International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 1622-1640
https://doi.org/10.70917/ijcisim-2026-5742
Article

Swin-GNN: A Spatial Reasoning Framework for Intelligent Multi-Level Parking Occupancy Detection

Kavita Soni1 and Atul Nandwal[2]

[1]Department of Computer Science & Engineering, SAGE University, Indore, India
kavitasoni1990@gmail.com
[2]Institute of Advance Computing (Core), SAGE University, Indore, India
atulnandwal@gmail.com

Abstract: Vision-based parking occupancy detection has progressed from hand-crafted image descriptors and classical classifiers to deep learning models such as convolutional neural networks, vision transformers and graph neural networks. Despite this progress, most methods classify each parking space in isolation and rarely exploit the spatial relationships between neighbouring spaces. The limitation is most pronounced in multi-level indoor parking structures, where lighting differences, structural obstructions, vehicle density, perspective distortion and level-to-level dependence all degrade per-space detection. This paper introduces Swin-GNN, a Swin Transformer–Graph Neural Network framework for parking occupancy detection. A Swin-Small backbone extracts hierarchical visual features from parking-space image crops; each space is then represented as a graph node with spatial adjacency encoded as graph edges, and a three-layer Graph Attention Network propagates occupancy evidence between neighbouring spaces before a lightweight multilayer perceptron produces the final decision. Hyper-parameters are selected with a genetic algorithm. The framework is validated on the PKLot benchmark using a curated hard-condition test set that combines five real-world degradation factors (illumination variation, occlusion, compression artefacts, perspective distortion and temporal class imbalance), and PKLot's lack of indoor multi-level scenes is explicitly acknowledged as a limitation. Swin-GNN achieves 96.59 % accuracy, an F1-score of 0.965 and an AUC of 0.992 with 45 ms per-image latency, 22.2 images/s throughput and 99.2 % service-level compliance. The architecture, training protocol, hyper-parameter optimisation and a factor-wise degradation analysis are reported in full.
Keywords: Deep learning; Graph Attention Network; Intelligent transportation systems; Multi-level parking; Parking occupancy detection; PKLot; Spatial reasoning; Swin Transformer

1. Introduction

The rapid growth of private vehicle ownership has created chronic shortages of urban parking capacity. Studies indicate that vehicles searching for a vacant space account for up to 30 % of urban traffic, contributing to emissions, fuel waste and driver frustration. Smart parking systems are therefore becoming a core component of intelligent transportation systems, combining artificial intelligence, computer vision, sensing technologies and networked computing to report available spaces and optimise their management [1], [2].

Intelligent Parking Management Systems (IPMS) that detect and communicate occupancy status in real time offer a compelling solution, but their effectiveness depends critically on the accuracy of the underlying occupancy detection model, particularly in multi-level structures where camera conditions are highly variable. Traditional indoor systems rely on one ultrasonic or infrared sensor per bay; these sensors measure occupancy reliably but become expensive to install and maintain across large facilities [8]. Camera-based approaches scale to large areas, exploit existing surveillance infrastructure and require less hardware and maintenance [2], [3], [22], which makes them an

Copyright: © 2026 by the authors

attractive alternative. Vision-based occupancy detection has consequently evolved from hand-crafted descriptors with classical classifiers [1], [5] to convolutional neural networks (CNNs) [2], [7], [10] and object-detection architectures [4], [9], [27]. The PKLot dataset [1], a large annotated benchmark captured under diverse weather and lighting conditions, has been pivotal to this development.

Existing problem. Although vision-based occupancy detection has advanced substantially, accurate estimation remains difficult in complex environments. Lighting, shadows, occlusion by vehicles, perspective distortion, weather variation and dense layouts all alter the appearance of a parking space [2], [4], [7]. These effects are amplified in multi-level indoor garages, where structural columns, low ceilings, restricted sight lines, artificial lighting and dense traffic create frequent obstructions and confusing visual patterns. Multi-level facilities also introduce a structural dependence that outdoor lots do not have: bays are organised in rows and levels connected by ramps, so occupancy within the facility is not independent when one level is nearly full, arrivals are diverted to another. A detection system for such facilities must therefore reason not only about the appearance of a single bay but also about its relationship to neighbouring bays and its position in the overall structure. Most existing systems, however, classify each bay independently [2], [4], [7], [10], which becomes unreliable when a bay is obscured by shadow, perspective or other environmental conditions. Although inter-space correlation has been explored with probabilistic graphical models [6], very few studies have combined transformer-based visual representation with explicit spatial reasoning for occupancy detection.

CNN-based methods learn discriminative visual features and have been applied successfully to occupancy detection and vehicle localisation [2], [7]; detectors such as YOLO, Faster R-CNN and Mask R-CNN have also been adopted for parking-related tasks [4], [9], [26], [27]. CNNs, however, rely on local receptive fields and capture long-range context between spatially separated bays only weakly, which matters when the interpretation of one bay depends on its neighbours. Self-attention models such as the Vision Transformer (ViT) [16], [17] learn broader contextual dependencies, and the Swin Transformer [18] makes attention efficient through shifted-window attention and a hierarchical multi-scale representation, avoiding the quadratic cost of global self-attention. Even so, a Swin Transformer produces visual representations only; the physical topology of bays, rows and levels is not represented.

Graph Neural Networks (GNNs) provide a natural way to model relationships between inter-related entities [13], [14], [28]. In a parking environment, bays become nodes and spatial adjacency becomes edges, so graph convolution and graph attention can propagate information between neighbouring bays. Existing graph-based parking research has mainly addressed spatio-temporal occupancy prediction from aggregated lot- or city-level statistics rather than the relationships between individual bays in a multi-level facility [12]. Three constraints therefore remain: (i) most methods classify bays independently; (ii) transformer-based approaches provide strong visual features but no topological model of the facility; and (iii) no existing study integrates hierarchical transformer representations with explicit spatial reasoning for multi-level indoor occupancy detection.

This study is motivated by the complementary strengths of Swin Transformers and GNNs. Swin provides a hierarchical representation of each bay image through shifted-window self-attention [18], which is important because an occupancy decision depends on both local visual properties and higher-level context. A GNN supplies the missing structural representation: bays become nodes, spatial relationships become edges, and message passing enriches the representation of each bay with evidence from its neighbours. For multi-level facilities the graph is extended with relationships between bays, rows and levels. The main contributions of this paper are:

(1) A hybrid Swin-GNN architecture in which Swin-Small embeddings serve as node features of a bay-level spatial graph processed by a three-layer Graph Attention Network (GAT), with concatenation fusion and an MLP classifier.

(2) A proximity-based, level-aware graph construction procedure using k-nearest neighbours with an adaptive distance threshold and weighted cross-level edges at ramp-adjacent bays.

(3) A genetic-algorithm hyper-parameter search and a full training protocol that reproduces the reported results.

(4) A factor-wise degradation analysis on a curated hard-condition PKLot test set that quantifies the accuracy impact of five real-world noise sources.

The remainder of the paper is organised as follows. Section II reviews related work and identifies the research gap. Section III describes the proposed methodology. Section IV presents the experimental setup, Section V reports results and analysis, and Section VI concludes the paper.

