Real-Time Urban Traffic Perception for Resource-Constrained Embedded Vision: A Review

Deyu Yu*

University of Sheffield, Sheffield, United Kingdom

* dyu16@sheffield.ac.uk

Abstract

Urban mobility is essential to the economic and social vitality of cities, but congestion, pollution, and security issues continue to plague urban transport systems. Conventional traffic sensing and control techniques cannot provide timely and scalable awareness at a large scale in complex urban environments, so the development of intelligence data-driven solutions is justifiable. Recent years have witnessed great strides in deep learning-based vision for tasks such as detection, segmentation, and multi-object tracking in urban areas, but deploying such models on omnipresent but resource-constrained edge hardware is a significant bottleneck. This review highlights the potential of embedded vision and lightweight deep learning for real-time traffic understanding in ITS. It surveys model compression, pruning, and compact architectures, and edge-optimised inferences, all of which, combined with edge–cloud collaborations that consider accuracy, latency, and energy tradeoff, render the model inferencing feasible at the edge. The survey also addresses practical evaluation criteria, challenges including robustness, data drift, worst case latency, safety, as well as maintainability. Finally, it outlines a set of future trends-adaptive inference, smart offloading, hardware-aware design, and standardised benchmarks pave the way for scalable, reliable, and green ITS deployment.

Keywords

Intelligent Transportation Systems (ITS), real-time urban traffic perception, embedded vision, deep learning, edge computing.

1. Introduction

Transport is vital to the social and economic development of cities because it allows the flow of people, goods, and services. However, it has become increasingly challenging to provide efficient and sustainable mobility due to rapid urbanisation, population growth, and environmental concerns. Transportation demand has long outstripped infrastructure in most developing areas, while the increasing number of private cars is overwhelming urban infrastructure. The result is congested urban roads, a serious problem that causes delays, loss of mobility, economic losses, and generates environmental pollution (such as air pollution and noise), as well as increased public health risks [1]. In response to these challenges, the concept of Intelligent Transportation Systems (ITS) has been proposed as a potential solution. ITS applies computing, communication, and sensing technologies to transportation infrastructures to promote efficiency, safety, and real-time traveler information, while addressing environmental concerns [2, 3]. In addition to the benefits of reducing traffic congestion, ITS is said to reduce passenger discomfort, improve air quality, and enhance transportation safety. ITS emerges as a promising architectural framework for enhancing traffic efficiency and safety, where the effectiveness of an ITS system primarily depends on the accurate and real-time perception of traffic. Traditional vision-based methods are usually too fragile to be applied in complex urban environments, whereas deep learning methods excel in terms of detection, segmentation, and tracking skills [4,5,6]. However, bringing these models to reality remains challenging, as the deployed devices at the edge are fundamentally limited by computational, memory, and energy constraints. Overcoming these limitations, lightweight deep learning model designs have been proposed to accommodate deep learning algorithms into resource-deprived computers, such as MobileNet [7] and ShuffleNet [8,9], which are vital for porting perception models onto resource-limited devices. Singh et al. also reveal that compression techniques can significantly mitigate the computational and memory burden [10]. Apart from on-device optimisation, near-edge offloading has been identified as an effective technique to compensate for hardware constraints [11]. All of this together forms the basis for the current methods for embedded real-time vision in low-resource environments [12]. This review explicitly addresses the combination of embedded vision and deep learning applied to real-time urban traffic perception, providing an extensive overview of the state-of-the-art, open problems, solutions, and future perspectives in this field. Gist of the developed perception models and deployment mechanisms is concluded in the light of literature review and discussions, model architecture comparison and analysis, discussion on optimisation strategies, and deployment frameworks with reviews; finally, based on reviews and discussions, this review proposes theoretical and practical perspectives that would help in adopting intelligent techniques on resource-constrained platforms. Last but not least, this work aims to provide robust, efficient, and environmentally friendly intelligent transportation applications for complex urban scenarios. Along the way, academic contributions and engineering principles are demonstrated, leading to scalable, adaptive, and more secure transportation engineering topics for future smart cities.

2. Literature Survey

Table 1 below summarizes representative studies in embedded traffic perception, including lightweight models, MobileNet-based detection, and pruning-based optimization. Each entry shows category, dataset, inference speed (FPS), accuracy (mAP), and hardware platform.

Table 1. Summary of Embedded Traffic Perception Models [13-16].

