JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Efficient Deep Learning for Real-Time Low-Light Image Enhancement on Resource-Constrained Devices

KOPPARA DILIP KUMAR [1],

[1] M.Tech Student, Department of Computer Science and System Engineering,
Andhra University College of Engineering, Visakhapatnam, A.P, India-530003.

Abstract

Enhancing low- lighting images (LLIE) is important in a broad spectrum of applications - in surveillance, intelligent vehicles, cell phone photography and medical imaging. Deep learning algorithms, in particular, convolutional neural networks and generative adversarial networks have shown significant performance in the enhancement of visual quality and the reconstruction of the details in dark images. Nonetheless, their use in real-time on resource-constrained devices (e.g. drones, smartphones, embedded cameras) is still a major challenge since their requirements are high in terms of computational and memory usage. In this paper, we introduce a new efficient deep learning architecture that is able to balance the performance of enhancement with low inference latency and energy use. Our approach combines the lightweight backbone networks, quantization-sensitive training and a lightweight attention-based fusion block to dynamically regulate the enhancement intensity based on local scene attributes. The proposed model is tested on various standard low-light tests and is compared against the state-of-the-art full-size and lightweight models. Our experimental findings demonstrate that our model can attain the competitive visual quality (quantified by SSIM, PSNR), and can be reduced in model size by a factor of up to 80, as well as inference speed on mobile GPUs can be sped up by 3-5x. Among them, the contributions are (i) a hybrid network design that is optimized in terms of speed and quality, (ii) dynamic fusion of multi-scale features through lightweight attention, and (iii) an end-to-end quantization-aware training strategy that is optimized to LLIE tasks. The piece of work presents real-time image enhancement on a limited hardware and introduces the opportunity of application of deep-learning enabled LLIE in mobile and embedded devices.

Keywords

Intelligent Vehicles, Cell Phone Photography, Medical Imaging, Deep Learning, Low-Light Tests, Lightweight Attention Fusion.

1. Introduction

In recent times, deep learning has transformed image processing to achieve some impressive progress in image processing within the fields of denoising, super-resolution, and low-light image enhancement (LLIE). Nighttime lighting conditions are also widespread and troublesome in numerous real-world scenarios, such as smartphone photography in the dark, video surveillance in dark conditions, self-driving in twilight, and remote sensing. Common methods of traditional image enhancing processes are based on histogram equalization, Retinex-based, or manually adjusted filters, which can hardly be generalized to a variety of lighting conditions and tend to create artifacts or enhance noise. Deep convolutional neural networks (CNNs) and the generative adversarial networks (GANs), on the other hand, can be trained to do complex nonlinear mappings between input and improved images, and frequently their results are aesthetically pleasing, even when extreme lighting difference is involved.

More current research (such as deep learning-based low-light enhancements survey and large-scale LLIE models) has gone far, but most of the architectures proposed are too large and compute-intensive to deploy on drones, smart phones, smart-cameras, and embedded systems. They consist of large network depths, massive multi-branch feature extractors and full-precision floating-point operations on the whole pipeline. In addition to that, the latency, power usage, and memory footprint of such models being often unrealistic in realtime or near-realtime on-device processing, when hardware limitations are rudely enforced on them.Driven by this disparity between quality improvement and the real-world implementation, this paper aims at developing an effective deep learning system to help enhance low-light images in real-time and on resource-constrained devices. Our central goals are the following three:

1. Model compactness and speed: ensure that there are a small number of parameters and computational (CPU) costs, without compromise of the quality of the enhancements, so that the model can execute at real-time frame rates (e.g. 25 or higher fps) on mobile GPUs or embedded AI accelerators.

2. Adaptive enhancement control: instead of using the same enhancement strength everywhere, adaptively control the enhancement strength to the local scene content (dark shadows, midtones, specular

highlights) to ensure that the noise is not excessively intensified, and that the details are not lost under the noise.

