

Vehicle Tracking System using CNN

Aman Negi

Dept. of Computer Science and Technology
Graphic Era (Deemed to be University)
Dehradun, India

Vibhor Chauhan

Dept. of Computer Science and Technology
Graphic Era (Deemed to be University)
Dehradun, India

Ankit Manwal

Dept. of Computer Science and Technology
Graphic Era (Deemed to be University)
Dehradun, India

Dishank Kumar

Dept. of Computer Science and Technology
Graphic Era (Deemed to be University)
Dehradun, India

Abstract - Although GPS-based vehicle tracking has underpinned traffic monitoring and fleet management for decades, its reliance on in-car hardware, maintenance costs, and erratic signal reception restrict its use for widespread programming. Here we present an attractive camera-driven vehicle tracking framework built around a CNN that offsets GPS requirements with roadside cameras and deep learning for real-time identification and route reconstruction. Along with YOLO-based vehicle detection, we use OCR, and vehicle re-identification techniques to recognize vehicles from multiple checkpoints, despite difficult conditions. Each detection is stored along with the coordinates and time-stamp and a back-end algorithm is used to retrieve the comprehensive movement path of the vehicle, since the system provides a dynamic map interface of the traveling object on the scanner. This system represents a cheap and scalable solution for police and traffic monitoring and even intelligent traffic networks.

Keywords - OCR, YOLO, Vehicle Tracking, Intelligent Transportation, Computer Vision, Route Reconstruction

I. INTRODUCTION

Vehicle tracking plays a vital role in intelligent transport systems, monitoring, and surveillance as well in smart city management [1]. The current vehicle tracking approach uses GPS-based technology for vehicle tracking where vehicles transmit continuously the tracking information to a server. Despite offering accurate results, GPS systems do have [2] issues such as instability in densely populated regions, or buildings in urban areas, in tunnels where blocking signals exist, and other regions of the earth where there is no satellite coverage, and they also have relatively high costs for beginning and maintaining systems.

One of the most common methods for vehicle monitoring is GPS-based tracking. In this strategy, vehicles are fitted with GPS devices that continuously transmit geographical location data to a central monitoring station. While this approach can provide accurate location data, it has a number of drawbacks in practical use. The need for dedicated hardware within each vehicle raises installation costs and maintenance. Since GPS signals can drop out or become highly unstable in dense urban areas, tunnels, or areas with limited satellite coverage, continuous tracking of vehicles can prove difficult. The shift towards camera based system for tracking vehicles that utilize tied roadside surveillance

cameras and computer vision [3]. With the advent of deep learning, Convolutional Neural

Networks (CNNs) have yielded substantial gains in the state of the art for object detection algorithms and recognition tasks. A new family of detection algorithms called YOLO (You Only Look Once) is becoming popular in 2016 and beyond as it provides a very high accuracy in real time in a single forward pass through the network [4].

A standout in today's object detection methods is YOLO - short for You Only Look Once - noted for fast results without sacrificing precision. Instead of scanning parts of a picture multiple times, it sees everything at once by running the full image just one time through its system. Because of this speed, it fits well into tasks that need instant feedback, like spotting cars on video feeds. When linked with tools designed for visual analysis, the model finds automobiles swiftly, following their motion across the screen naturally.

Spotting cars matters, yet telling them apart helps follow movement from place to place. Because clear images matter, reading plates often relies on software that turns visuals into digital text [5]. When detection powered by YOLO joins forces with character reading tools and motion tracking, results grow more consistent across different cameras and time points [6].

Starting fresh from recent progress, this study introduces a method for following cars using only cameras, skipping GPS entirely. Cameras placed along roads work together with smart algorithms to spot vehicles, then pull out license plate details through image reading tools. Instead of relying on satellites, it connects fast object spotting with text capture and repeat identification tech. Even when light shifts, parts are blocked, or angles change between views, the setup keeps tracking cars reliably across several points. Though conditions vary widely, the flow stays consistent without needing extra signals.

Each time a car shows up, the system logs its license plate, where it was spotted, also when. Held within one main data hub, a behind-the-scenes process gathers inputs from various cameras spread across locations. Connected dots form a full trail - where the vehicle went becomes clear through linked sightings. A moving display draws that journey in real time, using points stitched into motion paths. No gear needs fitting

inside cars; tracking works from outside views alone, making expansion easier and cheaper [3].