Study/Method	Approach	Benchmark Dataset	Inference Speed (FPS)	Accuracy (mAP/AP)	Deployment Platform
YOLOv7-tiny	Lightweight Model	Intersection dataset (13,000 images)	≈16	91%	NVIDIA Jetson Nano
MobileNet-v1-SSD	Lightweight Model (MobileNet-based)	HELMET + UrbanMotorbike + Self-collected (11,582 images)	≈96	63% (AP)	NVIDIA Jetson Xavier NX
YOLOv3-tiny	Lightweight Model	Bosch Small Traffic Light Dataset (BSTLD)	15	99% (classification)	Xilinx Kria KV260 (FPGA)
Pruned YOLOv4	Model Compression / Pruning	KITTI Object Detection Dataset	25.5	93.1%	NVIDIA Jetson Nano

2.1. Deep Learning for Urban Traffic Perception

Vision-based perception is a fundamental component in current ITS systems, which allows a system to perceive and recognise the traffic scene in the real world at runtime. Classic computer vision methods for traffic analysis were based on hand-tuned features but had difficulty generalising to even challenging urban scenes. Dolatyabi et al. state that deep learning models and methods have reformed traffic perception, achieving high accuracy gains and generalisation benefits as well [6]. They do so by observing that Convolutional Neural Networks (CNNs) and other deep models today outperform classical approaches in object detection, semantic segmentation, and multi-object tracking [6]. Additionally, some sophisticated architectures have been utilised in urban traffic videos, yielding promising results under diverse conditions. As El Zeinaty et al. have demonstrated, deep learning has achieved promising results in traffic perception by directly learning multi-hierarchical features from raw images, making it more robust to complex backgrounds, variations in illumination, and changes in object appearance than hand-designed methods [17]. Real-time performance remains critical for traffic perception systems, which also require running at frame rates that support real-time decision-making. State-of-the-art one-stage methods can run at real-time frame rates with comparable performance; however, they remain computationally expensive and resource-hungry, thereby limiting their use in embedded devices [17]. This issue has motivated recent studies on how to optimise deep learning models to retain high perception capabilities in edge devices with severely limited resources.

2.2. Resource Constraints and Embedded Vision Challenges

Low-power, low-memory, limited-energy embedded devices are standard, such as vehicle on-board units (OBUs), smart cameras, and roadside embedded GPUs. Csit enables large-scale deep models to be run on such hardware, giving rise to various challenges. One difficulty is that High-end convolutional neural networks (CNNs) require billions of operations per inference. When executing on embedded CPUs or low-power GPUs, simply executing these networks does not meet real-time constraints due to unacceptable latency [14]. Heavy models can still consume excessive compute and starve the frame rate, even on devices with accelerators. For example, it may not be possible to run an unoptimized YOLOv5 model in real-time on a Jetson Nano due to hardware limitations [14].

Another challenge is that the model parameters and intermediate feature maps are expensive to store on resource-constrained devices (limited RAM/flash). Models with high capacity, e.g., tens or hundreds of MB weights, which even exceed the memory or necessitate memory swaps, contribute to the slowdown of runs. As Setyanto et al. observe, a detector is based on millions of weight parameters, and these weights must also be stored in a large memory. During processing, the large data movements and operations place higher demands on computation, slowing down the system. This is even worse on microcontrollers, many of which have less than 1 MB of RAM [11].

2.3. Approaches for Efficient Deep Learning on Edge

According to Mittal and Sun et al., recent works on traffic perception for embedded platforms with limited resources have trended towards training lightweight models and exploiting knowledge distillation-based learning to strike a deployable tradeoff among accuracy, latency, and energy consumption [18, 19]. The results of the survey and comparative studies show that compact structures can retain the acceptable perception accuracy while drastically reducing the model size and memory footprint, which meets the real-time and stability demands of Jetson/ARM-class devices. This consensus has been organised and validated by a variety of benchmarks for edge-driven object detection [18,19]. Besides, Gong et al. argue that an edge-integrated training-deployment paradigm has gradually been established, where end-side data distribution, energy consumption constraints, and end-to-end latency have been considered in system deployment, jointly with hardware-constrained model optimisation and inference, in order to achieve a throughput-latency balance [20]. In the area of intelligent transportation, edge intelligence has demonstrated lower latency and higher availability in resource-constrained environments, which is the cornerstone of vehicle-road collaborative perception and online services [21]. Additionally, when a single device is not powerful enough, the latter technique employs layered/partitioned reasoning to allocate workloads between the terminal, roadside, and near-end server, and link-aware scheduling to control end-to-end latency and energy consumption [22]. Recent work has also pushed the boundary to real-time video-level detection on low-power embedded devices, indicating that the device path, combining end-device optimisation and hardware-aware reasoning for traffic, is both practical and scalable [23]. In summary, deep learning has significantly advanced urban traffic perception. However, deploying such models on embedded platforms remains challenging due to computational, memory, and power constraints. Recent research focuses on designing lightweight models, deploying hardware-aware solutions, and executing collaboratively at the edge–near the edge, which also facilitates realistic trade-offs among accuracy, latency, and energy. However, there are still deficiencies in the formalised assessment and consistent evaluation of real-world performance.