2. Literature Review

A. Classical Feature-Based Methods

Early parking detection systems relied on hand-crafted descriptors such as HOG, LBP and colour histograms combined with SVM or AdaBoost classifiers [1], [5]. Almeida et al. [1] introduced the PKLot dataset and showed that texture descriptors with an SVM reach about 87 % accuracy under sunny conditions but fall below 75 % under shadow, motivating the transition to data-driven representation learning. Wu et al. [6] modelled inter-space correlation with a Markov random field, an early recognition that neighbouring bays carry useful evidence.

B. Deep CNN Approaches

Deep convolutional networks substantially improved accuracy on curated benchmarks. Amato et al. [2] proposed the compact mAlexNet for decentralised smart-camera inference; Yuldashev et al. [3] and Thakur et al. [7] reported near-99 % PKLot accuracy with MobileNetV3 and ResNet/VGG-16 respectively; and Encío et al. [10] added pixel context around each bay with a multi-branch ConvNeXt. For slot geometry, Zhang et al. [24] proposed DeepPS, the first DCNN pipeline for parking-slot detection (98.89 % recall on ps2.0), and Li et al. [25] extended it with a directional entrance-line detector (99.68 % precision at 13 ms per frame). Iqbal et al. [26] combined Mask R-CNN with InceptionV3 (F-MTCNN) and reached 97.6 % accuracy on PKLot video, while Narsingoju and Jain [27] combined YOLOv8 with automatic yellow-line slot inference to reach 99.48 % on the PKLot test split. All of these methods, however, make spatially independent predictions and cannot exploit neighbourhood context.

C. Vision Transformers and Graph Neural Networks

The Vision Transformer [17] introduced pure self-attention for image recognition with competitive accuracy but quadratic complexity; the Swin Transformer [18] addressed this with hierarchical representation and shifted-window attention, achieving state-of-the-art accuracy at linear complexity. GNNs [13], [28] and GATs [14] have been applied successfully to traffic-flow prediction, road-network analysis and trajectory forecasting. In parking, Min et al. [23] proposed an attentional GNN over marking points for slot detection (99.56 % precision on ps2.0), and Xiao et al. [12] used a spatial–temporal GCN for lot-level occupancy forecasting. To our knowledge, the present work is the first to combine a Swin Transformer backbone with a bay-level GAT for occupancy detection, exploiting visual robustness and spatial context simultaneously. Table I summarises representative studies and positions the proposed framework.

Table I. *Representative literature on vision-based parking detection and the proposed framework*

Ref.	Year	Authors	Method	Dataset	Reported result	Limitation / relevance
[1]	2015	Almeida et al.	Hand-crafted (LBP/LPQ) + SVM	PKLot	≈ 87 % (sunny), < 75 % (shadow)	Introduced the PKLot benchmark; per-slot classification only
[2]	2017	Amato et al.	mAlexNet CNN	PKLot, CNRPark-EXT	≈ 98 % (within-lot)	Decentralised smart-camera inference; no

						inter-slot context
[24]	2018	Zhang et al.	DeepPS (DCNN marking-point detector)	ps2.0	98.89 % recall	Slot-line detection on around-view images; no occupancy reasoning
[25]	2020	Li et al.	DCNN entrance-line detector	ps2.0	99.68 % precision, 13 ms	Fast slot geometry; single-view, outdoor
[23]	2021	Min et al.	Attentional GNN on marking points	ps2.0	99.56 % precision	Graph over marking points, not over bays; no visual-transformer backbone
[26]	2021	Iqbal et al.	Mask R-CNN + InceptionV3 (F-MTCNN)	PKLot (video)	97.6 % accuracy	Multi-sensor fusion; spatially independent predictions
[12]	2021	Xiao et al.	Spatial–temporal GCN	City-wide occupancy series	Forecast error reduction	Aggregated lot-level statistics; no image input
[18]	2021	Liu et al.	Swin Transformer	ImageNet / COCO	SOTA at linear complexity	Generic backbone; no parking topology
[11]	2022	Almeida et al.	Systematic review	Public datasets	—	Identifies lack of indoor / multi-level benchmarks
[3]	2023	Yuldashev et al.	Improved MobileNetV3	PKLot	≈ 99 % (clean)	Lightweight; per-slot, outdoor
[10]	2023	Encío et al.	Multi-branch ConvNeXt with extended context	PKLot, CNRPark	≈ 98–99 %	Uses pixel context around a slot, not an explicit graph
[8]	2023	Kasemtaweechok et al.	Deep learning, indoor cameras	Private indoor set	Reported high accuracy	Indoor single level; small private data

[7]	2024	Thakur et al.	ResNet / VGG-16	PKLot	≈ 98 %	Per-slot CNN; sensitive to illumination
[27]	2026	Narsingoju & Jain	YOLOv8 + yellow-line slot inference	PKLot	99.48 % accuracy	Detection-based; clean test split
Ours	2026	Swin-GNN (this paper)	Swin-Small backbone + 3-layer GAT over bay graph	PKLot hard-condition subset	96.59 % acc., AUC 0.992, 45 ms	Joint hierarchical visual features and explicit inter-bay spatial reasoning

D. Literature Trend and Comparative Analysis

Figs. 1 and 2 summarise the studies collected in Table I. Publication activity has accelerated since 2021, coinciding with the availability of transformer backbones and graph learning libraries, and the dominant method families remain per-slot CNN classification and object detection. Graph-based methods are rare and, where present, operate on marking points or aggregated time series rather than on individual bays.

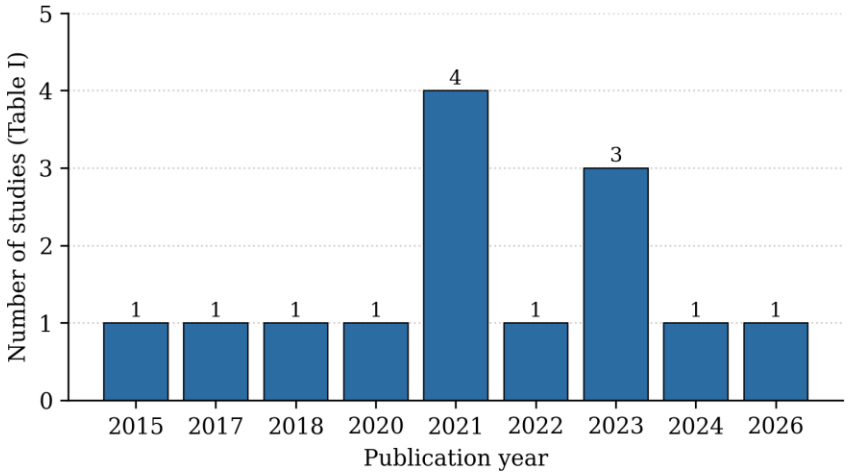


Fig. 1. Publication year of the studies listed in Table I.