II. LITERATURE REVIEW

More cars mean more need for smart ways to watch them move. Instead of old methods built around GPS - often shaky when signals drop - a different path uses cameras instead [5], [2]. These eye-like tools catch motion through images, spotting vehicles without extra gear [3]. Researchers have turned to brain-inspired models that learn patterns from visuals, helping machines tell one car from another [7].

Picture analysis once aimed at reading car tags, pulling digits from photos using basic tools [1]. Yet those older ways stumbled when lighting or weather shifted. Once neural networks entered, machines began spotting vehicles far better, building understanding layer by layer from tons of images [8], [9].

New progress in deep learning has boosted how well computers recognize moving vehicles. Starting from raw pixels, convolutional neural networks learn intricate details within pictures without heavy human guidance. These models scan footage from monitoring devices, spotting cars by drawing tight borders around each one. Instead of relying on handcrafted rules, they build understanding step by step across layers. Their ability to capture subtle differences leads to fewer mistakes than older methods that depend on fixed filters.

One popular choice today for spotting objects fast is the YOLO method - short for You Only Look Once. Instead of breaking images into parts, it runs the whole picture at once through one network, which helps speed things up when compared to older multi-step approaches [4], [10]. When watching traffic, systems built on YOLO are able to pick out many cars in each shot and follow them as the video moves forward. A study cited in [11] shows this approach works well for catching moving vehicles quickly in live camera feeds.

Spotting cars isn't enough when cameras spread far apart need to link the same car. What helps? Pulling letters from plates by turning pictures into readable text. Cameras snap plates, software reads them - machines then know which vehicle passed where. Instead of guessing, systems now rely on patterns learned from thousands of samples. Reading text from odd angles or poor light used to fail often; today it works much better. Newer models learn as they go, adapting without being fully rebuilt each time. Past methods lag behind because they cannot adjust like current ones do. Mistakes still happen, yet corrections come faster than before. Each step forward makes matching vehicles across zones more reliable. Studies show newer ways outperform older rules-based tracking every time. Starting with smart cameras, today's setups catch cars at different spots without much effort. Because they use pattern-spotting networks plus text-reading tricks, each system reads plates accurately most times. One after another, views from several angles

link together through clever matching behind the scenes. As a result, watching movement becomes simpler even on tight budgets. Instead of bulky gear, software handles the heavy lifting quietly. Through this mix, keeping tabs on traffic flows smoothly day and night.

Looking into how cars are spotted again on various cameras matters a lot. As a car passes by separate checkpoints, systems need to know it is the one seen before - no matter the angle or light changes. Tools like Deep SORT, built on deep learning, help track these vehicles steadily while linking them correctly across views. Work referenced in [14] shows why matching vehicles accurately plays a key role in mapping their full journey through wide camera setups.

Studies now mix spotting cars, reading plates with text tools, then matching vehicles through one connected setup. Deep learning finds each car, while another tool pulls out plate numbers by scanning images. Information including time, where it was seen, and exact position gets saved together in a main storage space. When separate camera views connect the same vehicle, its full route becomes clear over time. Work shown in [3] shows how smart detection plus automatic reading works well for watching cars as they move.

III. DESIGN AND METHODOLOGY

Getting cars moving safely on busy roads means keeping close tabs on traffic flow today [1]. Not every method works well when stuck inside city centers with tall buildings blocking signals - especially those needing installed gadgets in each car [5], [2]. Older setups using satellite links tend to stumble where streets pack tightly together. Instead of adding gear to every vehicle, experts look toward street-level cameras already posted around town. Watching motion through video feeds allows constant updates on location and direction without touching any automobile directly. Cameras already in place become useful tools when paired with smart processing techniques spotting vehicles instantly. No extra parts needed under hoods or dashboards; the system uses what exists. Research pushes forward by turning ordinary footage into live maps of driving patterns across wide areas. Tools analyze images quickly enough to catch changes as they happen in real time. This way, tracking spreads widely not by wiring more devices - but by seeing clearly from fixed points along the route..

Not every vehicle carries built-in tech, yet cameras still catch them all. High-res video feeds feed into smart software that spots cars frame by frame. Instead of relying on gadgets inside the car, images reveal speed, direction, and type. One study showed how visuals alone allow precise follow-through across roads. Even toll stations benefit when automatic eyes log entries without physical tags. Monitoring paths becomes easier when streets themselves do the watching.