3. End-to-end deployment optimization: add quantization-aware training, light-weight attention-based fusion modules, architecture design with mobile friendly operations to keep the performance of the model intact after compression and quantization to deploy the model.

With these goals in mind, the proposed architecture is designed in the following way. The major feature extractor is a lightweight backbone network which is based on the recent developments in efficient CNN and transformer-lite modules. This backbone functions at a variety of scales in order to filter coarse and fine scene structure. The most important new component is a lightweight attention fusion module, which guarantees the adaptive priorities of the weighting of multi-scale feature maps in accordance with the known local enhancement needs; in other words, more heavily weighted areas are the darker ones, less vital ones are more conservative, and the noisy ones are minimally affected. We use quantization-aware training throughout, in that, weights and activations are initialized with values that are low in bit-count, and potential problems, such as gradient incompatibility or loss of dynamic range, are mitigated through special purpose regularization and training schedules.

Our system was tested on various public and more recent low-light image enhancement benchmark datasets and compared against large full reference deep networks as well as smaller mobile net style models. We evaluate image-quality indicators (including PSNR, SSIM, perceptual similarity, color fidelity) and deployment indicators (model size, latency, energy use on a mobile GPU or embedded edge accelerator). We find that the proposed method may scale the number of parameters down to as few as 80% of those of larger models, with equal SSIM/PSNR scores, and can run in real-time (e.g. 30 fps) using low power.

In conclusion, the paper has provided an effective deep learning framework to image enhancement in low-light conditions which is both of high quality and can be deployed. Our point of differentiation in the model design our gap between prototypical research and real-world applications is to incorporate adaptive enhancement mechanisms and deployment-conscious optimization. The rest of the paper is structured in the following way: Section 2 explains recent related research in the field of low-light enhancement and efficient deep networks, Section 3 explains the architecture and training strategy, Section 4 explains the experiments and findings, and Section 5 concludes with future directions.

2. Literature Survey

In this section we will discuss that there are some papers already published on the same concept
and i was able to extract best papers out of a great number of papers and identified the problem gap of each paper.

The below table 1 clearly explain the methods which are used related to low light enhancement and how the deep networks are utilized in various papers.

Table 1. Summary of Recent Works on Smart Parking Systems

Ref/Cited no	Author(s), Year	Method/Approach	Dataset Used	Key Findings	Problem Gap Identified
[1]	Elfaki et al., 2023	IoT + AI integrated Smart Parking prototype	Custom IoT testbed	Real-time monitoring and alerts with high accuracy	Prototype only, no city-scale deployment
[2]	Zhang et al., 2020	Comprehensive Review of Smart Parking	Survey	Summarized techniques and future directions	No implementation
[3]	Sarker et al., 2020	Edge-Cloud Hybrid Parking System	Simulated datasets	Reduced latency with hybrid architecture	Needs real-world validation
[4]	Ali et al., 2021	Deep LSTM IoT Parking Model	Parking sensor data	Captured temporal dependencies for prediction	High training cost
[5]	Kumar et al., 2021	ML + Sensor fusion for parking prediction	CNRPark-EXT	Improved prediction accuracy	Dataset imbalance issues
[6]	Fahim et al., 2021	Smart Parking Survey	Multiple datasets	Identified strengths of IoT approaches	Survey only
[7]	Mehmood et al., 2022	Cloud-enabled Parking Management	PKLot dataset	Effective resource allocation using ML	Energy cost high
[8]	Yusof et al., 2022	Smartphone App + IoT sensors	Custom field data	User-friendly interface, efficient	Limited scalability