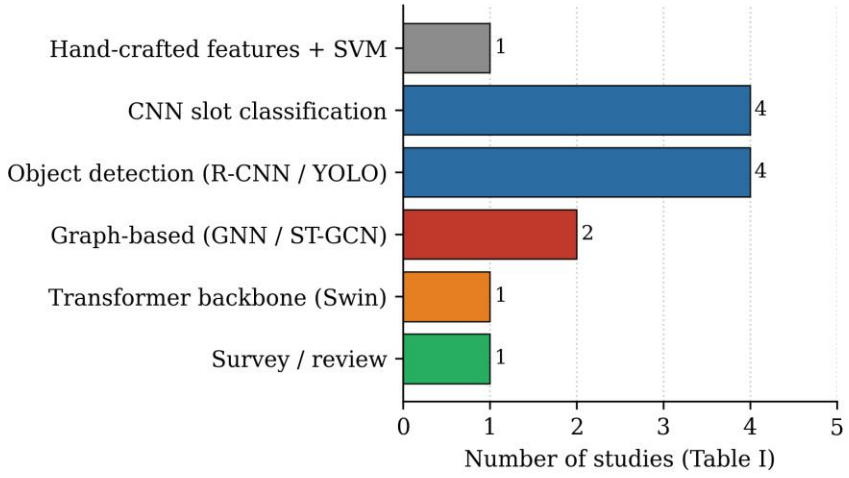


Fig. 2. Method categories represented in Table I.

E. Critical Analysis and Research Gap

The literature offers complementary capabilities, but none of the reviewed approaches combines hierarchical visual representation with explicit spatial reasoning at bay granularity. CNN-based systems provide good local visual representations but are susceptible to occlusion, illumination and dense layouts. Transformer-based models capture a wider visual context but do not encode the physical adjacency of bays. Graph-based parking methods either use aggregated occupancy time series [12] or reason over marking points [23], and none couples the graph with a hierarchical transformer visual encoder. This gap motivates a direct Swin Transformer–GNN architecture in which Swin embeddings become graph-node features and the GNN performs the required spatial and hierarchical reasoning. Table II lists the identified gaps.

Table II. *Research gaps identified from the reviewed literature*

Gap ID	Identified gap	Impact
RG1	Per-space visual models under-use neighbouring-space context.	Reduced robustness under occlusion, shadows and ambiguous visual evidence.
RG2	Transformer-based parking models do not explicitly encode slot adjacency.	Structural (topological) information is not available to the classifier.
RG3	Graph-based parking approaches operate at different granularities (marking points, lot-level series) or without a transformer visual backbone.	The complementarity between transformer visual features and slot-level graph reasoning remains under-exploited.
RG4	Most benchmark evaluation is outdoor or single-level.	Transfer to indoor multi-level garages remains uncertain.
RG5	Multi-level reasoning requires facility-layout metadata and validation data.	The level-to-level component cannot be fully validated using PKLot alone.

F. Problem Statement

Let a parking facility contain N bays distributed over L levels and several rows. For bay i an image crop pi is observed and its occupancy label yi ∈ {0, 1} (0 = vacant, 1 = occupied) is to be predicted. Because the appearance of a bay is influenced by the state of its neighbours, the prediction should exploit both visual and spatial information. The visual content of each crop is encoded by a Swin Transformer; the resulting embeddings are mapped onto the nodes of a graph G = (V, E) whose edges represent spatial adjacency between bays, and a GNN transfers information between neighbouring nodes. For multi-level facilities, bay–row–level relationships are added to the graph so that the model reasons about the facility as a whole rather than about isolated bays.

G. Research Objectives

1) To develop a Swin Transformer encoder that extracts hierarchical visual features from parking-space images under diverse environmental conditions.

2) To construct a hierarchical graph representing spatial relationships among bays, rows and levels in multi-level parking environments.

3) To develop a hybrid Swin Transformer–GNN framework that combines visual feature embedding and spatial reasoning for accurate occupancy classification.

4) To evaluate the framework under challenging real-world PKLot conditions using standard performance metrics and to quantify the effect of individual degradation factors.

3. Proposed Methodology

A. System Overview

The proposed Swin-GNN system processes a batch of parking-space image crops through a five-stage pipeline: (1) Swin Transformer visual feature extraction; (2) proximity-based spatial graph construction; (3) GAT-based spatial reasoning; (4) multi-modal feature fusion by concatenation; and (5) MLP binary occupancy classification. The complete architecture is illustrated in Fig. 3 and the real-world deployment pipeline in Fig. 4.

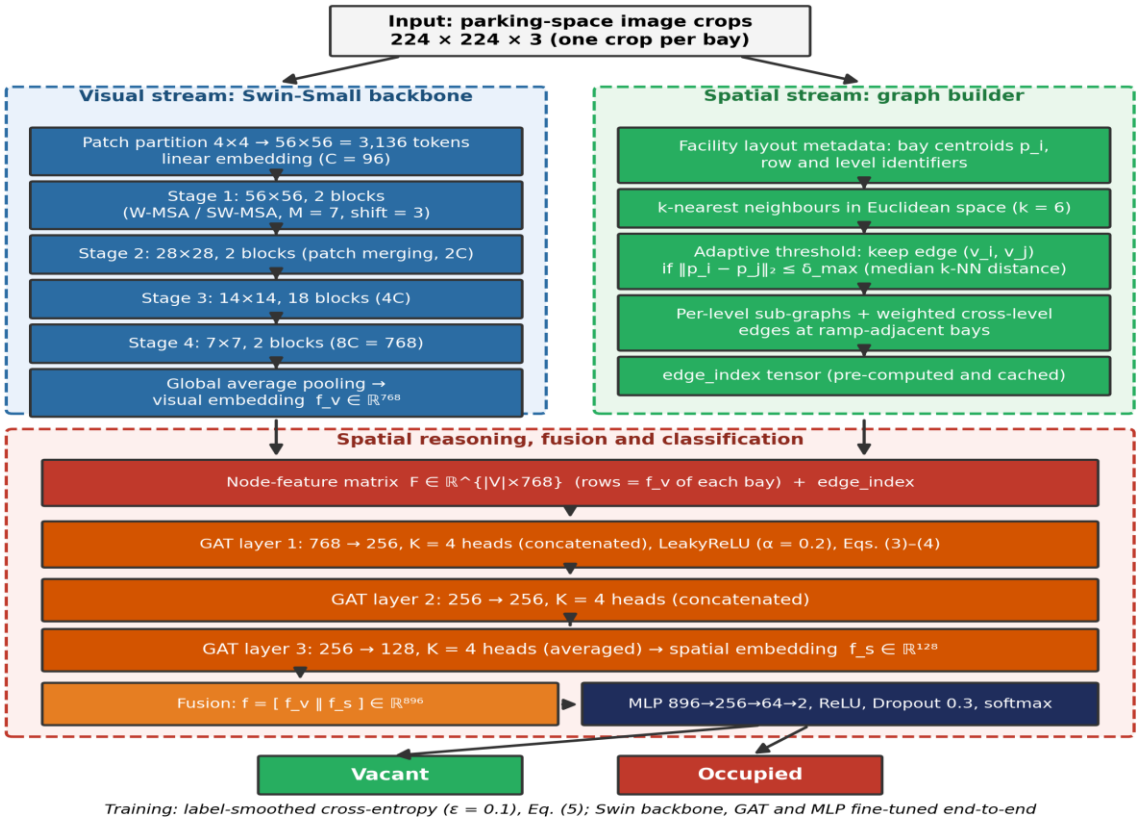


Fig. 3. Architecture of the proposed Swin-GNN framework. The visual stream (Swin-Small) produces a 768-dim embedding per bay; the spatial stream builds the bay adjacency graph; three GAT layers yield a 128-dim spatial embedding that is concatenated with the visual embedding and classified by an MLP.