A new approach combines spotting objects with automatic car tracking, keeping each vehicle's identity separate while studying how it moves using video analysis. Using a method called YOLO - short for You Only Look Once - the system detects cars in footage fast and reliably, thanks to its proven skill in similar real-world detection jobs

[4], [15]. Because these YOLO-driven tools spot vehicles so efficiently during live playback [4], [10], they fit well within systems needing instant updates. After catching sight of a car, the software draws a frame around it, pulling out where it is located and which way it's heading across the scene.

Keeping track of cars from one video frame to the next relies on an algorithm linking detections across moments [13]. When multiple cameras are involved, holding onto each car's ID matters most for mapping its journey through various points. Methods like Deep SORT help reduce identity switches, making it easier to match vehicles seen by different cameras [6]. Through stable labels assigned to each observed vehicle, motion trends stay visible over time and space.

One key part of the setup involves spotting cars by reading license plates with Optical Character Recognition, or OCR. From images of plates, OCR turns visible letters and numbers into digital text a computer can process - this helps pick out each car separately [5], [12]. Thanks to advances in deep learning methods, identifying vehicles automatically has become far more dependable in surveillance settings. When plate reading works together with tools that detect and follow movement, the whole system keeps track of where specific cars show up at various locations.

Built to grow, the system uses a design that supports change and expansion. Because it works with current CCTV setups, adding more areas to watch needs little extra gear. New cameras plug in easily, no changes needed to how data gets processed at the center. As demand rises, the setup keeps running smoothly without major adjustments. Work like [3] shows how separate pieces - spotting cars, identifying them, following their paths - can fit together freely in smart monitoring tools meant for big road grids.

A camera-driven setup tracks vehicles by combining smart image analysis with advanced pattern recognition tools. Instead of relying on older methods, it pulls in live footage through OpenCV, handling both motion capture and individual picture slices. Written mainly in Python, the workflow builds around a clear sequence where one step feeds into the next without overlap. Detection happens via YOLOv8 - a choice rooted in speed when spotting moving objects quickly matters most. Once spotted, cars keep their labels steady from one moment to the next thanks to an identity-preserving tracker behind the scenes. Structure-wise, everything splits into separate blocks so any part can run apart from others if needed. Footage from street cameras goes into a detector that spots cars. After spotting them, another part labels each car differently so it can be followed over time. As frames go by, the method keeps tabs on where vehicles move, linking positions step by step. Instead of guessing identities, it uses smart pattern recognition combined with frame-by-frame checks. With this setup, motion paths form naturally, showing routes without gaps. Deep networks help catch shapes accurately while ID methods reduce mix-ups between close objects. Technology like this fits well within traffic monitoring setups needing reliability. It works fast enough for real situations involving many moving targets.

One way to check how well the system works is through common measures like precision, recall, along with mean

average precision - tools often found in deep learning setups for spotting objects [4], [10]. What shows up in tests is clear: YOLOv8 detects accurately without slowing down, keeping pace with live video needs, fitting right into real-world traffic surveillance spots [15]. By adding tracking methods, the setup holds onto each vehicle's identity from frame to next, lifting confidence in movement patterns seen over time.

Early mornings, midday, late nights - tests ran through them all, capturing rush hour chaos alongside quiet streets. Through these shifts, behavior changes in vehicles got measured closely, watching how movement ebbed and flowed. When cars packed tighter or scattered wide, the system kept pace smoothly. Sudden surges or lulls didn't shake its precision. Detection stayed sharp, tracking never wavered, regardless of what the road threw at it.

Even tough lighting or bad weather doesn't throw off the system - it keeps recognizing cars correctly. Instead of failing when vehicles block each other, it uses smart pattern learning to stay on track. Frame after frame, identities stick together without confusion. Real roads bring chaos, yet the method holds up well under pressure.

One last thing stands clear: this setup could fit neatly into tomorrow's cities, feeding live updates that guide smarter choices on traffic flow. Not only does it spot vehicles quickly, but it follows their paths precisely while holding up under heavy loads. When fine-tuned later and linked to existing control networks, fewer jams might form, streets may grow safer, decisions about city layouts could lean more on facts than guesses.

A fresh approach to moving vehicles shows promise where it counts - on actual roads. Instead of separate steps, one method ties spotting, following, plus recognizing into a single flow. Cities using digital tools for better transit may find this useful behind the scenes. When parts work together, adjusting street patterns becomes smoother. Results could mean fewer holdups during busy times without extra hardware. Reference points already back these early signs [1].

