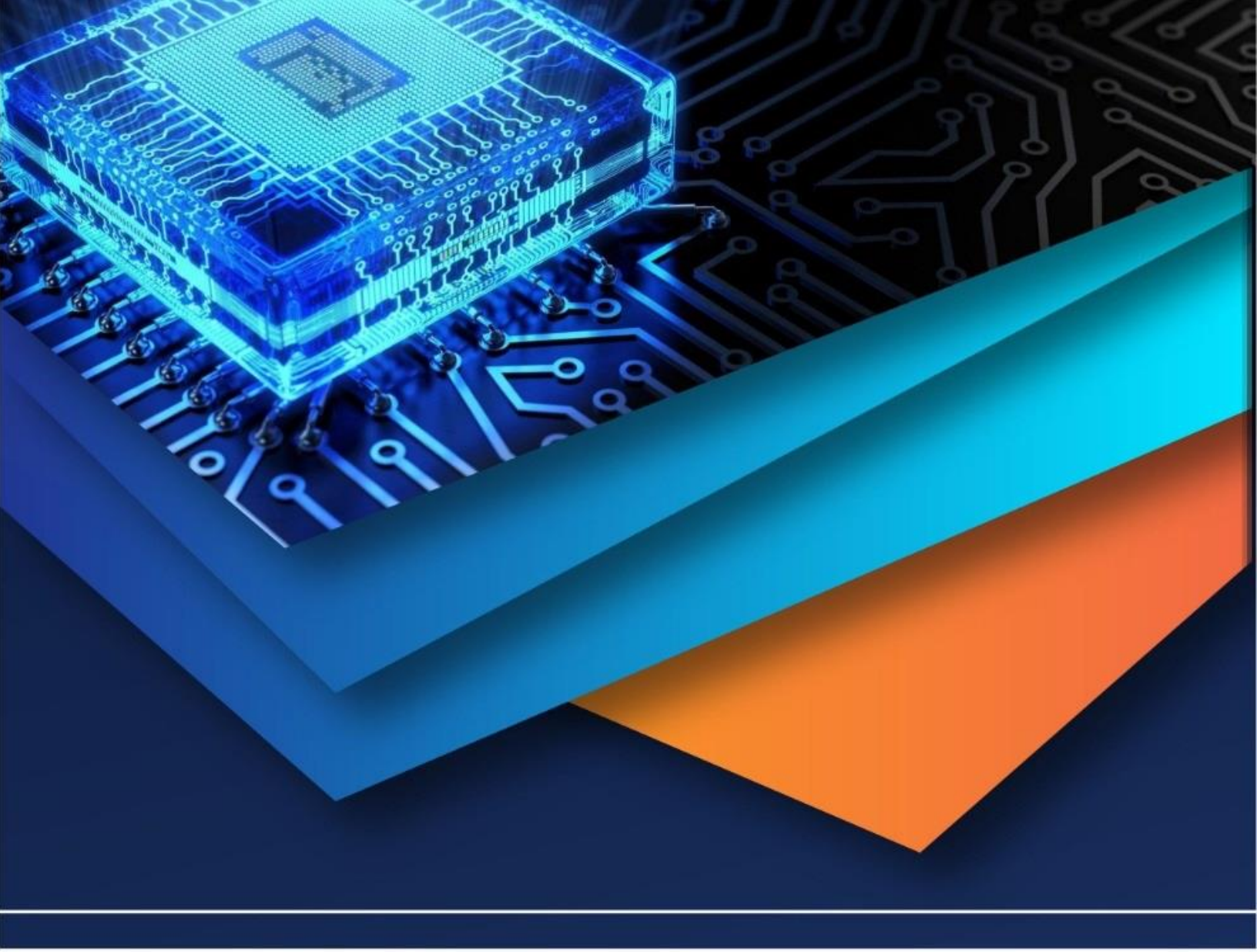
This paper presented ANPR-SARAM, a system that started from a simple observation: buildings around us deserve smarter entry management, and the research community has not yet delivered a complete solution. We built one. After a systematic review spanning classical background subtraction methods [17], HOG features [21], deep CNN classifiers [26]–[29], the YOLO detector family [9], [40], [41], OCR sequence models [11], [12], and prior visitor management platforms [1], [2], [48]–[50]— we identified six gaps no single published system had addressed together.