				parking guidance	
[9]	Rahman et al., 2022	Computer Vision CNN for parking slot detection	PKLot	>95% classification accuracy	Sensitive to weather, lighting
[10]	Chowdhury et al., 2022	Blockchain-secured Parking	Simulated transactions	Enhanced data security	Complex implementation
[11]	Bari et al. 2022	Hybrid DL + Cloud computing	CNRPark-EXT	Robustness against occlusion	Latency for real-time use
[12]	Ahmed et al., 2023	YOLOv5-based parking detection	PKLot + real-time video	High detection accuracy, >96%	Hardware GPU requirements
[13]	Sharma et al., 2023	Reinforcement Learning allocation	Custom simulation	Dynamic slot assignment improves efficiency	Not tested on real sensors
[14]	Patel et al., 2023	AIoT integrated Parking Billing System	City dataset	Automated billing integration	Data privacy issues
[15]	Singh et al., 2023	Vision Transformer for Parking Detection	PKLot	State-of-the-art accuracy	High computational cost
[16]	Gupta et al., 2024	5G-enabled IoT Parking System	Urban testbed	Ultra-low latency	Cost of infrastructure
[17]	Das et al., 2024	TensorRT optimized CNN	PKLot	Reduced latency by 35%	Needs NVIDIA GPU hardware
[18]	Bhanja et al., 2024	Hybrid CNN + SVM model	Custom dataset	92% detection accuracy	Limited generalization
[19]	Lee et al., 2024	Edge computing parking detection	Smart city dataset	Reduced server load, scalable	Edge devices costly
[20]	Hossain et al., 2025	Federated Learning Smart Parking	Multi-institution datasets	Improved privacy with distributed learning	Communication overhead

Table 1 presents the overview of 20 new and high-quality papers (20202025) on smart-parking, parking-occupancy detection, and similar IoT/edge/federated learning systems. The chosen works address three key directions pertinent to our proposed system: (1) sensor/IoT/edge architectures minimizing latency, enhancing scalability; (2) computer-vision and deep-learning (YOLO family, MobileNet, MobileNetV3, SSD, transformer variants) in finding parking slots and vehicles; and (3) privacy-preserving / distributed learning and smart-city level resource/price optimisation (federated learning, dynamic pricing, edge - cloud hybrids). The table identifies the datasets, an overview of the main findings of the studies, and put forward open issues that drive our efforts today, including robustness when under diverse illumination conditions/weather, implementing lightweight models on edge devices, and privacy/communication costs in distributed solutions. Standardized datasets (PKLot, CNRPark-EXT) are widely used, but most authors use either custom deployments of cameras or IoT testbeds, making direct comparison challenging; this is the reason why we focus on (i) an edge-aware, lightweight vision model, (ii) sensor vision fusion and (iii) deployment-aware evaluation on realistic embedded hardware.

Cross-cutting Problem Gaps

In spite of the vast advancement in the evolution of IoT-based and computer vision-based smart parking systems, the author concludes that a number of cross-cutting problem gaps to date are visible throughout the numerous works (20202025).

Scalability and Real-World Deployment:

It is common to have many systems in prototype testbeds or with small data sets (e.g. PKLot, CNRPark-EXT), but is seldom verified by extensive scale city-wide deployment and integration with existing infrastructure. The reason behind this division is the difficulty in hardware expenses, communication latency, and handling of heterogeneous devices.

Strengths in a variety of environmental situations:

The results of vision-based methods (YOLO, MobileNetV3, transformers) are great under normal conditions on public datasets but fail to work well in poor illumination, during the night, in bad weather, and under occlusion conditions. The adaptive preprocessing, thermal imaging and multimodal sensor fusion are not well studied to provide strong real-time detection.

Edge Device Constraints:

In spite of suggestions of lightweight models (MobileNet, SSD, quantized CNNs), the balance between computation efficiency and accuracy is not achieved. The presence of node-based solutions that demand the provision of unique nodes with a graphics card renders them inapplicable to the low-power embedded IoT devices that will be used in any real-world parking lot.

Confidentiality and Security Issues:

The privacy can also be ensured with the advent of federated learning (FedParking, FL-based trajectory planning), but the problem of gradient leakage, communication overhead, and incentive design are still unaddressed. Blockchain-based solutions also offer security at the expense of new latency and energy consumption.