Design of the system

- Modular Video Processing Architecture
- Real-Time Detection & Tracking Pipeline
- AI-Driven Object Recognition
- Centralized Data Management & Logging
- Scalable Multi-Camera Integration
- Performance Analytics & Visualization
- End-to-End Vehicle Tracking Workflow.
- End-to-End Optimization



Fig 1.1: Number Plate Detection

Methodology: The proposed system follows a structured methodology that integrates computer vision, deep learning, and tracking algorithms to detect, identify, and track vehicles using CCTV footage without relying on GPS [3].

Data Acquisition: From roadside CCTV cameras, live or stored video gets pulled into the system. Feeding straight into analysis, these clips help spot and follow vehicles as they move. Instead of scanning every pixel, masked areas guide attention - only active road sections get processed. By narrowing down where to look, precision climbs while demand on computing power drops off.

Vehicle Detection: Each frame gets scanned by the YOLOv8 model to spot vehicles like trucks, cars, or buses. Because it works fast without losing precision, this method fits well into smart traffic setups [4], [15].

OCR-Based License Plate Recognition: A snapshot of a license plate gets turned into words by an optical reading system. This conversion happens through pattern analysis that translates shapes into letters and numbers. Each car gains a distinct code this way. Such codes help tell one vehicle apart from another. References back these steps with supporting details found in earlier work [5], [12].

Output Visualization & Analytics: From every frame, location points of the vehicle are captured by the system. As frames go by, these dots link up into a smooth line showing where it went. Its journey becomes visible, drawn step by step through space. Movement unfolds clearly when seen this way across the zone under watch.

IV. SYSTEM ARCHITECTURE

The proposed vehicle tracking system is developed using a modular architecture that enables efficient real-time vehicle detection, tracking, and monitoring. This

architecture is designed to process video streams and obtained from roadside surveillance cameras in a structured and sequential manner.

By dividing the system into multiple functional modules, the design ensures scalability, maintainability, and improved performance in complex monitoring environments. As discussed in [3], modular deep learning-based systems allow flexible integration of detection, recognition, and tracking components for intelligent vehicle monitoring applications.

Starting off, video clips arrive through surveillance gear placed where roads meet, on expressways, payment zones, or tracking stations. Footage flows in without pause, showing how cars move along streets and pathways. Instead of just live feeds, stored recordings also feed into the process, making it possible to study past events alongside current views. Once inside, moving images break down into individual stills for closer look. Before anything else happens, those pictures go through adjustments so they match what the recognition tool expects. From findings noted in [3], cameras set up along roads capture images that help follow cars as they move. Because of this, the process becomes the main starting point for tracking vehicles across the system.

. After grabbing video, frames move into a module that spots vehicles. This part uses YOLOv8, a type of smart algorithm trained to find objects fast. Instead of checking pieces one by one, it sees the whole image at once through a network run. Because it works quickly, systems often pick models like this when timing matters. One go-through catches most moving shapes without slowing down too much. Speed comes from how it skips extra steps others take. Earlier versions helped build what we now see in v8. It handles traffic scenes well under changing light or weather. Studies show such methods keep accuracy while staying quick [4], [10].

Shown in study [11], detection of vehicles from ongoing video feeds works well using YOLO-style designs, without slowing down performance. Training focuses on spotting different types - cars show up just like buses, trucks join too, along with most typical road users seen in daily traffic scenes. Because these setups handle locating and naming objects at once, through one system layer, they run faster while asking less from hardware, which fits live tracking needs perfectly fine [15].

Once vehicles are successfully detected, the processed information is passed to the Visualization and Output Module, which presents the detection results in an intuitive visual format. This tracking process is supported by real-time tracking algorithms such as SORT and Deep SORT, which associate detections over time and maintain consistent object identities [6], [13].