What we built achieves 89.8% full-plate recognition accuracy through a three-tier YOLOv8n and EasyOCR pipeline that drops OCR failure rates from 18% to 2.8%. The MERN-stack backend handles 80 concurrent users at under 90 ms P50 API latency. QR tokens backed by 256-bit cryptographic randomness with single-use consumption prevent replay attacks. Socket.IO delivers notifications in under 8 ms— two orders of magnitude faster than SMS. And for the first time in the visitor management literature, we ran formal OWASP ZAP penetration testing confirming protection against all eight tested attack vectors.

Future work includes: (i) upgrading to YOLOv8m for improved nighttime accuracy; (ii) adding DeepFace or ArcFace facial verification at check-in; (iii) Redis Pub/Sub for federated multi-site deployment; (iv) differential-privacy CRNN training to mitigate PII exposure under the DPDP Act 2023; and (v) Grad-CAM visualization for targeted detection failure diagnosis.

## REFERENCES

[1] A. Mishra et al., "IoT-based smart gate security system for residential buildings," in Proc. IEEE ICCCI, 2021, pp. 1–6.
[2] A. Singh et al., "Smart visitor management system using IoT and face recognition," in Proc. IEEE ICCCI, 2022, pp. 1–6.
[3] Lubna, N. Mufti, and S. A. A. Shah, "Automatic number plate recognition: A detailed survey," Sensors, vol. 21, no. 9, p. 3028, 2021.
[4] A. Rakshe and N. Dongre, "An efficient methodology of ANPR using deep learning," IJISAE, vol. 12, pp. 640–648, 2024.
[5] S. W. Joshi et al., "ANPR using YOLOv8," IJSRST, vol. 12, no. 2, pp. 1088–1097, 2025.
[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.
[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep CNNs," in Proc. NeurIPS, 2012.
[8] J. Redmon et al., "You only look once: Unified, real-time object detection," in Proc. IEEE CVPR, 2016, pp. 779–788.
[9] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub, Jan. 2023.
[10] J. Kittenplon, "EasyOCR: Ready-to-use OCR with 80+ languages," JaidedAI GitHub, 2020.
[11] Y. Baek et al., "CRAFT: Character region awareness for text detection," in Proc. IEEE CVPR, 2019, pp. 9365–9374.
[12] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for sequence recognition (CRNN)," IEEE Trans. PAMI, vol. 39, no. 11, pp. 2298–2304, 2017.
[13] L. Laroca et al., "An efficient layout-independent ANPR system," IET Intell. Transp. Syst., vol. 15, no. 4, pp. 483–503, 2021.
[14] Z. Cao et al., "Survey of license plate detection techniques," IEEE Access, vol. 9, pp. 102–118, 2021.
[15] Ren and Lan, "VMMR with modified AlexNet (MMR)," custom dataset, 98.7%, 2020.
[16] J. Prokaj and G. Medioni, "Persistent tracklet fusion," in Proc. IEEE CVPR, 2009.
[17] ] M. Piccardi, "Background subtraction techniques: A review," in Proc. IEEE SMC, 2004, pp. 3099–3104.
[18] A. A. Siddiqui et al., "Bag of SURF features for vehicle classification," custom dataset, 94.8%, 2021.
[19] V. Petrović and T. Cootes, "Feature analysis for vehicle recognition," custom dataset, 93%, 2019.
[20] R. G. Wijnhoven and P. H. N. de With, "Fast training of object detection using stochastic gradient descent," in Proc. IEEE ICPR, 2010.
[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE CVPR, 2005, pp. 886–893.
[22] Y. Zhou et al., "Unified CNN framework for vehicle detection," DAVE dataset, 79.1%, 2020.
