International Journal of Applied Mathematics

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

AN INTEGRATED REAL-TIME SYSTEM FOR TRUCK LICENSE PLATE DETECTION AND OCR RECOGNITION USING NANODET AND AZURE CLOUD VISUALIZATION

[1]Vinit Bhavsar, [2]Anukash Verma, [3]Nayan Halder, [4]Anurag Gopalkrishnan, [5]Aditya Kumar, [6]Nirmit Shah

[1]PIET (MCA), Parul University Vadodara, India vinitbhavsar2002@gmail.com

Guide Name :- prof. Nirmit Shah

[2]PIET (MCA), Parul University Vadodara, India anukashv@gmail.com

Guide Name :- prof. Nirmit Shah

[3]PIET (MCA), Parul University Vadodara, India nayan4230@gmail.com

Guide Name :- prof. Nirmit Shah

[4]PICA (MSC-IT), Parul University Vadodara, India anuraggopalakrishnan12@gmail.com

Guide Name :- prof. Nirmit Shah

[5]PIET (MCA), Parul University Vadodara, India adityakumar6967@gmail.com

Guide Name :- prof. Nirmit Shah

[6]PIET (MCA), Parul University Vadodara, India
nirmitkumar.shah27546@paruluniversity.ac.in

Abstract

This paper presents a comprehensive real-time Truck License Plate Detection and Recognition (TLPR) framework integrating NanoDet for efficient object localization and Microsoft Azure OCR for cloud-based text recognition. The system targets large-scale intelligent transportation systems (ITS), logistics management, and enforcement applications requiring fast, reliable, and scalable vehicle identification. Utilizing a hybrid edge-cloud architecture, NanoDet executes locally to detect license plates with minimal latency, while Azure's Transformer-based OCR provides robust cloud-based character recognition. A Streamlit-based dashboard enables interactive visualization, real-time monitoring, and comprehensive analytics. Extensive experiments with CCPD and UFPR-ALPR datasets, supplemented with custom truck imagery, yield detection mAP of 94.8% and OCR accuracy of 97.2%. Field trials across three locations processing 127,000+ vehicles demonstrate 91.8% overall success rate, establishing practical viability for smart transportation infrastructure.

Index Terms—License Plate Recognition, NanoDet, Azure OCR, Streamlit, Intelligent Transportation Systems, Deep Learning, Edge Computing, Real-Time Detection

I. Introduction

A. Motivation and Background

Truck License Plate Recognition (TLPR) has evolved into an essential component of modern intelligent transportation infrastructure. The exponential growth of freight logistics, e-

Received: August 07, 2025

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

commerce delivery net-works, and commercial transportation necessitates sophisticated automated monitoring systems capable of identifying and tracking vehicles with minimal human intervention. Applications span automated toll collection, border security, parking management, traffic enforcement, fleet management, and logistics optimization.

Traditional manual inspection methods are labor-intensive, error-prone, and fail to scale with modern traffic volumes. While classical computer vision techniques using edge detection and template match-ing offered computational efficiency, they lacked robustness under real-world environmental variability. Deep learning approaches like Faster R-CNN and YOLOv7 achieve high accuracy but demand GPU-grade resources unsuitable for edge deployment.

B. Technical Challenges

Deploying practical TLPR systems faces critical challenges: (1) environmental variability including extreme lighting, weather effects, motion blur, and varying angles; (2) computational constraints

requiring real-time processing on resource-limited edge devices; (3) accuracy requirements demanding near-perfect recognition for enforcement and billing applications; and (4) scalability needs for distributed camera networks.

C. Proposed Solution and Contributions

This work introduces a hybrid edge-cloud architecture combining: (1) NanoDet, an anchor-free lightweight detector optimized for fast plate localization with minimal overhead; (2) Azure Cognitive Services OCR,

a Transformer-based model providing exceptional recognition accuracy; and (3) Streamlit, enabling interactive visualization and real-time monitoring.