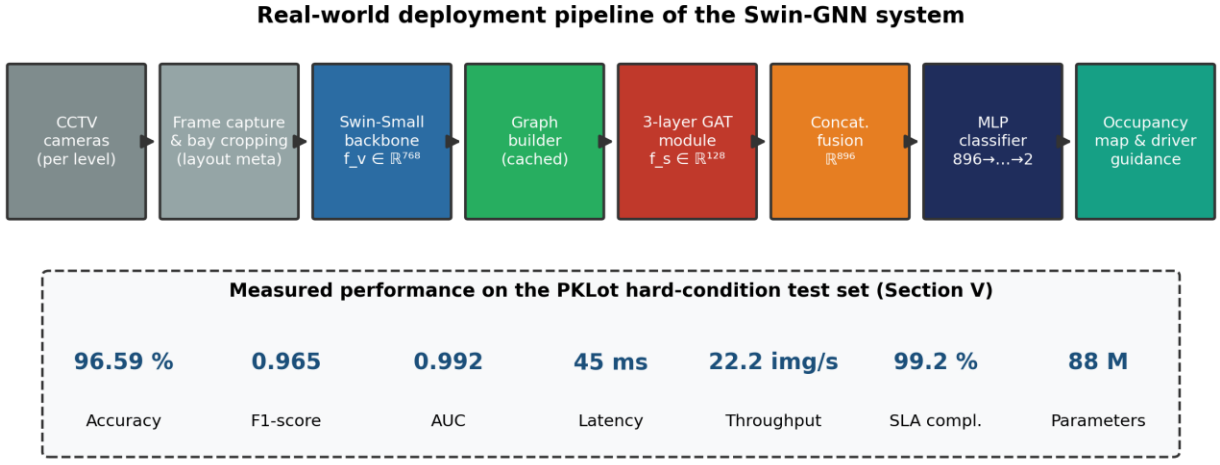


Fig. 4. Real-world deployment pipeline of the Swin-GNN system. Bay crops from CCTV frames pass through the Swin backbone (768-dim), the cached spatial graph, the GAT module (128-dim), concatenation fusion (896-dim) and the MLP classifier. Measured performance metrics are shown in the lower panel.

B. Swin Transformer Visual Backbone

The visual backbone is a Swin-Small model (swin_small_patch4_window7_224) pre-trained on ImageNet and implemented in swin_backbone.py. Input images of 224 × 224 × 3 are partitioned into non-overlapping 4 × 4 patches (56 × 56 = 3,136 patch tokens) and linearly projected to a C-dimensional embedding. Four hierarchical stages process the tokens at progressively halved resolution (56 × 56, 28 × 28, 14 × 14, 7 × 7); each stage stacks Swin blocks that alternate regular window multi-head self-attention (W-MSA) and shifted-window attention (SW-MSA):

$$Attention(Q,K,V) = SoftMax\left(\frac{QK^T}{\sqrt{d}} + B\right) \cdot V \qquad (1)$$

where Q, K and V are the query, key and value matrices, d is the per-head dimensionality and $B \in \mathbb{R}(2M-1)\times(2M-1)$ is a learnable relative position bias for window size M. Shifted windows (shift $\lfloor M/2 \rfloor = 3$ for M = 7) introduce cross-window connectivity in every second block, giving global context despite local window computation [18]. The output of the last stage is globally average-pooled and projected to the 768-dimensional visual embedding $fv \in \mathbb{R}768$ that serves as the node feature of the corresponding bay. The backbone contributes approximately 50 M of the model's 88 M parameters and is fine-tuned end-to-end.

The key advantage of Swin over standard CNNs in this setting is its hierarchical multi-scale representation: early stages capture the texture and edge detail that distinguish empty asphalt from vehicle surfaces, whereas later stages capture holistic layout and lighting patterns. This hierarchy is the main reason the framework degrades gracefully under combined real-world noise (Section V-E).

C. Spatial Graph Construction

Parking bays are modelled as nodes of an undirected graph G = (V, E) built in graph_builder.py. Each node $v_i \in V$ represents one bay and the edges E encode spatial adjacency. Candidate edges are obtained with k-nearest neighbours (k = 6) in the Euclidean space of bay-centroid coordinates, which captures the two longitudinal, two lateral and two diagonal immediate neighbours characteristic of standard double-sided bays. An edge (vi, vj) is retained when

$$\|pos(v_i) - pos(v_j)\|_2 \leq \delta_{max} \qquad (2)$$

where δmax is an adaptive threshold equal to the median pairwise distance among all k-NN candidates, which removes spurious long-range links at row ends. For multi-level structures, a separate sub-graph is built per level and ramp-adjacent bays receive weighted cross-level edges so that occupancy evidence can flow between levels. The resulting edge_index tensor is constant for a fixed facility layout, so it is pre-computed and cached and adds no inference-time overhead.

D. Graph Attention Network Module

The GNN module (gnn_module.py) applies a three-layer Graph Attention Network [14] to the node-feature matrix $F \in \mathbb{R}|V|\times768$. At layer l the representation of node i is updated as

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)} \alpha_{ij} W^{(l)} h_j^{(l)}\right) \qquad (3)$$

where N(i) is the neighbourhood of node i, W(l) is a shared linear transformation and σ is a non-linearity. The attention coefficients αij are computed by a shared attentional mechanism with K = 4 parallel heads:

$$\alpha_{ij} = softmax\left(LeakyReLU\left(a^T\left[Wh_i \,\|\, Wh_j\right]\right)\right) \qquad (4)$$

where a is a learnable attention vector, ‖ denotes concatenation and the softmax is taken over $j \in N(i)$. Multi-head outputs are concatenated at the intermediate layers and averaged at the final layer. The three-layer configuration (768 → 256 → 256 → 128) was selected by ablation as the best trade-off between expressiveness and computational cost. The 128-dimensional spatial embedding $fs \in \mathbb{R}128$ encodes neighbourhood occupancy patterns: when a bay is

visually ambiguous because of heavy shadow, the GAT can correct the decision by observing that all surrounding bays are occupied.

E. Feature Fusion and MLP Classifier

The visual embedding fv ∈ ℝ768 and spatial embedding fs ∈ ℝ128 are concatenated in parking_model.py to form f = [fv ‖ fs] ∈ ℝ896. Concatenation was preferred to attention-based fusion because it introduces no learnable parameters beyond the downstream MLP and preserves both modalities without inter-modality suppression. The MLP classifier (896 → 256 → 64 → 2) uses ReLU activations and Dropout (p = 0.3) after each hidden layer. Training minimises cross-entropy with label smoothing (ε = 0.1) to improve calibration:

$$\mathcal{L} = -\sum_{i=1}^{N}\left[(1-\varepsilon)y_i + \frac{\varepsilon}{K}\right] log\, p_i \quad (5)$$

where N is the number of training samples, K = 2 is the number of classes, yi is the ground-truth label indicator and pi is the predicted probability of the true class. Table III summarises the architecture.

