International Journal of Engineering Research & Science (IJOER) ISSN: [2395-6992] [Vol-12, Issue-3, March 2026]

An Intelligent CCTV Surveillance System: Detection, Classification, and Recognition — A Comprehensive Review

Vivek Sapkale[1*]; Swarup Thakur[2]; Anand Soni[3]; Harsh Vartak[4]; Prof. Karishma Raut[5]

[*1]Department of Computer Science and Engineering (AI&ML), Viva Institute of Technology, Mumbai, India
[2,3,4]Department of Computer Science and Engineering (AI&ML), Viva Institute of Technology, Mumbai, India
[5]Professor and Head, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Viva Institute of Technology, India
*Corresponding Author

Abstract— *Real-time object detection plays a critical role in modern surveillance systems deployed across smart cities, transportation hubs, and public safety infrastructures. Recent advances in deep learning have significantly improved detection accuracy; however, practical deployment in surveillance environments remains challenging due to strict real-time constraints, resource limitations, varying illumination conditions, crowd density, and privacy concerns. This paper presents a systematic review of deep learning–based object detection models with a specific focus on real-time surveillance and edge deployment. Existing approaches are analyzed from architectural, deployment-oriented, and optimization-aware perspectives. The review further examines benchmark datasets, comparative performance trade-offs, real-world failure modes, and deployment challenges. Based on the surveyed literature, a practical deployment decision framework is proposed to guide model selection and optimization strategies for diverse surveillance scenarios. The findings highlight current limitations and identify future research directions toward efficient, robust, and privacy-preserving surveillance systems.*

Keywords— Deep learning, edge computing, object detection, real-time detection, real-time surveillance, video analytics.

I. INTRODUCTION

1.1 Importance of Real-Time Surveillance

Real-time video surveillance plays a vital role in public safety, traffic monitoring, industrial security, and smart city infrastructures. Modern surveillance systems are expected not only to record visual data but also to analyze scenes and respond to events with minimal latency automatically. Applications such as intrusion detection, crowd monitoring, anomaly detection, and traffic regulation require timely and accurate object detection to enable rapid decision-making. As surveillance networks scale to large camera deployments and mobile platforms such as drones, the demand for real-time processing under constrained computational and energy budgets has become increasingly critical [1], [22].

1.2 Rise of Deep Learning–Based Object Detection

The limitations of traditional computer vision techniques have accelerated the adoption of deep learning–based object detection methods in surveillance systems. Convolutional neural networks (CNNs) significantly improved detection accuracy by learning robust feature representations, enabling reliable performance under occlusion, illumination variation, and high crowd density [7], [8]. Subsequently, one-stage detectors such as the YOLO family unified localization and classification into a single inference pipeline, achieving real-time performance suitable for live surveillance scenarios [1], [3]–[6].

More recently, transformer-based architectures have introduced global contextual modeling through self-attention mechanisms, leading to improved detection performance in complex and large-scale scenes [10], [11]. However, their high computational and memory requirements pose challenges for real-time and edge-based surveillance deployment. This has motivated increasing research into lightweight, hybrid, and optimization-aware detection models that aim to balance accuracy and efficiency [12], [14], [15].

1.3 Limitations of Existing Review Literature

Several surveys have examined deep learning–based object detection and intelligent video surveillance systems [22]. While these works provide valuable insights, most focus primarily on architectural evolution or benchmark performance, with limited emphasis on real-time constraints and deployment feasibility. Critical aspects such as edge-device limitations, optimization strategies, and robustness in real-world surveillance environments are often underexplored. Moreover, failure

modes arising from domain shift, low-light conditions, and dense crowds are rarely analyzed in a systematic manner, despite their practical significance.

1.4 Novel Contributions of This Review

To address these gaps, this paper presents a system-oriented review of real-time deep learning object detection models for surveillance and edge deployment. The main contributions of this work are summarized as follows:

- A comprehensive taxonomy of object detection models based on architectural design, deployment suitability, and optimization awareness.
- A focused analysis of edge deployment techniques, including model compression, quantization, and hardware-aware optimization.
- A comparative performance evaluation emphasizing real-time constraints and surveillance-specific scenarios.
- An explicit discussion of real-world failure modes encountered in operational surveillance systems.