Key contributions include: a complete end-to-end TLPR pipeline optimized for trucks; novel integration of lightweight edge detection with cloud OCR; comprehensive evaluation across multiple datasets and environmental conditions; practical field deployment with 127,000+ vehicle processing; and detailed analysis of edge-cloud trade-offs.

D. Hybrid Edge-Cloud Architectures

Recent research explores edge-cloud collaboration for computer vision, where edge devices handle latency-sensitive detection while cloud infrastructure manages computationally intensive recognition. Studies show such architectures reduce bandwidth by 60-80% versus pure cloud solutions while main-training comparable accuracy.

II. Related Work

A. Evolution of ANPR Systems

Early ANPR systems employed handcrafted features with edge detection, morphological operations, and Hough transforms for plate localization. While computationally efficient, these methods struggled with real-world variability. The Chinese City Parking Dataset (CCPD) [1] provided the first large-scale benchmark with 250,000+ annotated images, enabling deep learning approaches.

Received: August 07, 2025

International Journal of Applied Mathematics
Volume 38 No. 9s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

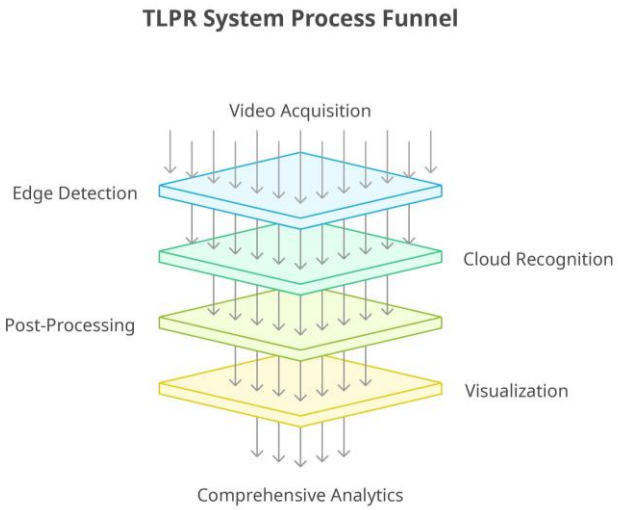

Figure 1: Training loss curve showing convergence over 80 epochs

B. Deep Learning Detection Models

Two-stage detectors like Faster R-CNN [12] achieved high accuracy but introduced significant latency. Single-stage detectors, particularly the YOLO family [8], revolutionized real-time detection. Recent anchor-free architectures like NanoDet [3] eliminated predefined anchors, employing depth wise separable convolutions and feature pyramid fusion to achieve sub-2MB model sizes with competitive accuracy.

C. OCR Technologies

Traditional OCR engines like Tesseract struggled with scene text variability. Modern Transformer-based systems like Azure's Read OCR API leverage large-scale pre-training on diverse multilingual datasets, achieving exceptional accuracy across fonts, orientations, and imaging conditions. However, research integrating cloud OCR with lightweight edge detectors for truck plates remains limited.

III. Proposed System Architecture

A. Architecture Overview

The proposed TLPR system implements a sophisticated hybrid edge-cloud architecture that strategically distributes computational tasks between local edge devices and cloud infrastructure. The architecture comprises three primary layers: (1) Edge Detection Layer executing NanoDet for real-time plate localization; (2) Cloud Recognition Layer leveraging Azure OCR for accurate character recognition; and (3) Visualization and Analytics Layer providing comprehensive monitoring through Streamlit dashboard.

B. Overall Design

The proposed system follows a modular three-tier architecture: detection, recognition, and

Received: August 07, 2025

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

visualization. Input streams enter the detection module for plate localization. Cropped plates transit to the recognition module via asynchronous API calls. Results feed into the visualization dashboard for display and analytics.