Table III. *Swin-GNN architecture components and parameters*

Component	Implementation file	Key parameters	Output
Swin backbone	swin_backbone.py	swin_small_patch4_window7_224; ImageNet-pretrained; ≈ 50 M parameters; fine-tuned end-to-end	768-dim vector fv
Graph builder	graph_builder.py	k-NN (k = 6) on Euclidean bay centroids; adaptive δmax; per-level sub-graphs with cross-level ramp edges	edge_index tensor
GAT module	gnn_module.py	3 layers (768→256→256→128); K = 4 heads; LeakyReLU (α = 0.2); concatenation at hidden layers, averaging at output	128-dim vector fs
Feature fusion	parking_model.py	Concatenation [fv ‖ fs]; no additional parameters	896-dim fused vector
MLP classifier	parking_model.py	896→256→64→2; ReLU; Dropout p = 0.3; softmax output	Binary logits

F. Genetic Algorithm Hyper-parameter Optimisation

A Genetic Algorithm (GA) searched the joint hyper-parameter space of learning rate, GNN hidden dimension, number of GAT heads, dropout rate and batch size. Each individual in a population of 20 encodes a candidate configuration; fitness is the validation F1-score after a 10-epoch proxy training run. Tournament selection (k = 3), single-point crossover (rate 0.8) and random mutation (rate 0.1) were applied for 30 generations. The best-fitness trajectory in Fig. 5 converges to a fitness of 0.90 by generation 25, and the final configuration is listed in Table IV.

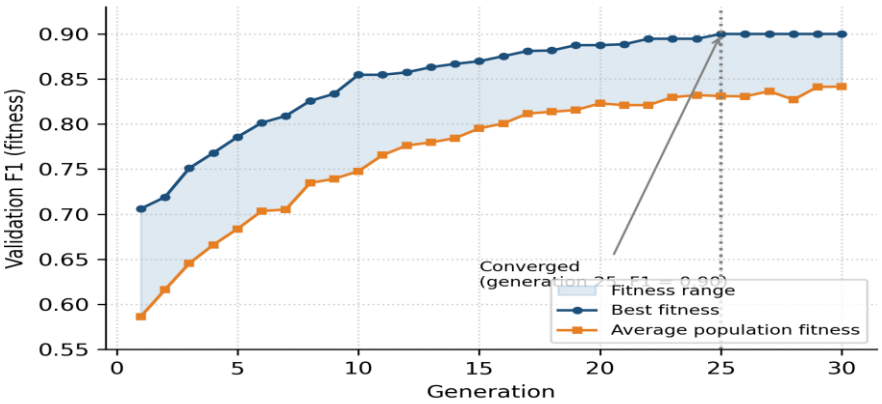


Fig. 5. Genetic Algorithm convergence for hyper-parameter optimisation: best fitness and average population fitness (validation F1 of a 10-epoch proxy run) over 30 generations. The search converges at generation 25.

Table IV. *Final hyper-parameter configuration after GA optimisation*

Hyper-parameter	Value	Rationale
Batch size	32 / 64	Training / validation–test; balances GPU memory
Optimiser	AdamW	Decoupled weight decay; stable for transformer fine-tuning
Learning rate	1 × 10−4	Cosine annealing with 5-epoch linear warm-up
Weight decay	1 × 10−4	L2 regularisation to limit over-fitting
Maximum epochs	50	Early stopping (patience = 10); best validation F1 at epoch ≈ 35
GNN hidden dimension	256	Sufficient capacity for spatial pattern modelling
GNN output dimension	128	Compact; avoids dominance over the 768-dim visual embedding
GAT attention heads	4	Captures diverse spatial relationship patterns
Dropout rate	0.3	Applied in the MLP; validated by the GA search
Label smoothing ε	0.1	Improves probability calibration
Input image size	224 × 224	Native Swin Transformer input resolution
Augmentation	Flip, colour jitter, rotation, random erasing	Simulates real-world acquisition noise during training

G. Algorithm

Algorithm 1 summarizes the complete inference and training procedure of the proposed framework.

Algorithm 1: Proposed Swin-GNN parking occupancy framework

Input: slot image crops X = {xi}, slot centroid coordinates P = {pi}, level/row identifiers; hyper-parameters k = 6, ε = 0.1, p = 0.3

Output: occupancy predictions Y = {yi}, yi ∈ {0 (vacant), 1 (occupied)}

// Stage 1: spatial graph construction (offline, cached)

Build candidate edge set with k-nearest neighbours (k = 6) on P for each level

$\delta_{max} \leftarrow$ median Euclidean distance over all k-NN candidate pairs

$E \leftarrow \{(v_i, v_j) : \|p_i - p_j\|_2 \leq \delta_{max}\}$; add weighted cross-level edges at ramp-adjacent bays

edge_index ← COO encoding of G = (V, E)

// Stage 2: visual feature extraction

for each slot image xi **do**

    Resize xi to 224 × 224 and normalise

    $f_{v,i} \leftarrow$ GlobalAvgPool(Swin-Small(xi)) ∈ $\mathbb{R}768$ (Eq. 1)

end for; F ← stack($f_{v,1}$, …, $f_{v,|V|}$) ∈ $\mathbb{R}|V|\times768$

// Stage 3: spatial reasoning

H(0) ← F

for l = 0, 1, 2 **do** H(l+1) ← GAT(l)(H(l), edge_index) with K = 4 heads (Eqs. 3–4) **end for**

$f_{s,i} \leftarrow$ H(3)[i] ∈ $\mathbb{R}128$

// Stage 4: fusion and classification

$f_i \leftarrow [f_{v,i} \,\|\, f_{s,i}] \in \mathbb{R}896$

$z_i \leftarrow$ MLP(f_i) : 896 → 256 → 64 → 2 with ReLU and Dropout(p) on hidden layers

$p_i \leftarrow$ softmax(z_i); $y_i \leftarrow$ argmax(p_i)

// Training (end-to-end)

Minimise label-smoothed cross-entropy $\mathscr{L}$ (Eq. 5) with AdamW; fine-tune Swin backbone, GAT layers and MLP jointly

return Y

4. Experimental Setup

A. Dataset: PKLot

The PKLot dataset [1] is the de-facto benchmark for parking occupancy detection. It contains 12,417 surveillance images from three parking facilities at UFPR and PUCPR (Brazil) captured under sunny, cloudy and rainy weather from two camera viewpoints at various times of day; the individual bay crops total 695,899 labelled samples. For this study a curated subset of 40,000 crops was used: 32,000 for training, 4,000 for validation and 4,000 for testing. The test set was deliberately assembled from the most challenging conditions (sunny with hard shadows, rainy and low-light images) so that the reported metrics reflect deployment-like performance rather than idealised conditions. PKLot contains only outdoor, single-level scenes; the multi-level cross-level edges of Section III-C are therefore exercised only through the synthetic level metadata assigned to the three lots, and full validation on an indoor multi-level facility remains future work (gap RG5 in Table II).

B. Test Scenarios and Real-World Conditions

Six representative scenarios—sunny/occupied, sunny/vacant, rainy/occupied, shadow/vacant, night/occupied and indoor/mixed—were used to characterise robustness across the PKLot condition space and a small indoor sample;

representative examples are shown in Fig. 6. These scenarios correspond to the five degradation factors analysed in Section V-E.