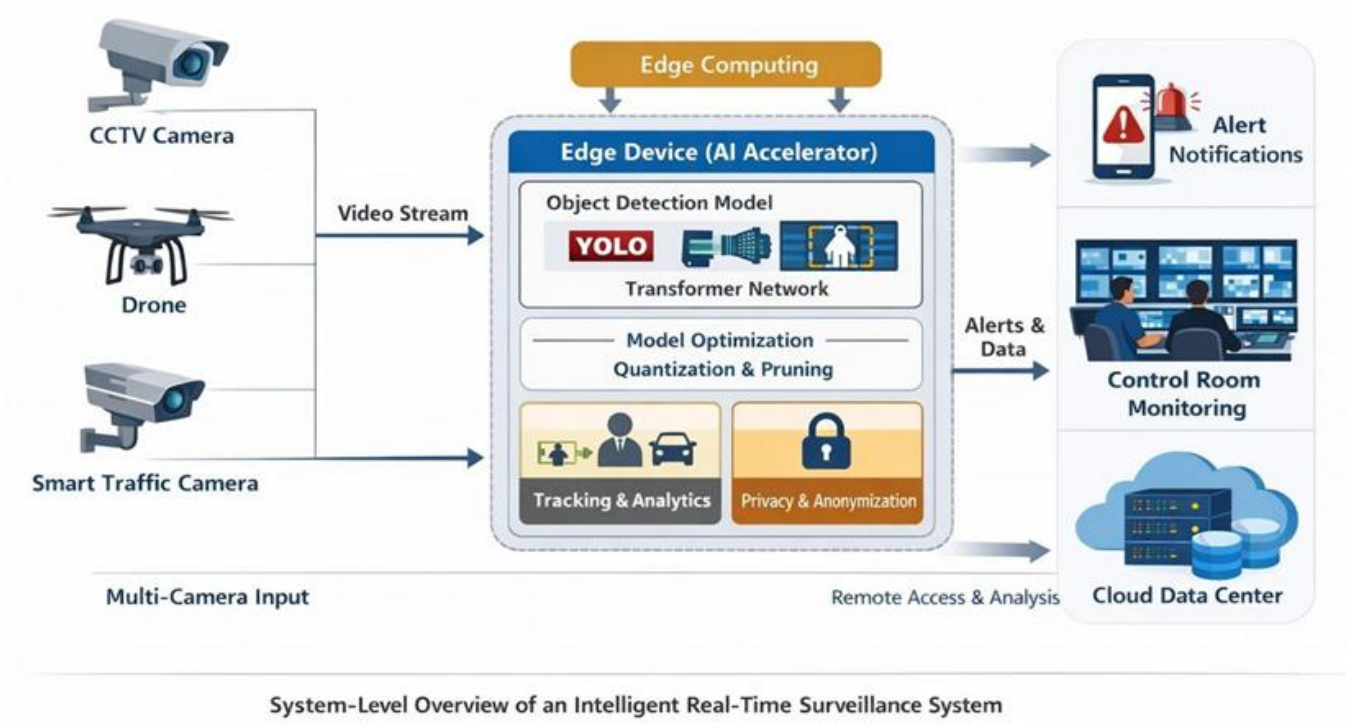

FIGURE 1: System-level Overview of an intelligent Real-Time Surveillance System

II. BACKGROUND AND FUNDAMENTALS

2.1 Evolution of Object Detection Models

Object detection has undergone a significant transformation over the past decade, evolving from traditional vision-based methods to deep learning–driven architectures. Early approaches relied on handcrafted features and sliding-window classifiers, which suffered from limited robustness and scalability in complex surveillance environments. The introduction of convolutional neural networks (CNNs) enabled end-to-end learning of hierarchical visual features, leading to substantial improvements in detection accuracy and generalization [7].