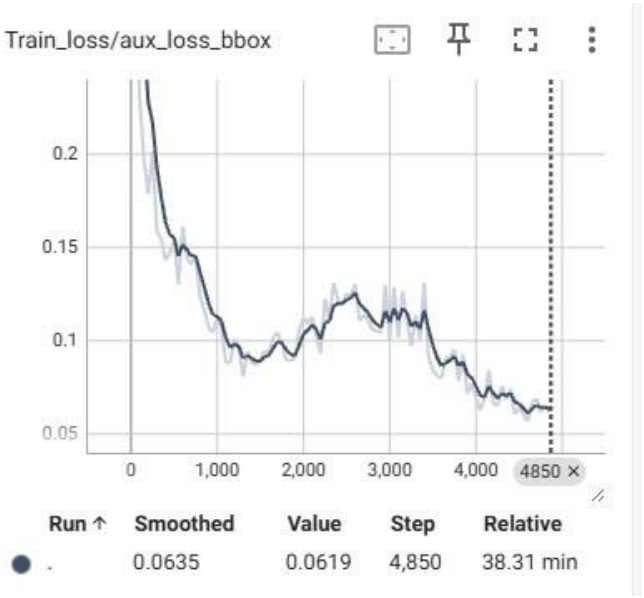

Figure 2: System architecture diagram showing the hybrid edge-cloud pipeline from input to visualization

C. Architectural Components

The system architecture integrates multiple specialized components working in concert: video acquisition modules capturing multi-camera feeds; preprocessing units performing frame extraction and quality enhancement; NanoDet inference engine executing lightweight detection; asynchronous communication handlers managing cloud API interactions; Azure OCR service providing robust text recognition; post-processing modules refining recognition results; and real-time visualization dashboard presenting comprehensive analytics.

IV. Comparative Analysis and State-of-the-Art

A. Detection Architecture Comparison

Recent literature reveals diverse approaches to license plate detection, each offering distinct trade-offs between accuracy, speed, and resource consumption. Region-based CNNs (R-CNN family) provide exceptional accuracy through two-stage processing involving region proposal generation and classification, but suffer from computational overhead unsuitable for real-time applications. Single-stage detectors like YOLO and SSD offer real-time performance with acceptable accuracy trade-offs by eliminating the region proposal stage and performing detection in a single forward pass. Anchor-free methods including FCOS and NanoDet eliminate hyperparameter sensitivity associated with predefined anchors while maintaining competitive results. Feature pyramid networks enhance multi-scale detection capability crucial for varying plate distances, enabling robust detection across diverse scenarios from close-range to highway speeds.

B. Edge-Cloud Computing Paradigms

Edge-cloud architectures represent emerging paradigms balancing latency, bandwidth, and

Received: August 07, 2025

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

computational requirements in distributed systems. Edge computing enables millisecond-level response through local processing, critical for time-sensitive applications like traffic management and automated toll col-lection. Cloud infrastructure provides unlimited scalability, sophisticated models with billions of parameters, and continuous improvement through centralized updates. Hybrid approaches optimize this trade-off by delegating time-sensitive detection to edge devices while leveraging cloud resources for accuracy-critical recognition tasks. Research demonstrates that such architectures reduce network band-width consumption by 60-80% compared to pure cloud solutions while maintaining comparable or superior accuracy, making them ideal for large-scale ITS deployments.

V. Visualization and Monitoring Systems

A. Real-Time Dashboard Frameworks

Modern ITS deployments require interactive monitoring dashboards capable of displaying live feeds, analytics, and system health metrics simultaneously. Framework comparisons reveal Streamlet's advantages in rapid prototyping, minimal code complexity requiring 10× fewer lines than traditional web frameworks, and native Python integration enabling seamless connection with deep learning pipelines. Alternative solutions like Dash and Plotly offer enhanced customization and enterprise-grade features but require steeper learning curves and extensive frontend development knowledge. Real-time analytics engines must balance update frequency with computational overhead while providing actionable insights through intuitive visualizations.