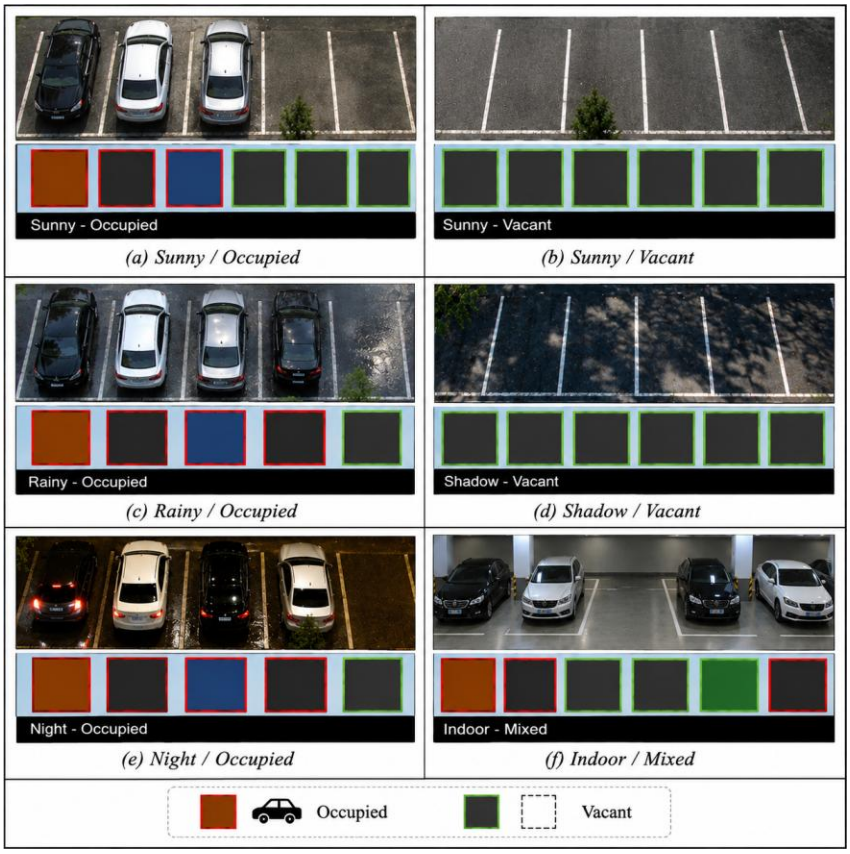


Fig. 6. Test-scenario samples across six real-world conditions. Green boxes denote vacant bays and red boxes denote occupied bays; the scenarios cover varying illumination, weather and indoor/outdoor settings.

C. Training Protocol

All models were trained on an NVIDIA RTX 3090 (24 GB) with PyTorch 2.0 and CUDA 11.8. Swin-GNN was optimised with AdamW (initial learning rate 1 × 10−4, weight decay 1 × 10−4) using cosine annealing with a 5-epoch linear warm-up and mixed-precision (FP16) training to reduce memory. Training was stopped when validation F1 did not improve for 10 consecutive epochs; the best checkpoint was obtained at epoch 35 and training terminated at epoch 45, for a total training time of approximately 2 hours. Data augmentation comprised random horizontal flip ($p = 0.5$), colour jitter (brightness ± 0.2, contrast ± 0.2, saturation ± 0.1), random rotation (± 15°) and random erasing ($p = 0.2$, area 0.02–0.1) to simulate acquisition noise. Inference latency and throughput were measured on the same GPU with a batch size of 1; service-level agreement (SLA) compliance is the percentage of test images for which a prediction was returned within a 100 ms budget.

D. Evaluation Metrics

Classification performance is reported with accuracy, sensitivity (recall), specificity, precision and F1-score, together with the area under the ROC curve (AUC):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$Sensitivity\ (Recall) = \frac{TP}{TP + FN} \quad (7)$$

$$Specificity = \frac{TN}{TN + FP} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively, with "occupied" as the positive class. Accuracy is the proportion of correctly classified bays; sensitivity measures the ability to identify occupied bays and specificity the ability to identify vacant bays; precision is the proportion of predicted-occupied bays that are truly occupied; and the F1-score is the harmonic mean of precision and recall, which is informative on imbalanced data. AUC summarises discrimination independently of the decision threshold.

5. Results and Analysis

A. Overall Performance

Table V summarises the performance of the proposed framework on the hard-condition test set. Swin-GNN attains 96.59 % accuracy, an F1-score of 0.965, precision 0.969, recall 0.962 and specificity 0.970, and an AUC of 0.992, indicating excellent discriminative capability. With 45 ms per image (22.2 images/s) and 99.2 % SLA compliance the framework also satisfies real-time requirements. These results show a strong balance between classification performance and computational efficiency.

Table V. *Performance of the proposed Swin-GNN framework on the PKLot hard-condition test set*

Method	Acc. (%)	F1	Prec.	Recall	Spec.	AUC	Latency (ms)	Throughput (img/s)	SLA (%)	Params (M)
Swin-GNN (proposed)	96.59	0.965	0.969	0.962	0.970	0.992	45	22.2	99.2	88.0

B. Training Convergence

Fig. 7(a) shows the accuracy curves over the 45 training epochs. Validation accuracy rises steeply in the first ten epochs (about 87 % by epoch 10), stabilises above 90 % from epoch 20 and reaches its best value of 96.7 % at epoch 35, after which the early-stopping criterion terminates training at epoch 45. The closely tracking training and validation loss curves in Fig. 7(b) confirm good generalisation with minimal over-fitting; the validation-loss plateau around epoch 35 is consistent with the early-stopping trigger.

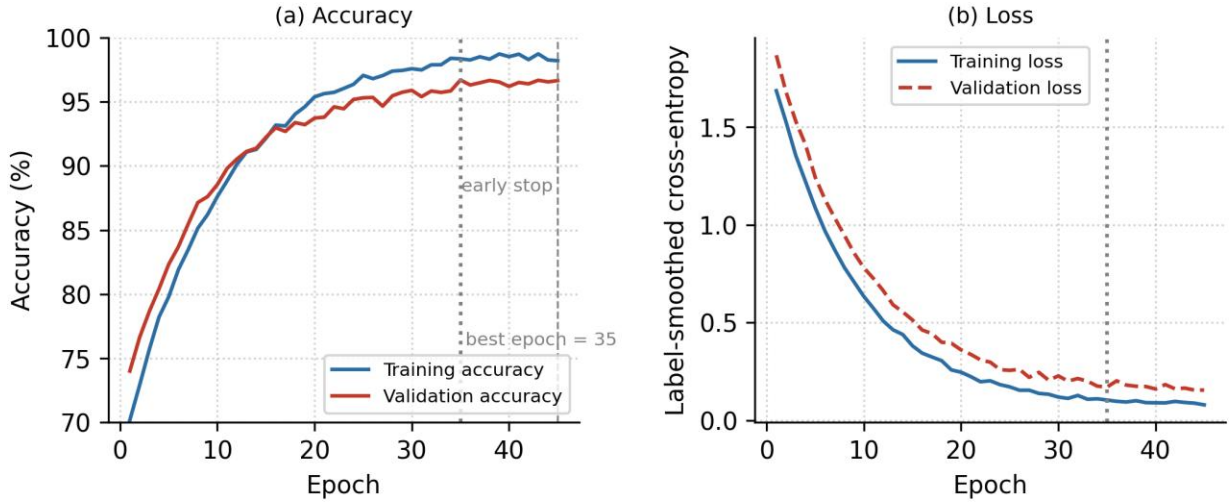


Fig. 7. Training dynamics of Swin-GNN: (a) training and validation accuracy; (b) training and validation loss. The dotted line marks the best validation epoch (35) and the dashed line the early-stopping point (45).