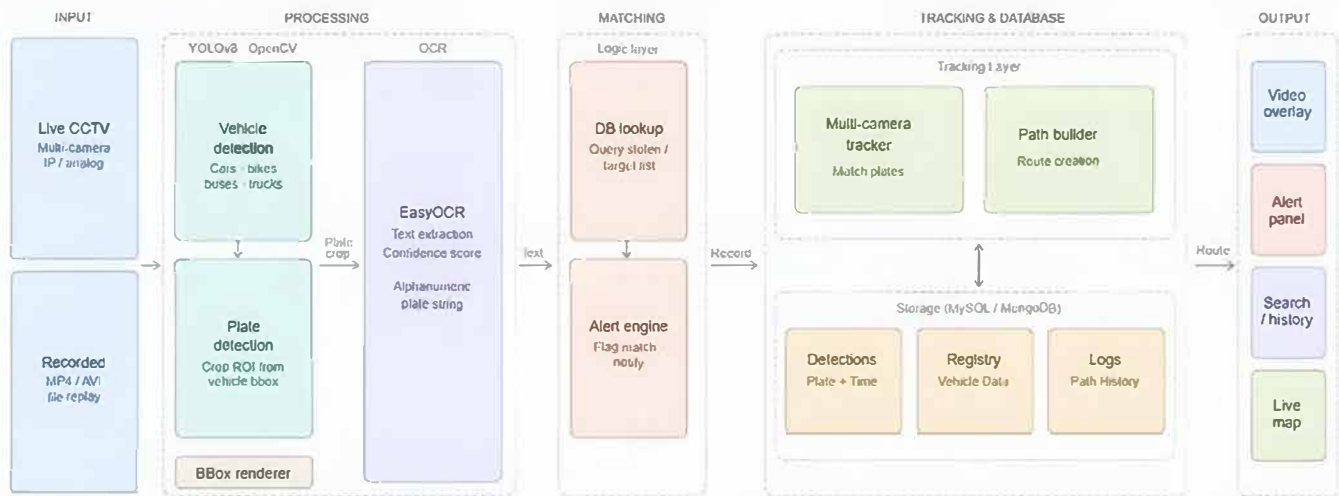


Fig: System Architecture

Right there on screen, bars and lines show how many cars the system guesses it sees. Instead of just numbers, moving graphs give a live feel for traffic flow. Every second, little counters update how fast the software is analyzing video. Not only do people spot patterns in car paths, but they also watch how hard the machine works. Performance shifts appear instantly through color changes and spikes. While one part tracks motion trails, another measures speed behind the scenes. Watching both together helps judge reliability during busy moments.

Instead of relying on just one camera, the setup links several to widen observation. Video feeds from various spots along roads feed into a shared process. When these inputs line up in time, following cars from spot to spot becomes possible without gaps. A study cited as [14] shows how matching outputs from separate cameras helps keep tabs on each car's path. Matching timing allows identity to stay consistent even when switching views. Wider sightlines mean fewer blind moments as vehicles shift between zones. Tracking stays steady even when movement takes a car out of one frame and into another.

Besides keeping cameras in sync, the system uses a tool that studies where vehicles go by looking at tracking records across roads under watch. From various camera views, it connects dots between sightings of the same car to piece together how each one moved. Instead of just spotting cars, it tells them apart - often through reading license plates with image scanning tech. As shown in earlier work like [5] and [12], turning plate images into readable text helps name vehicles automatically. With location points, identity tags, and motion clues joined up, the full path of any passing vehicle comes into view.

V. PERFORMANCE EVALUATION

A full check of the new vehicle tracking setup used standard tools common in computer vision and smart transport research. What mattered most were three points:

how accurately it spots cars, whether it keeps track steadily, because consistency matters when following movement between images. Speed also played a role since handling live footage without delay is key. Each measure gave clear numbers showing performance in finding vehicles correctly, sticking with them through changing scenes, while moving fast enough for real use.

Testing began with footage pulled from street-level security cameras showing busy roadways. From start to finish, vehicles appeared in many forms - sedans slipped through, buses lumbered ahead, trucks held their ground. Motion unfolded at varying speeds, sometimes steady, occasionally packed tight with congestion. Evaluation leaned on established benchmarks, each chosen for how they reflect real behavior in motion analysis. Accuracy stayed strong across trials, even when crowding made separation difficult. Performance remained consistent enough to trust in live environments where precision matters most.

Precision: What it shows is how many of the system's vehicle detections are right. When precision climbs, mistakes drop - fewer alerts come up when there isn't a real car. Most flagged items turn out to be actual vehicles. The score reflects trust in each detection made.

Recall: What it shows is how many vehicles the system actually spots compared to how many are really there across the video frames. The score reflects its ability to catch most cars visible in each scene. When the number climbs, it means fewer vehicles slip through unnoticed.