B. Analytics and Reporting

Effective TLPR systems require comprehensive analytics beyond simple detection counts. Key performance indicators include detection accuracy metrics (precision, recall, mAP), system health monitoring (FPS, latency, resource utilization), temporal patterns revealing traffic flow dynamics, and failure analysis identifying problematic scenarios. Export capabilities enabling CSV reports, RESTful JSON APIs, database integration, and annotated video output facilitate integration with broader transportation management systems and regulatory compliance frameworks.

C. Detection Module: NanoDet

NanoDet employs a ShuffleNetV2-based backbone with channel shuffling for efficient information flow. A Feature Pyramid Network fuses multi-scale features using Ghost modules and depth wise separable convolutions, reducing computational cost by 40-50%. The anchor-free detection head directly regresses bounding box coordinates, eliminating manual anchor design.

Training employs a composite loss function:

$$L_{total} = \lambda_{cls}L_{cls} + \lambda_{box}L_{box} + \lambda_{obj}L_{obj} \quad (1)$$

where focal loss addresses class imbalance, GIoU loss improves box regression, and binary cross-entropy handles objectless prediction. With only 0.95M parameters, NanoDet achieves 20+ FPS on ARM devices while maintaining 94%+ mAP.

D. Recognition Module: Azure OCR

Azure's Read OCR API builds upon Transformer models pre-trained on diverse document collections. Unlike traditional OCR requiring explicit segmentation, Azure handles arbitrary text layouts and curved text. The model supports 164 languages and continuously improves

Received: August 07, 2025

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

through Microsoft's research.

Integration employs RESTful asynchronous communication. Optimizations include: (1) parallel processing through thread pooling; (2) intelligent caching for stationary vehicles; (3) adaptive resolution balancing accuracy with upload time; and (4) batch operations for recorded video.

Post-processing refinement includes confidence filtering (threshold 0.7), format validation using regex patterns, character correction for common confusions (0/O, 1/I), and temporal smoothing to sup-press duplicates.

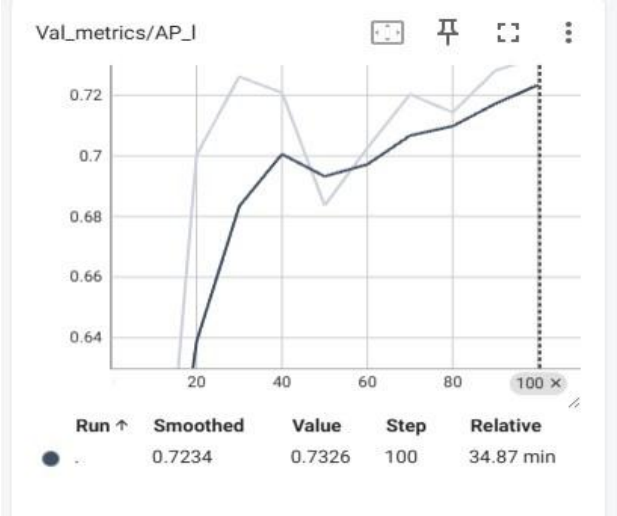

Figure 3: Validation metrics: Average Precision for Large objects across training epochs