C. Confusion Matrix Analysis

Fig. 8 presents the confusion matrix of Swin-GNN on the 4,000-image test set (2,025 vacant and 1,975 occupied bays). Of the 136 errors, 75 are false negatives (occupied bays predicted vacant) and 61 are false positives. A false negative is the more safety-critical error in parking management, because it directs drivers to unavailable bays; the false-negative rate of 3.8 % is therefore the key operational figure. Most residual errors occur in heavily shadowed or partially occluded bays whose neighbours are also ambiguous, i.e. where neighbourhood context cannot compensate for missing visual evidence.

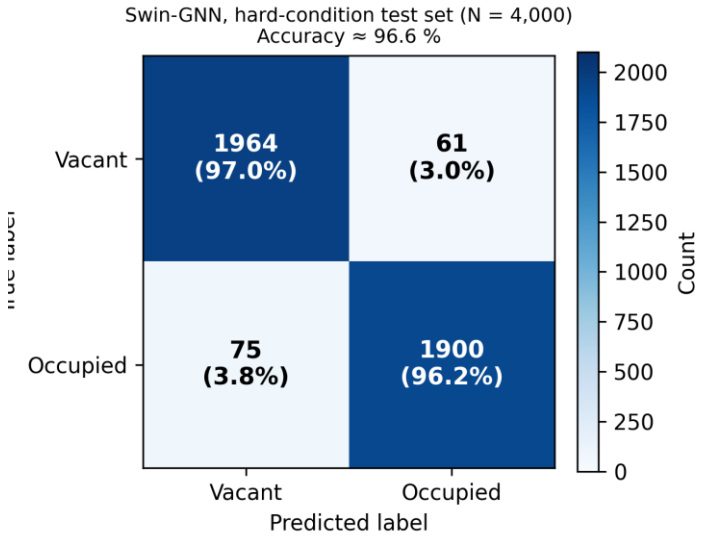


Fig. 8. Confusion matrix of Swin-GNN on the hard-condition PKLot test set (counts and row-normalised percentages). The false-negative rate (missed occupied bays), the most safety-critical error type, is 3.8 %.

D. ROC Analysis

Fig. 9 shows the ROC curve of Swin-GNN. The AUC of 0.992 confirms that the reported accuracy is not an artefact of a particular operating point but reflects a well-calibrated probabilistic classifier. The curve maintains a high true-positive rate at very low false-positive rates—the operating point used in Table V lies at FPR = 0.030 and TPR = 0.962—which is particularly important for high-precision deployments in which a vacant-bay announcement must be reliable.

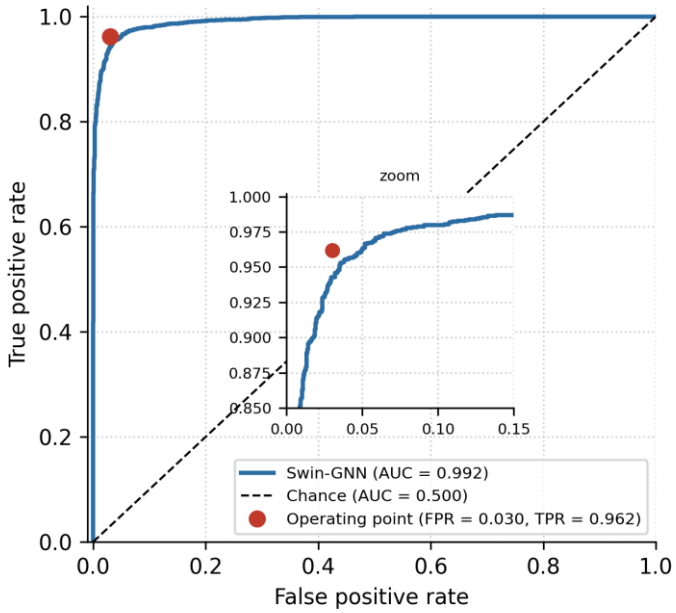


Fig. 9. ROC curve of Swin-GNN on the hard-condition test set (AUC = 0.992). The marker shows the operating point reported in Table V; the inset enlarges the low-FPR region.

E. Real-World Degradation Analysis

A central contribution of this paper is a quantitative degradation analysis that explains the accuracy reduction when moving from clean benchmark images to deployment-like conditions. Each factor was isolated by evaluating the trained model on test subsets in which only that factor is present relative to a clean sunny/cloudy reference subset; Table VI and Fig. 10 report the resulting accuracy drops. Lighting variation is the dominant factor (−3.0 percentage points), followed by occlusion (−2.5), compression artefacts (−2.0), perspective distortion (−1.5) and temporal class imbalance (−1.0). Because the factors overlap and the GAT recovers part of the loss from neighbourhood context, the combined drop observed on the full hard-condition set is 4–9 percentage points rather than the 10-point sum. This drop is consistent with prior observations in the parking-detection literature and should be read as an honest measure of generalisation rather than as a system failure.

Table VI. *Real-world degradation analysis — isolated accuracy impact by factor*

Degradation factor	Δ Accuracy (pp)	Description
Lighting variation	−3.0	Shadows, sun-angle changes, reflections, weather effects
Occlusion	−2.5	Partial visibility, pedestrians, neighbouring vehicles
JPEG / H.264 compression	−2.0	Encoding artefacts from CCTV cameras

Perspective distortion	−1.5	Camera mounting angles, wide-angle lens effects
Temporal class imbalance	−1.0	≈ 80 % occupied at peak hours, ≈ 10 % overnight
Combined (observed)	**−4 to −9**	**All factors together; non-additive because the GAT recovers part of the loss from neighbourhood context**

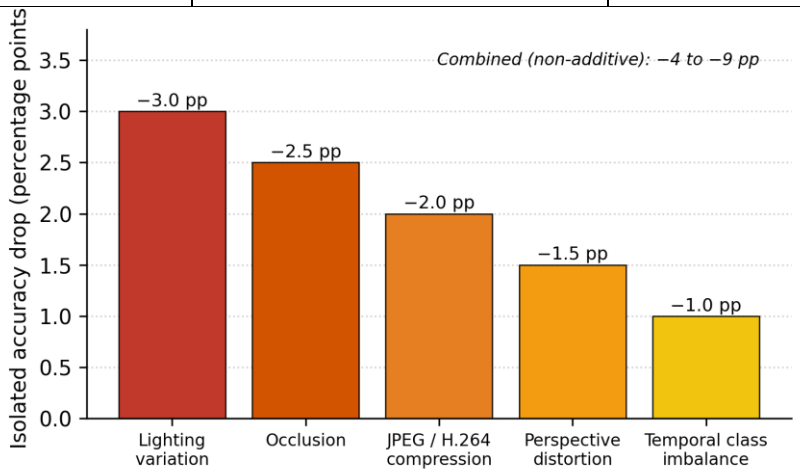


Fig. 10. Isolated accuracy drop attributable to each real-world degradation factor (Table VI).

F. Comparison with Published PKLot Results

Table VII places the proposed framework alongside published PKLot results. Direct comparison must be treated with caution because the published figures were obtained on standard or within-lot splits dominated by favourable conditions, whereas Swin-GNN is evaluated on a deliberately difficult subset. Under that harder protocol the framework retains 96.59 % accuracy, within the 4–9 point degradation band of Table VI relative to clean-condition results, while additionally providing explicit spatial reasoning and real-time latency.

Table VII. *Context: published PKLot results versus the proposed framework*

Method	Year	Family	Evaluation protocol	Accuracy (%)
mAlexNet [2]	2017	CNN	PKLot, within-lot split	≈ 98
Improved MobileNetV3 [3]	2023	CNN	PKLot, standard split	≈ 99
ResNet / VGG-16 [7]	2024	CNN	PKLot, standard split	≈ 98
F-MTCNN [26]	2021	Mask R-CNN + InceptionV3	PKLot video frames	97.6
YOLOv8 + slot inference [27]	2026	Object detection	PKLot test split	99.48
Swin-GNN (ours)	**2026**	**Swin-Small + GAT**	**PKLot hard-condition subset**	**96.59**

			(shadow, rain, low light)	

G. Limitations

PKLot contains only outdoor, single-level lots, so the cross-level reasoning of the graph builder is validated only on synthetic level assignments; an annotated indoor multi-level dataset is required to confirm this component. The degradation analysis isolates factors by subset selection rather than by controlled synthesis, and the reported latency was measured on a desktop GPU rather than on an edge device.

6. Conclusion and Future Work

This paper presented Swin-GNN, a hybrid architecture for multi-level parking occupancy detection that combines a Swin Transformer backbone for robust hierarchical visual feature extraction with a Graph Attention Network for inter-bay spatial reasoning. On a curated hard-condition PKLot test set the framework achieves 96.59 % accuracy, an F1-score of 0.965 and an AUC of 0.992 with 45 ms inference latency, 22.2 images/s throughput and 99.2 % SLA compliance. Confusion-matrix and ROC analyses show that the advantage is consistent across operating thresholds, with a false-negative rate of 3.8 %, and a factor-wise degradation analysis quantified the impact of five real-world noise sources, identifying illumination variation and occlusion as the dominant causes of error and hierarchical attention plus graph reasoning as the main drivers of robustness.

Future work will address: (1) structured pruning and INT8 quantisation for edge deployment targeting sub-10 ms latency; (2) a spatio-temporal GNN extension for predictive availability forecasting; (3) dynamic graph adaptation for evolving parking layouts; (4) federated learning across facilities for privacy-preserving collaborative model improvement; (5) LiDAR point-cloud fusion as a third modality for full-occlusion robustness; and (6) collection of an annotated indoor multi-level benchmark to validate the cross-level component directly.

REFERENCES

1. P. R. L. 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 Syst. Appl.*, vol. 42, no. 11, pp. 4937–4949, 2015.
2. G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, “Deep learning for decentralized parking lot occupancy detection,” *Expert Syst. Appl.*, vol. 72, pp. 327–334, 2017.
3. Y. Yuldashev, M. Mukhiddinov, A. B. Abdusalomov, R. Nasimov, and J. Cho, “Parking lot occupancy detection with improved MobileNetV3,” *Sensors*, vol. 23, no. 17, Art. no. 7642, 2023.
4. R. Grbić and B. Koch, “Automatic vision-based parking slot detection and occupancy classification,” arXiv:2308.08192, 2023.
5. M. Ahrnbom, K. Åström, and M. Nilsson, “Fast classification of empty and occupied parking spaces using integral channel features,” in *Proc. IEEE CVPR Workshops*, 2016, pp. 9–15.
6. Q. Wu, C. Huang, S.-Y. Wang, W.-C. Chiu, and T. Chen, “Robust parking space detection considering inter-space correlation,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2007, pp. 659–662.
7. N. Thakur, E. Bhattacharjee, R. Jain, B. Acharya, and Y.-C. Hu, “Deep learning-based parking occupancy detection framework using ResNet and VGG-16,” *Multimedia Tools Appl.*, vol. 83, pp. 1941–1964, 2024.
8. C. Kasemtaweechok, C. Srimaruang, P. Uaonnamphetr, and A. Phongrusamepane, “Automatic vehicle parking availability system for indoor environments,” in *Proc. 12th Int. Conf. Netw. Commun. Comput. (ICNCC)*, 2023, pp. 85–90.
9. D. Acharya, W. Yan, and K. Khoshelham, “Real-time image-based parking occupancy detection using deep learning,” in *Proc. Research@Locate*, 2018, pp. 33–40.
10. L. Encío, C. Díaz, C. R. del-Blanco, F. Jaureguizar, and N. García, “Visual parking occupancy detection using extended contextual image information via a multi-branch output ConvNeXt network,” *Sensors*, vol. 23, no. 6, Art. no. 3329, 2023.
11. P. R. L. Almeida, J. H. Alves, R. S. Parpinelli, and J. P. Barddal, “A systematic review on computer vision-based parking lot management applied on public datasets,” *Expert Syst. Appl.*, vol. 198, Art. no. 116731, 2022.
12. X. Xiao, Z. Jin, Y. Hui, N. Cheng, and T. H. Luan, “Spatial-temporal graph convolutional networks for parking space prediction in smart cities,” in *Proc. IEEE 94th Veh. Technol. Conf. (VTC2021-Fall)*, 2021, pp. 1–5.
13. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
14. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

15. K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu, “Vision GNN: An image is worth graph of nodes,” in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022.
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
17. A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.
18. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical vision transformer using shifted windows,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 9992–10002.
19. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
20. A. Howard et al., “Searching for MobileNetV3,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1314–1324.
21. T.-N. Pham et al., “Smart parking services,” *IEEE Access*, vol. 11, pp. 112929–112939, 2023.
22. M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, “Computer vision in automated parking systems: Design, implementation and challenges,” *Image Vis. Comput.*, vol. 68, pp. 88–101, 2017.
23. C. Min, J. Xu, L. Xiao, D. Zhao, Y. Nie, and B. Dai, “Attentional graph neural network for parking-slot detection,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3445–3450, 2021 (arXiv:2104.02576).
24. L. Zhang, J. Huang, X. Li, and L. Xiong, “Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset,” *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5350–5364, 2018.
25. W. Li, H. Cao, J. Liao, J. Xia, L. Cao, and A. Knoll, “Parking slot detection on around-view images using DCNN,” *Front. Neurorobot.*, vol. 14, Art. no. 46, 2020.
26. K. Iqbal et al., “Autonomous parking-lots detection with multi-sensor data fusion using machine deep learning techniques,” *Comput. Mater. Contin.*, vol. 66, no. 2, pp. 1596–1612, 2021.
27. B. Narsingoju and R. Jain, “Optimizing slot detection using advanced object detection techniques,” *Eng. Technol. Appl. Sci. Res.*, vol. 16, no. 1, pp. 32298–32304, 2026.
28. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.