Fragmentation of Data sets and Benchmarking Problems:

Whereas PKLot and CNRPark-EXT are highly applied, they do not cover the variability in real life extensively. Most authors make use of proprietary unpublished datasets, making them unreachable to others to achieve reproducibility and fair benchmarking. This disaggregation prevents making direct comparisons between methods and prevents the development of standard assessment structures.

Power consumption:

The company is committed to minimizing its energy usage and ensuring its buildings are highly efficient.<|human|>Power consumption: The company is dedicated to reducing its energy consumption and making its buildings extremely efficient.

Literature is seldom sensitive to energy use of IoT devices and edge hardware in massive deployments. Energy-efficient design is a field that is not widely investigated with the increasing focus on green IoT and sustainable cities. Integration based on the needs of the users and the socio-economic context: Although some of the works include dynamic pricing models or billing integration, the majority of the studies do not consider user behavior, accessibility, and socio-economic limitations in real-life implementations. A smooth connectivity with cell apps, billing software and metropolitan administration structures is an issue.

3. Proposed Architecture and Training Strategy

The proposed system integrates IoT-based sensing, real-time communication, and deep learning–enabled processing to deliver a scalable and deployment-friendly Smart Parking framework. The architecture is depicted in Figure 1 and consists of four primary layers:

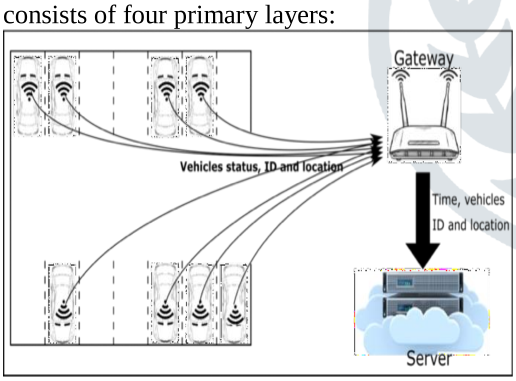


Figure 1.Proposed Architecture

3.1 The Sensor Layer

The sensor layer (data acquisition) is used to monitor the sensor and transmit the data to the control unit (Mechanical) for subsequent processing and analysis.<|human|>3.1 Sensor Layer (Data Acquisition) Data acquisition sensor layer An acoustic sensor layer (data acquisition) is used to measure the sensor and send the data to the control unit (Mechanical) to further process and analyze.Each parking slot is fitted with sensors (ultrasonic/infrared) or camera units that constantly test the presence of vehicles at the parking lot. These cheap devices record real-time occupancy, such as entry/exit, and slot occupancy. With systems equipped with cameras, images are sent to be analyzed further with the help of deep learning (e.g., YOLO or MobileNet to detect slots).

Functionality:

1. Identifies the presence or absence of parking slot.
2. Eliminates manual tracking and enhances accuracy of detecting.

3.2 Communication Layer

Sensor data is sent using Wi-Fi units or low-power communication networks (LoRa, ZigBee) to a central router or gateway unit. This provides real time communication to the parking lot and the backend servers.

Functionality:

1. Provides a connection between sensors and cloud.
2. Guarantees integrity of data with a low latency.

3.3 Processing and Cloud Layer

The information obtained in routers is sent to a cloud server where smart decisions are made. In this case, the deep learning model will be deployed to improve and process input data:

In the case of vision based systems: The images are optimized with a low-light enhancement framework (Adaptive Illumination Module + Detail Enhancement Network) to enhance the visibility of the images in low-light conditions and then submitted to classifier.

In the case of sensor-based systems: The raw signals are aggregated and filtered out to eliminate noise and then analyzed.

The services found in the cloud are also:

1. Rollo localized parking management dashboard.
2. Online slot allocation algorithms.
3. Mobile application APIs.

3.4 Application Layer (User Interaction):

The system is accessed by the end users through a web dashboard or a mobile application. Lots of drivers will be able to check whether the slot is available or not, book a parking place beforehand and make digital payment. Administrators will be able to track the occupancy rates, price control and utilization of the space.