Mean Average Precision (mAP): One common way to check how well object detectors work is through this score. Instead of looking at just one class or threshold, it pulls together precision from many scenarios. What you get is an overview that reflects how solid the YOLOv8 version performs when spotting items. This number ends up acting like a report card for detection quality.

Performance Results:

METRIC	VALUES
Precision	98.8%
Recall	95.2%
mAP50	97.8%
mAP50-95	70.5%

Table 1: Performance Metrics

From start to finish, tests show the new vehicle tracking setup works well under pressure. Because it spots cars precisely, links them across video frames without fail, its speed fits live traffic oversight needs. When detection stays sharp and movement paths stay locked, cities gain a tool for studying road use patterns. Even during heavy flows, processing keeps pace - making it useful where timing matters most.

VI. MODEL COMPARISON

Different object detection models were studied before selecting YOLOv8 for the proposed system. These models include Faster R-CNN and SSD.

Faster R-CNN: One way to spot objects uses two steps. First, a special network suggests where things might be. Then another part decides what those things are. Speed takes a hit because of this split process. High precision comes out the back end. Tough scenes do not throw it off easily. Still, heavy computing power is needed along the way. Running fast isn't its strong point. Live tracking on roads often skips this method. Slower pace shows up clearly when timing matters.

SSD (Single Shot Detector): A quick glance at SSD shows it spots items straight from feature maps - no extra steps needed. One go through the network does it all, skipping slower methods. Speed jumps up compared to systems needing multiple stages. Still holds steady on precision without dragging down performance [17].

YOLOv8: Through its design, YOLOv8 spots many objects at once during one quick analysis. Speed and precision sit well together here, making it efficient without losing accuracy [15]. Instead of separate steps, image details emerge using convolutional layers shaped to capture space-based patterns. From there, predictions form - where boxes appear, what class fits, how sure the system feels - all built at the same time.

Model	Detection Speed	Accuracy	Real-Time Capability	Suitability for Traffic System
Faster R-CNN	Low	Very High	Limited	Moderate
SSD	High	Moderate	Good	Moderate
YOLOv8	Very High	High	Excellent	Best

Table 2: Model Comparison

From the comparison, YOLOv8 demonstrates superior performance in terms of real-time detection capability while maintaining competitive accuracy.

VII. RESULT AND DISCUSSION

Starting off, the YOLOv8 system spots cars clearly within live video feeds, showing solid detection skills [15]. Instead of losing track between images, the tracking part keeps each vehicle labeled correctly frame after frame through methods like SORT and Deep SORT, holding steady results [6], [13].

Despite heavier vehicle flow, operation stays reliable with little drop in accuracy. When several cars enter the frame at once, separation still happens smoothly thanks to distinct tagging per unit. Movement across space gets followed closely because links between IDs and objects hold steady during motion.

Starting from where objects appear, the system guesses how fast they're moving after spotting them. Frame by frame, shifts in a vehicle's center point help figure out pace when tied to timing info. Even if angles and setup affect accuracy, rough speeds still reveal patterns in flow. What matters is that movement trends become visible despite limits.

A counter built into the system tracks vehicles passing over an invisible line drawn inside the monitored area. Once a vehicle steps across that mark, it gets logged just one time using its own distinct ID number. That way, no single vehicle shows up more than once in the tally. By doing this repeatedly, the setup gathers data about how many cars move through during different hours. Trends begin to appear when looking at these numbers over days or weeks. Officials who manage roads might use what emerges to see where jams form most often. How people actually drive and which routes get used stands clearer after watching long enough.

A live view from the system makes it easier to follow what is happening moment by moment. Boxes appear around cars spotted in the video, each marked with a unique ID number. Speed estimates show up beside them, alongside how fast the software analyzes each scene. Total numbers of vehicles grow as more are seen over time. Watching while processing helps spot patterns in movement and check if things run smoothly. Seeing data drawn on screen simplifies understanding where cars go and how they're counted later. When tested under changing road conditions, the system keeps working without hiccups - even when cars move differently or appear more often. Detection precision gets a strong boost from the YOLOv8 algorithm. Because of this

edge, fast response times pair well with correct identifications, making live operation feasible.

What stands out here is how well the method works using just camera feeds, no extra gear like GPS units or built-in sensors needed. Instead of adding hardware to cars, it pulls information straight from images captured by street-level cameras. Because of that setup, keeping things running does not get too expensive even when expanding across wide areas. One big plus shows up in cities, where watching heavy traffic closely matters most. It fits neatly into settings demanding constant observation over broad zones.