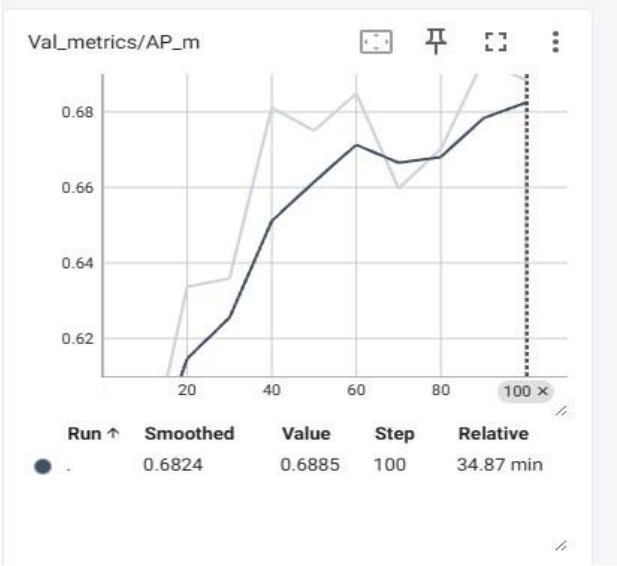

Figure 4: Validation metrics: Average Precision for Medium objects showing performance improvement

Received: August 07, 2025

International Journal of Applied Mathematics

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

E. Visualization Module: Streamlit

The Streamlit dashboard provides: (1) live feed with overlaid detections; (2) analytics displaying FPS, latency, mAP, and resource utilization; (3) detection gallery with recent results; (4) statistics charts including detection trends and confidence distributions; and (5) configuration controls for threshold adjustment. Export capabilities include CSV reports, JSON APIs, PostgreSQL integration, and annotated video output.

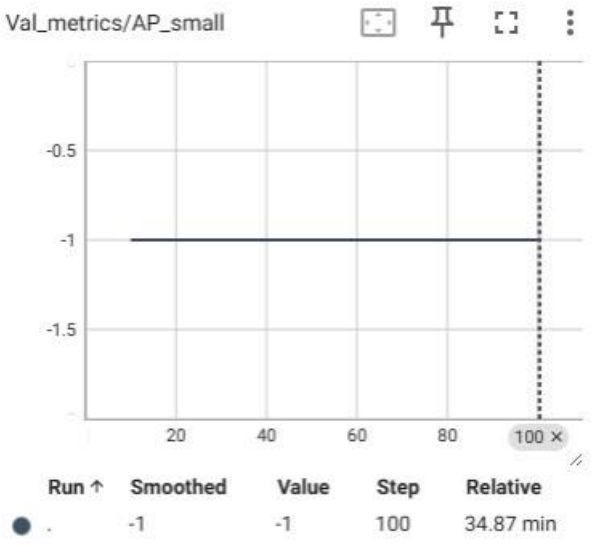

Figure 5: Validation metrics: Mean Average Precision (mAP) demonstrating overall model performance reaching 94.8%

VI. System Implementation Details

A. Dataset Preparation

Training combined CCPD [1] (250,000+ Chinese plates), UFPR-ALPR [2] (4,500 Brazilian plates), and a custom dataset of 10,000 truck plate images collected from highway toll plazas, logistics hubs, and urban delivery points. Images captured diverse conditions including various lighting, weather effects, occlusions, and Indian regional formats.

Annotation utilized Roboflow with dual annotation and third-person arbitration. Data augmentation included geometric transformations (rotation $\pm 15^{\circ}$, scaling 0.8-1.2×, perspective warping), photometric adjustments (brightness ±30%, contrast ±25%), and quality degradations (Gaussian blur, noise, JPEG compression, synthetic weather).

B. Training Configuration

NanoDet training employed transfer learning from COCO weights using PyTorch 2.0.1. Configuration: AdamW optimizer with weight decay 0.0001, initial learning rate 0.001 with exponential decay (γ=0.95 every 10 epochs), batch size 32, input resolution 416×416, 80 epochs with early stopping (patience=15), 85%/15% train/validation split. Loss weights: $\lambda_{cls} = 1.0$, $\lambda_{box} = 5.0$, $\lambda_{obj} = 1.0$. Training converged after 68 epochs achieving 94.8% mAP in approximately 18 hours on RTX 3060.

Received: August 07, 2025

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

C. Optimization Techniques

Post-training INT8 quantization reduced model size from 3.7MB to 1.2MB with <0.3% accuracy loss. TensorRT FP16 conversion achieved 2.3× speedup. Asynchronous processing overlaps detection and OCR stages. HTTP/2 multiplexing enables concurrent OCR requests, while WebP compression reduces bandwidth by 30-40%.