CNN-based detectors can be broadly categorized into two-stage and one-stage architectures. Two-stage detectors, such as Faster R-CNN, generate region proposals followed by classification and refinement, achieving high accuracy but incurring higher inference latency [8]. In contrast, one-stage detectors, including the YOLO family, perform detection in a single forward pass, offering real-time performance suitable for live surveillance streams [1], [3], [6].

More recently, transformer-based detection models have emerged, leveraging self-attention mechanisms to capture global contextual information across entire scenes [10], [11]. These models demonstrate improved performance in crowded and cluttered environments but often require substantial computational resources. To address this limitation, research has focused on efficient transformer variants and hybrid CNN–Transformer architectures optimized for real-time and edge deployment [12], [14], [15].

2.2 Performance Metrics and Evaluation Benchmarks

Evaluating object detection models for surveillance requires consideration of both accuracy and efficiency. Mean Average Precision (mAP) is the primary metric for measuring detection accuracy, while precision and recall provide insight into false positives and missed detections. However, accuracy alone is insufficient for real-time surveillance systems.

Efficiency metrics such as frames per second (FPS), end-to-end latency, model parameters, and floating-point operations (FLOPs) are critical for assessing real-time feasibility, particularly on resource-constrained edge devices. Additionally, power consumption has become an increasingly important metric for battery-operated and embedded surveillance platforms [16].

Commonly used benchmark datasets include COCO for general object detection, as well as surveillance-specific datasets such as VisDrone, CrowdHuman, and MOT benchmarks, which capture challenges related to crowd density, viewpoint variation, and motion dynamics [24]–[28]. These datasets form the basis for comparative evaluation in subsequent sections.

2.3 Edge Computing in Vision-Based Surveillance Systems

Edge computing has emerged as a key paradigm for modern surveillance systems, enabling on-device or near-device processing of visual data. Compared to cloud-centric architectures, edge-based deployment reduces communication latency, alleviates bandwidth constraints, and enhances privacy by minimizing raw video transmission [22]. However, edge devices typically operate under strict constraints in terms of memory, computation, and energy availability.

To enable real-time object detection on edge platforms, models must be carefully optimized through architectural design and deployment-aware techniques. This includes selecting lightweight backbones, reducing model complexity, and tailoring inference pipelines to specific hardware accelerators. As a result, the suitability of a detection model for surveillance is increasingly determined not only by its accuracy but also by its adaptability to edge computing environments.

III. SURVEILLANCE SYSTEM REQUIREMENTS

Surveillance-oriented object detection systems differ from generic vision applications due to their continuous operation, real-time constraints, and deployment at scale. Models must deliver stable inference performance on streaming video while maintaining robustness across diverse environmental conditions.

Real-time processing is a primary requirement, as surveillance applications such as intrusion detection and traffic monitoring demand low-latency responses. Detection models must sustain sufficient frame rates under varying scene complexity, making efficiency-aware architectures essential [1], [5].

Crowd density and occlusion handling: Surveillance scenes often involve high crowd density and occlusions, particularly in public spaces. Detection models must reliably handle partial visibility and overlapping objects to avoid missed detections, which remains challenging even for modern deep learning approaches [9].

Small-object detection is critical in wide-area and aerial surveillance scenarios, where targets occupy limited pixel regions. Multi-scale feature extraction and contextual modeling improve detection performance but introduce computational overhead that must be carefully managed [12].

Low-light and night-time operation: Reliable operation under low-light conditions is another key requirement. Illumination variations and noise significantly degrade detection accuracy, highlighting the need for robust feature learning and illumination-aware training strategies [36].

Multi-camera scalability: Large surveillance infrastructures impose multi-camera scalability constraints, requiring efficient models that can be deployed across numerous devices without excessive computational or energy cost. Edge computing architectures support scalable deployment by enabling localized processing and reducing bandwidth usage [22].