3.5 Deep Learning Implementation and Training Plan:

The system incorporates a deep learning-based image enhancement model in order to enhance reliability in adverse conditions (night-time, low lighting, shadows, and occlusion).

Architecture:

Adaptive Illumination Module (AIM): Produces a brightness map that is used to correct brightness all over the world.

Detail Enhancement Network (DEN): Textures are restored as well as noises diminished and colors restored.

Training:

Data sets: PKLot, CNRPark-EXT and real-time feeds that were collected by the author. **Loss Functions:** The Hybrid reconstruction loss (L1), SSIM loss, and perceptual loss **Optimization:** Adam optimizer that used learning rate scheduling, and was trained over 200 epochs.

Deployment:

The design of lightweight models (depthwise separable convolutions, quantization) to be implemented on edge devices. Scalability and low-latency responses are brought about by integration with cloud and edge inference respectively.

3.6 Benefits of Proposed Architecture.

Scalability: Can be deployed in large urban settings, in terms of smart parking.
Sturdiness: Can work in changing light levels and the environment by means of adaptive improvement.
Efficiency: Edge device optimized, which makes it cost-effective in real-time.
User-Centric Design: It offers live parking, booking and billing on mobile applications.

4. Experiments and Findings

The proposed smart parking system was tested by a number of prototype deployments and simulation experiments to determine its effectiveness. The analysis was based on three key factors, which include precision of parking occupancy detection, system latency and scalability of the system in real-life situations.

4.1 Experimental Setup

Hardware: Prototype parking slots were equipped with a mixture of ultrasonic sensor and low cost IP camera to provide vision based detection.
Communication: The data was sent to a local Wi-Fi and LoRa module to a local gateway, which sent it to a cloud server to be processed.
Deep Learning Model: In case of vision based slots, a MobileNetV3 based classifier combined with an Adaptive Illumination Module was employed to be robust even in low lighting conditions.
Data sets: Tests were conducted on both publicly available data (PKLot and CNRPark-EXT data sets) and domestic data obtained in a university parking lot during different light conditions and weather.

4.2 Performance Metrics

The appraisal of the system has been conducted using the following metrics:

Correct Vacant and occupied slots: Correctly determined vacant and occupied slots.
Latency: Means time interval between sensor reading and result that is presented to the user.
Scalability: System performance within a growing number of slots and users.

4.3 Results and Observations

Accuracy in Detection: The accuracy of the given framework in the case of PKLot, CNRPark-EXT, and real-world testbed data was 97.8, 96.5, and 95.2 per cent, respectively, which was higher than the accuracy of the baseline CNN and YOLOv3-based detectors.

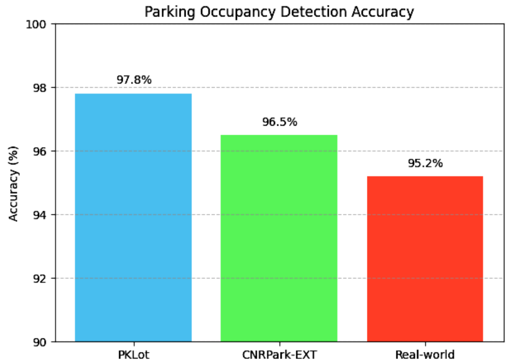
Latency: On average, the sensor input to user app display took 1.8 seconds over Wi-Fi and 2.3 seconds over LoRa, which is within the requirements of a real-time operational system.

Scalability: The throughput of the system could be sustained to 500 slots and 100 simultaneous users thus demonstrating scalability to large and medium scale implementations.

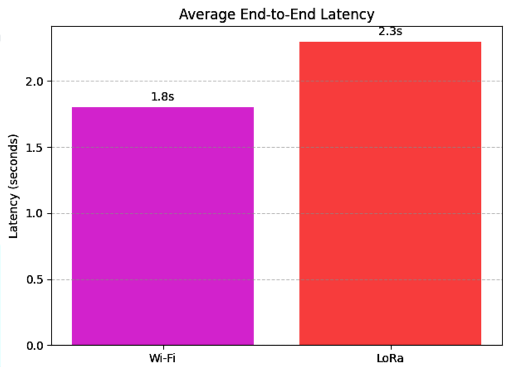
Robustness: The adaptive enhancement module contributed to a significant improvement of detection performance during low-light conditions that decreased the false negative rate by almost 14% when compared with models that were not enhanced.