3. Methodology

3.1. Model Compression and Pruning Optimisation

For real-time target detection, the promotion of Edge YOLO [24] has significantly reduced the computational burden on edge devices through model fine-tuning and edge-cloud cooperative processing. An offline-online hybrid, lightweight detection-based approach was proposed, which modifies a YOLOv4-based model to process the computationally expensive part offline in the cloud, thereby realising efficient model inference on edge devices. This approach achieves a 60% parameter reduction and 47.3% mAP accuracy on the COCO and KITTI datasets. Meanwhile, Squeezed Edge YOLO [25] adopts the extreme model compression technology to compress the YOLO model into a parameter scale of kilobytes. This approach is well-suited for on-the-fly deployment on ultra-low-power reconfigurable hardware, such as the GAP8 microcontroller, reducing the model size by 8 and achieving an average energy reduction of up to 76% cumulatively, with up to 3.3x inference speed-up, depending on the workload, thereby fulfilling real-time capabilities for a controller in an autonomous navigation system.

3.2. Lightweight Architecture Improvements

For small object detection and segmentation in urban traffic, a dedicated, lightweight network has been proposed to strike a balance between accuracy and speed. Du et al. note that MASG-Net can modify the MobileNetV3 backbone (E-MobileNet with channel-wise attention) to transform it into the YOLOv4-tiny architecture, thereby reducing the number of parameters while improving the calibre of the extracted features [26]. This network utilises a multi-scale dilated convolutional pooling module to capture context and employs a semantic information guidance module to enrich shallow features, achieving encouraging outcomes in detecting small traffic signs against a complex background. Experiments on several traffic sign datasets demonstrate that, compared to existing lightweight models, MASG-Net achieves the best inference speed of 203.6 FPS and optimal detection accuracy, demonstrating a superior trade-off between efficiency and accuracy [26]. Moreover, another lightweight model, termed PSC-YOLO for urban road instance segmentations, is proposed, which enhances the efficiency of mask prediction by design choices such as point segmentation and grouping. The average mask accuracy of PSC-YOLO is approximately 2% higher than that of YOLOv8n-seg on urban traffic data, while achieving an inference speed of 91 FPS, which is approximately 4 times faster than

YOLOv8n-seg. This demonstrates that PSC-YOLO shows a significant improvement in real-time performance [27].

3.3. Edge-optimised Inference Strategies

Besides model architecture advancements, some works are interested in efficient inference mechanisms targeting edge devices, fully utilizing hardware capabilities to satisfy real-time needs under the constraint of scarce computing capability. For instance, HopTrack [28] uses content-aware dynamic frame sampling and a staged data association algorithm for multi-object tracking to alleviate the unnecessary frequency of detection. It runs at a frame rate of about 30 FPS and attains a mean-of-the-time accuracy of 63 % under the NVIDIA Xavier embedded platform, greatly enhancing the accuracy of previous embedded tracking methods, while having low memory and power consumption. An unsupervised video object detection method self-supervisingly utilises optical flow for motion region extraction and unsupervised clustering, thus requiring less annotated data and less complex models. This method exploits all the computational resources present in Jetson AGX Xavier and performs mixed precision inference. It is able to speed up the detection process up to 32.3 times without significant accuracy degradation and enhance energy efficiency up to 23.6 times better than an unoptimized baseline. Based on the above strategies, the deployed intelligent traffic perception system can run on limited computing, memory, and power consumption hardware in real time, no matter how the hardware conditions are, through task decomposition, heterogeneous parallelism, and precision trade-off [29].