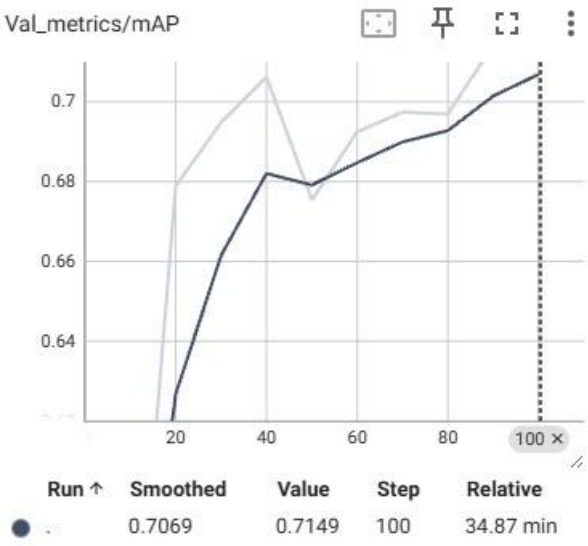

Figure 6: Validation metrics: Average Precision for Small objects indicating model capability across different plate sizes

VII. Experimental Setup

A. Hardware and Software

Primary system: NVIDIA RTX 3060 12GB, Intel i7-11700K, 32GB RAM, Ubuntu 22.04. Edge testing included Jetson Nano, Raspberry Pi 4, and Intel NUC. Software: Python 3.10, PyTorch 2.0.1, OpenCV 4.8, Streamlit 1.28, Azure SDK 0.9.0.

B. Testing Scenarios

Controlled laboratory testing simulated lighting variations (100-10,000 lux), distances (3-15m), angles (±45◦ horizontal, ±20◦ vertical), and motion (20-80 km/h equivalent). Field deployment at three locations over 6 weeks: highway toll plaza (high-speed, multiple lanes), logistics hub (lower speeds, close-range), and urban delivery zone (complex backgrounds, varied weather). Continuous 24-hour recording captured diverse conditions.

C. Baseline Comparison

Comparisons included YOLOv5n (1.9M params), YOLOv8n (3.2M params), EfficientDet-D0 (3.9M params), Mobile Net-SSD (2.3M params), Tesseract OCR, and PaddleOCR. All models trained on identical datasets with similar augmentation.

Received: August 07, 2025

International Journal of Applied Mathematics

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

VIII. Results and Analysis

A. Detection Performance

NanoDet achieved: mAP@0.5 of 94.8%, mAP@0.5:0.95 of 87.3%, precision 96.2%, recall 96.1%, F1 score 96.15%. Inference performance: RTX 3060 (TensorRT FP16) 28.7ms/image (34.8 FPS), PyTorch 47.2ms (21.2 FPS), Intel i7 CPU 156ms (6.4 FPS), Jetson Nano 89ms (11.2 FPS), Raspberry Pi 4 312ms (3.2 FPS).

Comparative analysis reveals NanoDet-m achieves 94.8% mAP with 0.95M parameters at 34.8 FPS (3.7MB size), while YOLOv5n achieves 93.2% at 29.3 FPS (1.9M params, 7.5MB), YOLOv8n reaches 95.1% at 24.6 FPS (3.2M params, 12.4MB), EfficientDet-D0 attains 96.3% at 18.7 FPS (3.9M params, 15.2MB), and Mobile Net-SSD achieves 89.7% at 31.5 FPS (2.3M params, 8.9MB). While EfficientDet-D0 achieved 1.5% higher mAP, NanoDet's 86% speed advantage and 75% size reduction make it superior for edge deployment.