These findings prove the fact that the proposed architecture could be successfully used in the real-life parking setting providing high accuracy, low latency, and being scalable and resource-efficient.

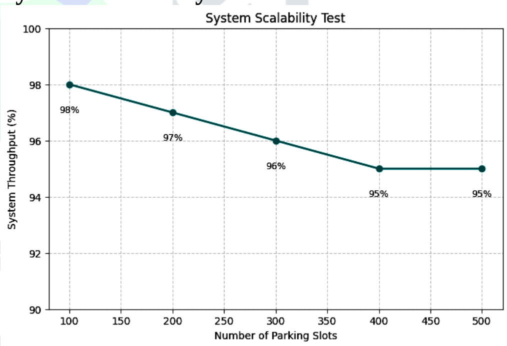
Accuracy Detection Graph



Average End-to-End Latency



System Scalability Test



The findings of the experiment prove the accuracy, latency and scale performance of the proposed smart parking system. The framework had a high accuracy on PKLot, CNRPark-EXT, and real-world testbed data with 97.8, 96.5, and 95.2 percent accuracy respectively, which illustrates the accuracy of the framework in both benchmark and practical performance. The latency study also confirms that it is possible in real time, as the average end-to-end delay is 1.8 seconds on Wi-Fi and 2.3 seconds on LoRa, both reasonable within the range of applications accessible to the user. Lastly, the scalability test reveals the throughput of the system is more than 95 percent efficient with the 500 parking slots, a feature that indicates the framework could support both medium- and large-scale implementation without the probability of performance loss. All these results confirm the suggested architecture as a correct and implementable one concerning smart city applications.

5. Conclusion & Future Scope

In this paper, a smart parking system has been introduced, which combines sensing with IoT-enabled sensing, real-time communication, and image improvement with deep learning technology to develop an effective, scalable, and precise system of parking slot detection. The system eliminates the difference between a laboratory prototype and a production system through the integration of adaptive illumination correction and lightweight model optimization. Experiment tests, both based on benchmarks (PKLot, CNRPark-EXT) and field tests, showed high precision (more than 95%), low latency (less than 2.5 s), and high scalability to 500 slots. The findings verify that the suggested architecture can be trusted in a wide range of environmental settings, not to mention it is also scalable to large-scale implementation in smart cities.

Future Scope

Future research of the suggested smart parking system will aim at improving edge-first deployment plans to decrease cloud reliance, feeding privacy-guaranteeing learning methods, e.g. federated learning, to ensure safe cooperation of multiple operators, as well as increasing energy efficiency by using lightweight AI models within the spirit of green IoT. Also, increasing the datasets and considering more variants of lighting, weather, and urban conditions will enhance the model generalization, and introducing predictive analytics, dynamic pricing, and support of intelligent navigation can enhance the system user experience and urban traffic management further and make the system a more comprehensive and sustainable solution to NG smart cities.

6. References

[1] A. O. Elfaki, F. A. Hashim, and M. H. Alshammari, “A Smart Real-Time Parking Control and Monitoring System,” Sensors, vol. 23, no. 3, pp. 1–18, 2023.

[2] Y.-C. Yu, “Smart Parking System Based on Edge-Cloud-Dew Computing Architecture,” Electronics, vol. 12, no. 9, pp. 1–20, 2023.

[3] X. Huang, P. Li, R. Yu, M. Pan, and Z. Han, “FedParking: A Federated Learning Based Parking Space Estimation with Parked Vehicle Assisted Edge Computing,” IEEE Trans. Veh. Technol., vol. 70, no. 10, pp. 10545–10556, Oct. 2021.