4. Future Directions

Future research in real-time urban traffic perception on resource-constrained devices should be relatively easy, pragmatic, and aimed at solving problems that most commonly arise in real-world usage. First, examine what the objectives are for each device, e.g., target latency, energy/frame, and memory usage. These purposes are a standard or yardstick to which new models or, as in this case, changes do not immediately turn the system on its head. Second, we should not only consider average latency, but also 95th percentile (p95) latency for future systems, since it indicates how a system (with traffic to stress its devices) behaves for heavy traffic and devices under pressure. And third, the assessment should be more authentic. Short benchmark tests, ideally in tricky conditions under several regimes, help ensure that models are robust in daily use. Future systems will also need better ways to offload tasks that are shared between the device, RSUs, and nearby servers. A simple decision on where to run the model can be made using rule-based methods according to network speed, device temperature, and queue delay. Another critical issue is data drift. Frequent calibration, incremental updates, and system fallback solutions can ensure that the accuracy does not deteriorate over time. And, of course, you can never take shortcuts on safety and surveillance. Straightforward error reporting, including confidence checks, can keep the system level as quiet as possible and easy to support. On these two levels, future research could help to build lightweight, robust, and more practical traffic perception systems for real urban scenarios. These directions highlight the need for joint research efforts across computer vision, embedded systems, and transportation engineering.

5. Conclusion

Embedded vision that merged with deep learning provides a practical path to achieve scalable urban traffic perception. Recent progress in model compression, pruning, and lightweight network structures has significantly reduced the computational and memory costs for deploying high-performance perception models on low-power cameras, vehicle on-board units, and roadside machinery. Edge-optimal inference eliminates redundant work and energy usage, emphasising embedded traffic perception as a kernel of urban intelligence. All these approaches show that near-real-time detection, segmentation, and localisation can be achieved in various realistic traffic scenarios. Yet despite these advances, meaningful obstacles persist. Memory and computation resources limit the throughput and complexity of models that can be deployed, and variations in latency or the worst-case scenario of an overly congested network, bad weather, or sensor malfunctions can impact reliability. Long-term model strength also succumbs to data drift and changing urban dynamics, meaning that without periodic recalibration, incremental updating, or lightweight phone-side adaptation, even in the context of carefully pre-selected training data, model performance can degrade. To fill those gaps, however, future work needs to focus on practical, holistic solutions that are attuned to operations. In addition to average accuracy, performance should be evaluated with respect to p95 latency, energy per frame, and memory footprint on representative video streams and hardware. Adaptive inference techniques, which trade accuracy for resource savings on the fly, will enable systems to scale their computational footprint as scene complexity increases. To balance the workload on the local device, the roadside unit, and the neighbouring server, we propose an intelligent decision rule for offloading that considers the device's load, network state, and application preference. Equally important to performance are safety and maintainability. Uncertainty estimates, anomaly detection, explicit fallback strategies, and automated recalibration all serve to reduce operational risk in safety-critical systems. Summarising, the embedded traffic perception systems that have demonstrated their ability to perceive the traffic environment through the efficient model design, flexible operation, and well-coordinated edge–cloud interaction appear as promising as a lightweight and robust solution。 They have also proved to be practically deployable in real usage toward safer, efficient, and environmentally-friendly urban mobility.

References

[1] Kumarage, A. S. (2004). Urban traffic congestion: The problem & solutions. Asian Econ. Rev, 2(2004), 10-19.

[2] Ravi, S., & Mamdikar, M. R. (2022, May). A review on ITS (intelligent transportation systems) technology. In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 155-159). IEEE.

[3] Aisyah, G., & Bestari, P. (2018, November). Implementation E-Tilang in Bandung to Increase Awareness of Cross as Moral Law Passed Citizenship (Civic Virtue). In Annual Civic Education Conference (ACEC 2018) (pp. 660-663). Atlantis Press.

[4] Chen, J., Wang, Q., Cheng, H. H., Peng, W., & Xu, W. (2022). A review of vision-based traffic semantic understanding in ITSs. IEEE Transactions on Intelligent Transportation Systems, 23(11), 19954-19979. IEEE.

[5] Muhammad, K., Hussain, T., Ullah, H., Del Ser, J., Rezaei, M., Kumar, N., ... & De Albuquerque, V. H. C. (2022). Vision-based semantic segmentation in scene understanding for autonomous driving: Recent achievements, challenges, and outlooks. IEEE Transactions on Intelligent Transportation Systems, 23(12), 22694-22715. IEEE.

[6] Dolatyabi, P., Regan, J., & Khodayar, M. (2025). Deep Learning for Traffic Scene Understanding: A Review. IEEE Access.