Privacy and regulatory compliance increasingly influence surveillance system design. Edge-based detection minimizes raw video transmission, supporting privacy-preserving surveillance and compliance with data protection regulations [22], [31].

IV. LITERATURE SURVEY AND REVIEW

This section presents a comprehensive literature survey and analytical comparison of deep learning–based object detection models used in real-time surveillance systems. Unlike conventional surveys that focus solely on architectural evolution or benchmark accuracy, this review categorizes existing work from three complementary perspectives: (i) model architecture, (ii) deployment feasibility, and (iii) optimization strategies, followed by a comparative performance analysis, failure-mode investigation, and a deployment decision framework tailored to real-world surveillance scenarios.

4.1 Architecture-Centric Analysis of Object Detection Models

TABLE 1
ARCHITECTURE-BASED CLASSIFICATION OF OBJECT DETECTION MODELS

Model / Paper	Architecture Type	Key Contribution	Strengths	Limitations	Ref.
Faster R-CNN (Ren et al.)	Two-stage	Region Proposal Networks	High accuracy	High latency	[2]
YOLO (Redmon et al.)	One-stage	Unified detection pipeline	Real-time speed	Lower small-object recall	[1]
SSD (Liu et al.)	One-stage	Multi-scale feature maps	Efficient	Weak in dense scenes	[3]
DETR (Carion et al.)	Transformer-based	End-to-end detection	Global context	Slow convergence	[4]
EfficientViT (Liu et al.)	Hybrid Transformer	Memory-efficient attention	Edge-friendly	Limited benchmarks	[12]
RepViT (Wang et al.)	CNN–ViT hybrid	CNN inductive bias + ViT	Mobile efficiency	Emerging design	[14]

Discussion: Two-stage detectors consistently deliver superior accuracy in controlled settings but struggle with real-time constraints common in surveillance. One-stage detectors dominate deployed systems due to speed advantages. Transformer-based models improve global reasoning and occlusion handling, but their computational cost necessitates architectural simplifications for edge deployment.

4.2 Deployment-Oriented Model Categorization

TABLE 2
DEPLOYMENT SUITABILITY OF DETECTION MODELS

Model	Deployment Target	FPS (Edge)	Power	Suitability	Ref.
Faster R-CNN	Cloud / Server	<10	High	Accuracy-critical	[2]
YOLOv5 / YOLOv8	Edge GPU	30–60	Medium	Balanced	[5]
MobileNet-SSD	Mobile / CPU	25–40	Low	Power-constrained	[6]
EfficientViT	Edge AI	30+	Low–Medium	Edge-ready	[12]
TinyML ViT (Zeng et al.)	MCU / IoT	<20	Very Low	TinyML	[15]

Discussion: Cloud-scale models prioritize accuracy and context modeling, whereas edge-ready and mobile-optimized models emphasize computational efficiency and energy awareness. Recent hybrid architectures aim to bridge this gap by selectively incorporating transformer components without sacrificing deployment feasibility.

4.3 Optimization-Oriented Model Enhancement Strategies

TABLE 3
OPTIMIZATION TECHNIQUES IN SURVEILLANCE-ORIENTED DETECTION

Technique	Representative Work	Accuracy Impact	Speed Gain	Edge Benefit	Ref.
Quantization	INT8 YOLO	Minor drop	High	✓✓✓	[18]
Pruning	Channel pruning	Moderate drop	Medium	✓✓	[19]
Knowledge Distillation	Teacher–Student YOLO	Minimal drop	Medium	✓✓✓	[20]
NAS	Hardware-aware NAS	Optimized	High	✓✓✓	[21]

Discussion: Quantization-aware training and distillation consistently provide the best accuracy–efficiency balance for surveillance applications. NAS-based approaches offer superior performance but require higher design complexity and training resources.