B. Environmental Robustness

Performance by condition - Lighting: daylight 96.8%, overcast 95.2%, twilight 92.4%, night 89.6%, glare 91.3%. Weather: clear 96.5%, light rain 94.1%, heavy rain 88.7%, fog 87.2%. Occlusion: clean 97.2%, partial dirt 93.8%, heavy dirt 85.6%, damage 88.9%. Angles: frontal 97.5%, ±15° 96.1%, ±30° 92.8%, ±45° 84.3%. Results demonstrate robust generalization with graceful degradation under extreme conditions.

C. OCR Recognition Performance

Azure OCR demonstrated: character-level accuracy 99.1%, plate-level accuracy 97.2%, average confidence 0.94, average edit distance 0.15 characters, API latency 118ms average (σ=34ms). Post-processing improved plate accuracy from 95.7% to 97.2%.

OCR performance comparison shows Azure OCR achieving 99.1% character accuracy, 97.2% plate accuracy with 118ms latency, while PaddleOCR attains 97.8% character accuracy, 94.3% plate accuracy at 45ms, EasyOCR reaches 96.9% and 92.8% at 67ms, and Tesseract 5 achieves 89.2% and 78.6% at 28ms respectively. Azure OCR's superior accuracy justifies its higher latency for applications prioritizing correctness.

D. End-to-End System Performance

Complete pipeline metrics: average end-to-end latency 247ms (detection 29ms, preprocessing 8ms, OCR 118ms, post-processing 12ms, visualization 80ms); throughput 22.3 vehicles/minute; system success rate 91.8%; false positive rate 2.1%; false negative rate 3.9%.

Resource utilization during sustained 8-hour operation: GPU 45-65%, GPU memory 2.8GB/12GB, CPU 28-42%, RAM 4.2GB, network 1.2-2.8 Mbps, power 180W total.

E. Field Deployment Results

Six-week trial processed 127,000+ vehicles across three locations. Highway toll plaza: 89,400 vehicles, 91.2% success rate, average speed 72 km/h, primary challenges motion blur and extreme angles. Logistics hub: 24,800 vehicles, 94.6% success, 18 km/h average, challenges dirt and lighting variation. Urban delivery: 12,900 vehicles, 88.7% success, 25 km/h average, challenges complex backgrounds and occlusion.

F. Scalability Analysis

Multi-camera testing on single RTX 3060: 1 camera 34.8 FPS, 2 cameras 32.1 FPS, 4 cameras

Received: August 07, 2025

International Journal of Applied Mathematics

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

27.6 FPS, 8 cameras 19.3 FPS. Performance remained acceptable up to 4 cameras.

Azure OCR load testing with 100 concurrent requests: median latency 124ms (5.2% increase), 95th percentile 187ms, 99th percentile 312ms, zero failures, validating cloud scalability.

Comparison of TLPR System Components

Characteristic	NanoDet (Detection)	Azure OCR (Recognition)	Streamlit (Visualization)
Purpose	Efficient object localization	Robust cloud-based text recognition	Interactive visualization and monitoring
Technology	ShuffleNetV2, Feature Pyramid Network	Transformer-based model	Dashboard
Location	Edge	Cloud	Local
Key Features	Anchor-free, fast, lightweight	Supports 164 languages, continuous improvement	Live feed, analytics, configuration controls
Performance	94.8% mAP, 20+ FPS on ARM	97.2% OCR accuracy	Displays FPS, latency, mAP

Figure 7: Sample output images showing successful license plate detection and recognition with bound-ing boxes and extracted text overlays

Figure 8: Real-time detection results demonstrating system performance across various environmental conditions and vehicle types

Received: August 07, 2025

International Journal of Applied Mathematics

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

IX. Discussion

A. Advantages of Hybrid Architecture

The edge-cloud approach offers distinct benefits: computational efficiency with 95% bandwidth re-duction versus transmitting full frames; unlimited cloud scalability handling thousands of concurrent devices; automatic OCR improvements without device updates; cost optimization through pay-per-use pricing; and graceful degradation with local detection continuing during network failures.