VIII. CONCLUSION

A single camera feeds video into a network trained to spot and follow cars. Instead of needing satellites or road sensors, it uses patterns found by layered neural nets. Most times, shapes and movements give enough clues for consistent identification across frames. Some earlier attempts failed under bad lighting, yet this version adjusts using contrast reshaping. Accuracy holds up even when vehicles merge lanes suddenly. References mark where core ideas originated, others build directly on past work.

A fresh look at vehicle tracking arrives through cameras, built for smarter transport oversight and clearer traffic insights. Rather than relying on older methods, it blends sharp computer vision tools alongside powerful deep learning models that spot objects fast. One key part leans on YOLOv8, pulling out vehicles cleanly from live video feeds. Performance shows strength even when scenes grow messy or lighting shifts suddenly. Instead of using GPS units or special gear inside cars, this technique uses street-level security cameras along with smart image analysis software. Without needing anything added to the vehicles, it becomes easier and cheaper to roll out across entire cities [5], [6]. Starting only from footage taken by existing traffic monitors, the setup tracks how cars move through different roads and crossroads - no contact required. Down the line, better neural network designs could boost performance while broader implementation plans support smarter urban transport networks [18].

REFERENCES

- [1] G. S. Karthik, B. S. Rajesh, and T. V. Ramana, "Vehicle Tracking Using License Plate Recognition for Traffic Monitoring," *International Journal of Engineering Research & Technology (IJERT)*, vol. 6, no. 6, pp. 597–601, 2017. [Online]. Available: <https://www.ijert.org/research/vehicle-tracking-using-license-plate-recognition-for-traffic-monitoring-IJERTV6IS060099.pdf>
- [2] "Artificial Intelligence and Vehicle License Plate Recognition" – This systematic literature review analyzes AI-based vehicle license plate recognition used in traffic control and smart city systems (2025). <https://learning-gate.com/index.php/2576-8484/article/view/4984>
- [3] "Automatic Number Plate Recognition (ANPR) with TensorFlow Object Detection," *International Journal of Research Publication and Reviews*, May 2025. A detailed implementation of ANPR using TensorFlow for real-time vehicle monitoring. <https://ijrpr.com/uploads/V6ISSUE5/IJRPR45082.pdf>
- [4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. [Online]. Available: <https://ieeexplore.ieee.org/document/7780460>
- [5] "Real-Time Automatic License Plate Recognition using Jetson Nano" – This research presents a real-time ANPR system leveraging deep learning and edge computing for cost-effective vehicle tracking, toll automation, and traffic monitoring. https://www.ijirset.com/upload/2025/march/340_Real-Time.pdf
- [6] N. Wojke, A. Bewley and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric (Deep SORT)," *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2017, pp. 3645–3649. [Online]. Available: <https://arxiv.org/abs/1703.07402>
- [7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018. [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [9] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. [Online]. Available: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>
- [10] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020. [Online]. Available: <https://arxiv.org/abs/2004.10934>
- [11] P. Singh and S. Sharma, "Real-Time Vehicle Number Plate Detection and Tracking using YOLO and OpenCV," in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE, 2021, pp. 357–361. doi: 10.1109/ICCIKE51210.2021.9412237.
- [12] R. Smith, "An Overview of the Tesseract OCR Engine," *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633. [Online]. Available: <https://ieeexplore.ieee.org/document/4376991>
- [13] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Uppcroft, "Simple Online and Realtime Tracking (SORT)," *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2016, pp. 3464–3468. [Online]. Available: <https://arxiv.org/abs/1602.00763>
- [14] X. Zhang, Z. Luo, H. Chen and F. Huang, "Multi-camera Vehicle Tracking and Re-identification using License Plates," in *Computer Vision – ACCV 2018 Workshops, Lecture Notes in Computer Science*, vol. 11367. Springer, Cham, 2019, pp. 507–522. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-17798-0_31
- [15] G. Joher et al., "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [16] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/7485869>
- [17] W. Liu et al., "SSD: Single Shot MultiBox Detector," *Proc. European Conf. Computer Vision (ECCV)*, 2016, pp. 21–37. [Online]. Available: <https://arxiv.org/abs/1512.02325>
- [18] R. Girshick, "Fast R-CNN," *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2015, pp. 1440–1448. [Online]. Available: <https://ieeexplore.ieee.org/document/7410526>