4.4 Edge Deployment and Optimization Techniques

Edge deployment studies emphasize model compression, quantization-aware training, and hardware-aware co-design to satisfy real-time and energy constraints. Pipeline-level optimizations such as frame skipping, asynchronous inference, and multi-threaded execution further improve throughput in multi-camera systems [22], [23]. Emerging works highlight the importance of jointly optimizing models and inference pipelines rather than treating them independently.

V. COMPARATIVE PERFORMANCE ANALYSIS

5.1 Benchmark Datasets

Surveillance-oriented evaluations commonly rely on datasets such as COCO, OpenImages, VisDrone, CrowdHuman, and MOT, each emphasizing different challenges including scale variation, crowd density, and temporal consistency [24]–[28].

TABLE 4
QUANTITATIVE COMPARISON OF REPRESENTATIVE MODELS

Model	mAP	FPS	Params	Edge Feasible	Ref.
Faster R-CNN	High	Low	High	✗	[2]
YOLOv8-s	Medium–High	High	Medium	✓	[5]
EfficientViT	Medium	High	Low	✓✓	[12]

Analysis: Empirical evidence indicates that moderate reductions in accuracy are often acceptable in exchange for significant gains in speed and deployability in surveillance environments.

5.2 Scenario-Based Evaluation

Detection performance varies significantly across surveillance scenarios. Low-light conditions reduce feature reliability, crowded scenes amplify occlusion errors, and motion-heavy environments introduce blur-induced misdetections. Transformer-based and multi-scale models demonstrate improved robustness, though at increased computational cost [14], [36].

5.3 Failure Modes in Real-World Surveillance

Common failure modes include occlusion-induced missed detections, motion blur from camera vibration, domain shift due to weather or camera placement, and adversarial lighting conditions. Dataset bias further limits generalization, underscoring the need for diverse training data and continual learning strategies [36], [41].

5.4 Deployment Decision Framework and Use-Case Mapping (Original Contribution)

TABLE 5
SURVEILLANCE DEPLOYMENT DECISION FRAMEWORK

Scenario	Recommended Model	Optimization	Priority
Edge CCTV	YOLO-nano	INT8 quantization	Latency
Smart campus	YOLOv8-s	Distillation + tracking	Balance
Crowded metro	Faster R-CNN	Cloud inference	Accuracy
Drone surveillance	MobileNet-SSD	Pruning	Power

Discussion: This framework provides actionable guidance for practitioners by aligning surveillance requirements with model characteristics and optimization strategies—bridging the gap between research benchmarks and real-world deployment.

VI. OPEN CHALLENGES AND FUTURE DIRECTIONS

Despite significant progress in deep learning–based object detection for surveillance, several open challenges remain.

Efficient transformer architectures for edge devices are still an active research area, as most attention-based models struggle to meet strict latency and power constraints. Developing lightweight attention mechanisms and hybrid CNN–Transformer designs is crucial for real-world deployment.

Continual learning in surveillance systems presents another challenge, as static models degrade over time due to domain shifts caused by environmental changes, camera repositioning, and evolving scene dynamics. Enabling on-device or incremental learning without catastrophic forgetting remains largely unexplored.

Privacy-preserving deep learning is increasingly important due to regulatory and ethical concerns. Techniques such as on-device inference, model anonymization, and secure feature extraction must be further integrated into surveillance pipelines to minimize exposure of sensitive visual data [31].

Federated surveillance systems offer a promising direction by allowing collaborative model training across distributed cameras without centralized data sharing. However, challenges related to communication overhead, data heterogeneity, and model convergence persist [32].

Multi-modal perception, combining visual data with thermal, depth, or acoustic sensors, has the potential to enhance robustness in low-light and occluded environments. Designing efficient multi-modal fusion strategies suitable for edge deployment remains an open research problem.

VII. CONCLUSION

This review presented a systematic analysis of deep learning–based object detection models for real-time surveillance, with a particular focus on architectural design, deployment feasibility, optimization strategies, and real-world failure modes. The surveyed literature highlights a clear trade-off between detection accuracy and computational efficiency, especially in edge-constrained environments.