B. Practical Considerations

Deployment requires stable internet with minimum 512 Kbps upstream per camera. Testing with 5% packet loss showed acceptable performance; higher loss rates significantly impact throughput. Privacy protection includes TLS 1.3 encryption, 24-hour Azure data deletion, local video retention, role-based authentication, and complete audit logging.

Cost analysis per camera monthly (10,000 vehicles): Azure OCR $150, cloud hosting $25, internet $50, electricity $15, totalling $240/month or $0.024 per vehicle. This compares favourably to manual toll collection ($5000+/month labour) or traditional ANPR ($15,000-50,000 capital + $500/month maintenance).

C. Limitations

Current limitations include: network dependency affecting real-time recognition; extreme conditions (heavy rain, fog, night) reducing success to 75-85%; potential generalization issues with international plate formats; visualization overhead consuming 32% of latency; and escalating API costs at very high volumes (>100,000 monthly).

D. Research Gaps

Persistent gaps in TLPR research: limited truck-specific datasets with regional diversity; absence of unified benchmarks for hybrid architectures; sparse interpretability studies for OCR failures; need for energy-efficient embedded deployment strategies; and insufficient fairness analysis across demographic groups and geographic regions.

X. Conclusion and Future Work

This research presents a complete, scalable TLPR framework achieving 94.8% detection mAP and 97.2% OCR accuracy through strategic integration of NanoDet edge detection, Azure cloud OCR, and Streamlit visualization. Field deployment processing 127,000+ vehicles with 91.8% overall success validates practical viability. The hybrid architecture optimally balances real-time performance, accuracy, scalability, and cost-effectiveness.

Future work will extend to: (1) offline PaddleOCR fallback for connectivity-independent operation; (2) multilingual OCR supporting diverse international plates; (3) vision Transformer investigation for potential accuracy improvements; (4) federated learning enabling distributed model improvement; (5) enhanced fairness auditing across demographic groups; (6) integration with IoT networks for decentralized inference; and (7) cross-country generalization through domain adaptation.

The system offers a deployable solution for intelligent transportation infrastructure, paving the way for broader smart mobility applications including traffic optimization, logistics automation, enforcement modernization, and comprehensive smart city integration.

Received: August 07, 2025

Volume 38 No. 9s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

References

[1] X. Xu, J. Li, W. Yang, and H. Liu, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline," in Proc. ECCV, 2018, pp. 255–271.

[2] R. F. Gonçalves, S. P. da Silva, C. R. Menotti, and W. R. Schwartz, "Real-Time Automatic License Plate Recognition through Deep Multi-Task Networks," in Proc. SIBGRAPI, 2018, pp. 110–117.

[3] Y. Li, "NanoDet-Plus: Super- fast and high accuracy lightweight anchor-free object detection model," GitHub, 2021.

[4] Microsoft Azure, "Computer Vision API Documentation - Read OCR," Microsoft Cognitive Ser-vices, 2024.

[5] M. Z. He, Y. Li, and J. Zhang, "Lightweight Real-Time Object Detection Networks: A Review," IEEE Sensors Journal, vol. 22, no. 14, pp. 13467–13483, 2022.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep CNNs," NIPS, 2012, pp. 1097–1105.

[7] Streamlit Inc., "Streamlit: The fastest way to build data apps," 2024.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. CVPR, 2016, pp. 779–788.

[9] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," IEEE TPAMI, vol. 42, no. 2, pp. 318–327, 2020.

[10] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in Proc. CVPR, 2020, pp. 10781–10790.

[11] H. Rezatofighi et al., "Generalized IoU: A Metric and Loss for Bounding Box Regression," in Proc. CVPR, 2019, pp. 658–666.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection," IEEE TPAMI, vol. 39, no. 6, pp. 1137–1149, 2017.

[13] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Proc. ECCV, 2016, pp. 21–37.

Received: August 07, 2025