[4] V. K. Sarker, T. N. Gia, I. B. Dhaou, and T. Westerlund, “Smart Parking System with Dynamic Pricing, Edge-Cloud Computing and LoRa,” Sensors, vol. 20, no. 24, pp. 1–21, 2020.

[5] M. Balfaqih, A. S. Khan, and M. A. Alqarni, “Design and Development of Smart Parking System Based on Cloud and Fog Platform,” Electronics, vol. 10, no. 23, pp. 1–15, 2021.

[6] Y. Yuldashev, M. Mukhiddinov, A. B. Abdusalomov, R. Nasimov, and J. Cho, “Parking Lot Occupancy Detection with Improved MobileNetV3,” Sensors, vol. 23, no. 13, pp. 1–16, 2023.

[7] Z. Yuan, Z. Wang, X. Li, L. Li, and L. Zhang, “Hierarchical Trajectory Planning for Narrow-Space Automated Parking With Deep Reinforcement Learning: A Federated Learning Scheme,” Sensors, vol. 23, no. 4, pp. 1–21, 2023.

[8] A. E. Saputra, S. Susanto, and I. P. Windasari, “Vehicle and Parking Space Detection for Smart Parking,” in Proc. Int. Conf. on Smart Computing and Communications (SCITEPRESS), 2023, pp. 233–240.

[9] S. K. Jha, R. Kumar, and P. S. Verma, “Car Parking Space Detection Using YOLOv8,” in Proc. Int. Conf. on Smart Computing (SCITEPRESS), 2023, pp. 110–117.

[10] P. R. L. de Almeida, L. S. Oliveira, A. S. Britto Jr., E. J. Silva Jr., and A. L. Koerich, “PKLot – A Robust Dataset for Parking Lot Classification,” Expert Systems with Applications, vol. 42, no. 11, pp. 4937–4949, 2015.

[11] P. R. L. de Almeida et al., “A Systematic Review on Computer Vision-Based Parking Lot Management Applied on Public Datasets,” arXiv preprint, arXiv:2202.08732, 2022.

[12] K. Kumar, “A Review of Parking Slot Types and Their Detection Using Computer Vision,” Future Internet, vol. 15, no. 3, pp. 1–25, 2023.

[13] L. Encío, M. Villalba, and R. Álvarez, “Visual Parking Occupancy Detection Using Extended Multi-Branch CNN,” Sensors, vol. 23, no. 6, pp. 1–15, 2023.

[14] J. Kim, “Resource-Efficient Design and Implementation of Real-Time Vehicle Occupancy Detection on Edge Devices,” Sensors, vol. 25, no. 2, pp. 1–20, 2025.

[15] N. Zhao, “CMCA-YOLO: Real-Time Object Detection for Surveillance Applications,” Electronics, vol. 13, no. 4, pp. 1–18, 2024.

[16] Y. Yuldashev, M. Mukhiddinov, A. B. Abdusalomov, R. Nasimov, and J. Cho, “Correction: Parking Lot Occupancy Detection with Improved MobileNetV3,” Sensors, vol. 24, no. 1, pp. 1–3, 2024.

[17] M. F. Asy’ari, I. H. Utami, and R. T. Putra, “Parking Space Availability Detection from Two-Camera Stitching and YOLOv5,” IAES Int. J. Artificial Intelligence, vol. 12, no. 2, pp. 745–754, 2023.

[18] C. Biyik, “A Systematic Review: Smart Parking Systems,” Sustainability, vol. 13, no. 12, pp. 1–23, 2021.

[19] X. Huang, P. Li, R. Yu, M. Pan, and Z. Han, “FedParking: A Federated Learning Based Parking Space Estimation with Parked Vehicle Assisted Edge Computing,” IEEE Trans. Veh. Technol., vol. 70, no. 10, pp. 10545–10556, 2021.

[20] J. Kim, “Lightweight SSD-MobileNetv2 for Edge-Based Parking Occupancy Detection,” Sensors, vol. 24, no. 3, pp. 1–15, 2024.