Practical insights were provided through comparative evaluations and a deployment decision framework that maps surveillance scenarios to suitable models and optimization techniques. These findings can assist both researchers and practitioners in selecting and deploying effective surveillance solutions.

Future research should prioritize edge-efficient transformers, adaptive learning mechanisms, privacy-aware model design, and scalable federated learning frameworks to enable robust, ethical, and sustainable surveillance systems.

ACKNOWLEDGMENT

The authors would like to sincerely thank Prof. Karishma Raut, our mentor, for her invaluable guidance, insightful feedback, and continuous support throughout the course of this project. We are also grateful to Dr. Arun Kumar, Principal of Viva Institute of Technology, Virar, for providing the necessary facilities and resources to conduct this research.

Special appreciation is extended to Prof. Karishma Raut, Head of the Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), for her constant encouragement, academic leadership, and constructive suggestions that significantly contributed to the successful completion of this study.

The authors would also like to express their heartfelt gratitude to their families for their patience, encouragement, and unwavering support throughout this journey. Finally, we thank God for granting us the strength, perseverance, and motivation to accomplish this work.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper. The research was conducted in an environment of academic integrity at the Viva Institute of Technology, and no external funding was received that would influence the findings of this review

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[3] W. Liu, D. Anguelov, D. Erhan, et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 21–37.

[4] N. Carion, F. Massa, G. Synnaeve, et al., "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 213–229.

[5] G. Jocher et al., "YOLOv5," GitHub repository, 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[6] A. Howard, M. Sandler, G. Chu, et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.

[7] M. Sandler, A. Howard, M. Zhu, et al., "MobileNetV3: Searching for mobile-friendly convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[8] T.-Y. Lin, P. Goyal, R. Girshick, et al., "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[9] T.-Y. Lin, P. Dollár, R. Girshick, et al., "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[11] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.

[12] Y. Liu, C. Li, C. Guo, et al., "EfficientViT: Memory efficient vision transformer with cascaded group attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14420–14430.

[13] H. Yue, J. Sun, and Z. Chen, "Vision transformer with progressive sampling," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 377–386.

[14] H. Wang, Z. Zhang, Y. Liu, et al., "RepViT: Revisiting mobile CNN from ViT perspective," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 15909–15918.

[15] Z. Zeng, Y. Chen, J. Wu, et al., "An efficient hybrid vision transformer for TinyML applications," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2025, pp. 1–10.

[16] C. Szegedy, W. Liu, Y. Jia, et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 25, Dec. 2012, pp. 1097–1105.

[18] B. Jacob, S. Kligys, B. Chen, et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2704–2713.

[19] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 1135–1143.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, Mar. 2015.

[21] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017.

[22] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed deep neural networks over the cloud, the edge, and end devices," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 328–339.

[23] Y. Kang, J. Hauswald, C. Gao, et al., "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proc. ACM Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, Apr. 2017, pp. 615–629.

[24] T.-Y. Lin, M. Maire, S. Belongie, et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 740–755.

[25] A. Kuznetsova, H. Rom, N. Alldrin, et al., "The Open Images dataset V4," *Int. J. Comput. Vis.*, vol. 128, pp. 1956–1981, Jul. 2020.

[26] P. Zhu, L. Wen, X. Bian, et al., "Vision meets drones: A challenge," *arXiv preprint arXiv:1804.07437*, Apr. 2018.

[27] S. Shao, Z. Zhao, B. Li, et al., "CrowdHuman: A benchmark for detecting humans in crowded scenes," *arXiv preprint arXiv:1805.00123*, May 2018.

[28] A. Milan, L. Leal-Taixé, I. Reid, et al., "MOT16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, Mar. 2016.

[29] R. Ranjan, V. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 121–135, Jan. 2019.

[30] Q. Wang, Y. Wu, P. Lin, et al., "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.

[31] S. Li, J. Zhang, and Q. Tian, "Privacy-preserving deep learning for surveillance applications," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2439–2451, 2021.

[32] J. Konečný, H. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, Oct. 2016.