[7] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[8] Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-shot refinement neural network for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4203-4212). IEEE.

[9] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6848–6856). IEEE.

[10] Singh, R., & Gill, S. S. (2023). Edge AI: a survey. Internet of Things and Cyber-Physical Systems, 3, 71-92.

[11] Setyanto, A., Sasongko, T. B., Fikri, M. A., & Kim, I. K. (2023). Near-edge computing aware object detection: A review, 12, 2989-3011. IEEE.

[12] Beltrán-Escobar, M., Alarcón, T. E., Rumbo-Morales, J. Y., López, S., Ortiz-Torres, G., & Sorcia-Vázquez, F. D. (2024). A review on resource-constrained embedded vision systems-based tiny machine learning for robotic applications. Algorithms, 17(11), 476.

[13] Elmanaa, I., Sabri, M. A., Abouch, Y., & Aarab, A. (2023). Efficient roundabout supervision: real-time vehicle detection and tracking on Nvidia Jetson Nano. Applied sciences, 13(13), 7416.

[14] Nocua, F., Perez-Holguin, W. J., & Pardo-Beainy, C. (2025). Urban traffic monitoring based on deep learning on an embedded GPU. Expert Systems with Applications, 273, 126847.

[15] Al Amin, R., Hasan, M., Wiese, V., & Obermaisser, R. (2024). FPGA-based real-time object detection and classification system using YOLO for edge computing, 12, 73268-73278. IEEE Access.

[16] Wang, H., Mo, R., Chen, Y., Lin, W., Xu, M., & Liu, B. (2023). Pedestrian and vehicle detection based on pruning YOLOv4 with cloud-edge collaboration. Computer Modeling in Engineering & Sciences, 137(2), 1–15.

[17] El Zeinaty, C., Hamidouche, W., Herrou, G., & Menard, D. (2025). Designing Object Detection Models for TinyML: Foundations, Comparative Analysis, Challenges, and Emerging Solutions. ACM Computing Surveys.

[18] Mittal, P. (2024). A comprehensive survey of deep learning-based lightweight object detection models for edge devices. Artificial Intelligence Review, 57(9), 242.

[19] Sun, K., Wang, X., Miao, X., & Zhao, Q. (2025). A review of AI edge devices and lightweight CNN and LLM deployment. Neurocomputing, 614, 128791.

[20] Gong, T., Zhu, L., Yu, F. R., & Tang, T. (2023). Edge intelligence in intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 24(9), 8919-8944. IEEE.

[21] Jouini, O., Sethom, K., Namoun, A., Aljohani, N., Alanazi, M. H., & Alanazi, M. N. (2024). A survey of machine learning in edge computing: Techniques, frameworks, applications, issues, and research directions. Technologies, 12(6), 81.

[22] Xu, D., He, X., Su, T., & Wang, Z. (2023). A survey on deep neural network partition over cloud, edge and end devices. arXiv preprint arXiv:2304.10020.

[23] Ruiz-Barroso, P., Castro, F. M., & Guil, N. (2025). Real-time unsupervised video object detection on the edge. Future Generation Computer Systems, 167, 107737.

[24] Liang, S., Wu, H., Zhen, L., Hua, Q., Garg, S., Kaddoum, G., ... & Yu, K. (2022). Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 23(12), 25345-25360. IEEE.

[25] Humes, E., Navardi, M., & Mohsenin, T. (2023). Squeezed edge yolo: Onboard object detection on edge devices. arXiv preprint arXiv:2312.11716.

[26] Du, C., Su, S., Lin, C., Yao, Y., Jin, R., & Hong, X. (2025). A lightweight network for traffic sign detection via multiple scale context awareness and semantic information guidance. Scientific Reports, 15(1), 10110.

[27] Gu, X., & Zhang, G. (2025). PSC-YOLO: a lightweight model for urban road instance segmentation. Journal of Real-Time Image Processing, 22(2), 1-13.

[28] Li, X., Chen, C., Lou, Y. Y., Abdallah, M., Kim, K. T., & Bagchi, S. (2024). HopTrack: A Real-time Multi-Object Tracking System for Embedded Devices. arXiv preprint arXiv:2411.00608.

[29] Guerrouj, F. Z., Florez, S. A. R., El Ouardi, A., Abouzahir, M., & Ramzi, M. (2025). Quantized Object Detection for Real-Time Inference on Embedded GPU Architectures. International journal of advanced computer science and applications (IJACSA), 16(5).