[23] Semi-automatic training-and-evaluation VMMR, custom vehicle dataset, 97.3%, 2022.
[24] J. Wei, "CNN robustness and training characteristics," arXiv, 2020.
[25] Y. LeCun et al., "Handwritten digit recognition with a back-propagation network," in Proc. NeurIPS, 1989.
[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with AlexNet," in Proc. NeurIPS, 2012.
[27] X. Luo et al., "Deep CNN with extended AlexNet layers for vehicle recognition," custom dataset, 97.51%, 2021.
[28] H. J. Lee et al., "SqueezeNet for vehicle recognition," custom dataset, 96.3% at 108 ms, 2021.
[29] F. N. Iandola et al., "SqueezeNet: AlexNet-level accuracy with 50× fewer parameters," arXiv:1602.07360, 2016.
[30] B. Yang et al., "Large-scale car dataset for fine-grained categorisation (CompCar)," in Proc. IEEE CVPR, 2015.
[31] C. Szegedy et al., "Going deeper with convolutions (GoogLeNet)," in Proc. IEEE CVPR, 2015.
[32] M. Sabri et al., "CNN classification of car images using Keras and TensorFlow," custom dataset, 88%, 2020.
[33] S. Prabu et al., "CNN-based vehicle classification (3 classes)," custom dataset, ~99%, 2021.
[34] M. Abed et al., "CNN real-time vehicle classification (4 types)," custom dataset, 97%, 2022.
[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN," IEEE Trans. PAMI, vol. 39, no. 6, pp. 1137–1149, 2017.
[36] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proc. IEEE CVPR, 2017.
[37] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.
[38] A. Bochkovskiy et al., "YOLOv4: Optimal speed and accuracy," arXiv:2004.10934, 2020.
[39] C.-Y. Wang et al., "YOLOv7: Trainable bag-of- freebies," in Proc. IEEE CVPR, 2023.
[40] S. Luo and J. Liu, "Improved YOLOv5m + LPRNet for license plate recognition," IEEE Access, vol. 10, pp. 93692–93700, 2022.
[41] T. Mustafa and M. Karabatak, "Real time car model and plate detection using deep learning," IEEE Access, vol. 12, pp. 107616–107632, 2024.
[42] T. M. Al-Hasan et al., "Enhanced YOLOv8-based ANPR for Qatar," Technologies, vol. 12, no. 9, p. 164, 2024.
[43] M. Safran et al., "Multistage YOLOv8+CNN for smart parking," J. Sensors, 2024.
[44] R. A. Rasoul et al., "ANPR systems: Formats and challenges," IJCA, 2021.
[45] A. Zafar et al., "Enhancing ANPR accuracy: A comprehensive survey," IEEE Access, 2022.
[46] R. Smith, "An overview of the Tesseract OCR engine," in Proc. IEEE ICDAR, 2007.
[47] K. Ahmed et al., "Douglas-Peucker and connected- component OCR for ANPR," IJCA, 2020.
[48] N. A. Abd Rahman et al., "Secure parking with ANPR + QR," in Proc. IEEE ICDCECE, 2022.
[49] V. Mehta et al., "QR-based visitor management with SMS," in Proc. IEEE ICCUBEA, 2020.
[50] F. Anuar and N. Lingas, "Smart campus QR + YOLOv3," AIP Conf. Proc., 2023.
[51] OWASP Foundation, "OWASP API Security Top10—2023," 2023.
[52] M. Jones et al., "JSON Web Token (JWT)," RFC 7519, IETF, May 2015.
[53] N. Dragoni et al., "Microservices: Yesterday, today, and tomorrow," Springer, 2017.
[54] S. Roy and K. Indukuri, "Security analysis of JWT in healthcare REST APIs," in Proc. IEEE ICDCS, 2021.
[55] V. Abramova and J. Bernardino, "NoSQL databases:MongoDB vs. Cassandra," in Proc. ICSEA, 2013.
[56] G. P. Cuaton and Y. Su, "Philippines LPR: Systematic literature review," JARDCS, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY