

ICT Research

JOURNAL ON INFORMATION TECHNOLOGIES & COMMUNICATIONS

ISSN 1859-3534

BỘ KHOA HỌC VÀ CÔNG NGHỆ - MINISTRY OF SCIENCE AND TECHNOLOGY

Research and Development on Information & Communication Technology

Volume 28
Number 1
March 2026

Các công trình
nghiên cứu, phát triển
và ứng dụng công nghệ
thông tin và truyền thông

Established in 1999, the *Journal on Information Technologies & Communications* is a peer-reviewed scholarly journal published by the *Vietnam Journal of Science and Technology - MOST*, under the Ministry of Science and Technology of Vietnam.

The Journal publishes original research articles and review articles covering a broad spectrum of topics across information technology, communications, electronics, and automation. It welcomes submissions in, but is not limited to, the following areas:

- **Computer Science**
- **Computer Engineering**
- **Information Systems**
- **Software Engineering**
- **Information Security**
- **Electronics**
- **Communications**
- **Computer Networks**
- **Automatic Control**

CONTENTS

An effective statistical model to evaluate the two-phase query algorithm in the encrypted database with a secure index structure. **3**

Ninh Duong-Bao, Huy Phuoc Dang, Luong Nguyen Thi, Khanh Nguyen-Huu

Integrating explainable AI for transparent and accurate cryptocurrency prediction. **13**

Anh Hoang, Minh Phuc Huynh

Enhancing hate speech detection with knowledge-based prompting for large language models. **26**

Thanh Le, Tuong Le, Thien Khai Tran

Enhancing parking space recognition from surveillance cameras. **37**

Dung Nguyen, Van-Dung Hoang, Van-Tuong-Lan Le

Probing the hidden memory: Detecting training data inclusion in language models. **48**

Giang H.H. Nguyen, Long H.B. Nguyen

Exploring rebalancing strategies in forecast-informed portfolio optimisation with deep learning techniques on VN100 stocks. **57**

Minh Duc Nguyen

Enhancing parking space recognition from surveillance cameras

Dung Nguyen¹, Van-Dung Hoang^{2*}, Van-Tuong-Lan Le³

¹University of Sciences, Hue University, 77 Nguyen Hue Street, Phu Nhuan Ward, Hue City, Vietnam

²Ho Chi Minh City University of Technology and Engineering, 1 Vo Van Ngan Street, Thu Duc Ward, Ho Chi Minh City, Vietnam

³Hue University, 3 Le Loi Street, Vinh Ninh Ward, Hue City, Vietnam

Received 12 August 2025; revised 11 October 2025; accepted 27 November 2025

Abstract:

The development of accurate parking spot detection systems is essential for intelligent transportation infrastructure. Conventional approaches, such as those employing convolutional neural networks, often face challenges in addressing complex conditions, including variable lighting, suboptimal camera perspectives, and limited capacity to model global spatial contexts. To address these limitations, this study introduces a novel Transformer-based architecture that utilises the self-attention mechanism to effectively capture long-range dependencies and comprehensive spatial relationships. The proposed model integrates data augmentation, fine-tuning, query optimisation, and feature extraction to enhance detection accuracy. Evaluation on a real-world parking dataset shows that the model achieves a mean average precision of 98.1%, which exceeds the results of conventional methods based on convolutional neural networks.

Keywords: computer vision, convolutional neural networks, parking spot detection, surveillance cameras, transformer.

1. Introduction

As cities grow and traffic volume increases, the demand for efficient parking management systems becomes more pressing. Reliable detection and allocation of parking spaces play a key role in improving traffic flow and reducing emissions from idling or circulating vehicles. As a result, the development of automated, accurate, and robust parking detection systems has emerged as a central priority in the field of intelligent transportation research.

Significant progress has been made in parking spot detection, evolving from traditional image processing methods to advanced deep learning techniques. Convolutional Neural Networks (CNNs) [1] have been widely adopted due to their ability to autonomously extract features from image data, yielding notable success in various detection tasks. Nevertheless, CNN-

based approaches often struggle with challenges such as fluctuating lighting conditions, diverse camera perspectives, and complex data distributions. In recent years, Transformer-based architectures [2], originally developed for natural language processing, have been adapted for computer vision tasks and have shown effectiveness in modelling long-range spatial relationships. Models such as DETR (DEtection TRansformer) [3] and its variants [4-7] have shown superior performance in object detection, suggesting their potential for parking spot detection, where understanding spatial relationships is critical.

This study proposes a Transformer-based approach tailored for parking spot detection, designed to achieve high accuracy across diverse real-world scenarios. A complete detection pipeline has been developed, incorporating advanced data augmentation and fine-tuning strategies to optimise performance for parking

*Corresponding author: Email: dunghv@hcmute.edu.vn

management applications. The proposed method is evaluated using real-world parking datasets, with performance assessed through metrics such as mean Average Precision (mAP), Intersection over Union (IoU), and inference time. The findings indicate that Transformer-based models effectively address the limitations of traditional approaches, offering a practical foundation for the integration of such technologies into intelligent transportation systems to support the development of adaptive and reliable parking solutions.

2. Related work

2.1. Object detection

Object detection, a cornerstone of computer vision, has undergone significant transformation with the development of machine learning methodologies. From early rule-based techniques to modern deep learning models and, more recently, Transformer-based architectures, each approach has introduced specific improvements to the field.

Early object detection techniques relied heavily on manually designed features and rule-based algorithms. Methods such as the Histogram of Oriented Gradients (HOG) [8] and the Scale-Invariant Feature Transform (SIFT) [9], often paired with classifiers like Support Vector Machines (SVMs), were prevalent. These approaches offered interpretable and computationally straightforward solutions. However, their performance was limited by challenges in accommodating variations in object appearance, scale, and perspective, particularly in complex or dynamic environments.

The emergence of deep learning marked a paradigm shift in object detection, with CNN becoming the foundation of modern detection frameworks. Notable architectures, including Faster R-CNN [10-12], YOLO (You Only Look Once) [13-24], and SSD (Single Shot MultiBox Detector) [25], have significantly enhanced detection accuracy and efficiency. Innovations such as region proposal networks in Faster R-CNN and end-to-end detection pipelines in YOLO and SSD have enabled real-time performance and robust handling of diverse object detection tasks. Recent refinements, such as enhancements to YOLOv5 for improved detection of small and occluded objects [18], underscore the adaptability of CNN-based approaches. Nevertheless, these methods primarily emphasise local feature

extraction, which can constrain their ability to model global contextual relationships in intricate visual scenes.

Transformer-based architectures have recently gained prominence as a compelling alternative to CNN, owing to their capacity to capture long-range dependencies and global contextual relationships in visual data. A seminal contribution in this domain is DETR [3], which redefines object detection as a direct set prediction task, eliminating the need for complex region proposal networks. This streamlined approach achieves robust accuracy across diverse visual scenarios. Subsequent advancements, such as Deformable DETR [4], Swin Transformer [26, 27], and models designed for enhanced spatial structure comprehension [28], have further refined Transformer-based methods by improving computational efficiency and adaptability to varying image resolutions. Additionally, specialised models like Gogglesdet [29], which employs clustering techniques for targeted object detection, demonstrate the adaptability of Transformer architectures in addressing specific detection challenges.

Beyond general object detection, Transformer-based models have proven effective in perception tasks for autonomous systems and complex detection scenarios. For example, H. Lin, et al. (2025) [30] developed AEFusion, an attention-based ensemble learning framework for Bird's Eye View (BEV) fusion in autonomous modular buses. By using self-attention mechanisms, their approach enhances perception accuracy through efficient integration of multi-sensor data. Similarly, C. Hao, et al. (2025) [31] introduced a Vision Transformer-based network tailored for camouflaged object detection (COD) and salient object detection (SOD). Their work demonstrates the strength of self-attention in modelling global spatial dependencies, enabling superior detection of objects in challenging, visually intricate environments. These studies highlight the adaptability of Transformer-based architectures across diverse detection tasks, positioning them as a robust alternative to traditional CNN-based approaches.

2.2. Parking spot detection

The detection of parking spaces has become an important area of research in the development of intelligent transportation systems, driven by the need to enhance urban mobility and efficiency. Recent

advancements have shifted from traditional image processing techniques to sophisticated machine learning approaches, including CNN- and Transformer-based architectures. This section reviews key studies in this domain, highlighting their contributions and limitations.

CNNs have been widely adopted for parking spot detection due to their robust ability to extract local image features. For instance, L. Zhang, et al. (2018) [32] utilised deep convolutional neural networks (DCNNs) with a large benchmark dataset, achieving notable improvements in detection accuracy. Similarly, H. Do, et al. (2020) [33] enhanced CNN performance by incorporating contextual information, enabling better handling of complex environments. Additionally, W. Li, et al. (2020) [34] applied CNNs to panoramic images, attaining high precision but at the expense of increased computational demands. Despite these advancements, CNN-based methods often rely heavily on local feature extraction, rendering them susceptible to challenges such as varying lighting conditions and dynamic parking layouts.

Transformer-based architectures have emerged as a powerful solution, excelling in capturing global spatial relationships critical for complex parking scenarios. J.K. Suhr, et al. (2021) [35] proposed a one-stage model that integrates global and local information, achieving high accuracy with reduced inference times. Similarly, Y. Wang, et al. (2022) [36] introduced DetPS, a hybrid

model that incorporates global spatial learning into a convolutional framework, enhancing generalisability for real-world applications. However, the computational complexity of Transformer models often poses challenges for real-time deployment. Recent studies, such as those on Bird's Eye View (BEV) perception for autonomous vehicles [30] and camouflaged object detection [31], further demonstrate the versatility of attention mechanisms in spatial reasoning.

Unlike AEFusion [30], which focuses on multi-sensor fusion for autonomous buses, our approach is tailored specifically for parking spot detection using surveillance imagery, emphasising structured spatial relationships in urban settings. In contrast to the camouflaged object detection method in [31], which prioritises subtle visual cues, our model addresses the structured layout of parking areas, where spatial alignment and occlusion handling are paramount. These distinctions demonstrate the unique aspects of our Transformer-based approach in tackling the unique challenges of parking spot detection.

3. Methods

3.1. Overview of the proposed method

The proposed model builds upon the DETR architecture and is designed to enhance parking space detection under diverse real-world conditions. It combines a ResNet backbone pretrained on ImageNet [37] with a Transformer structure consisting of six

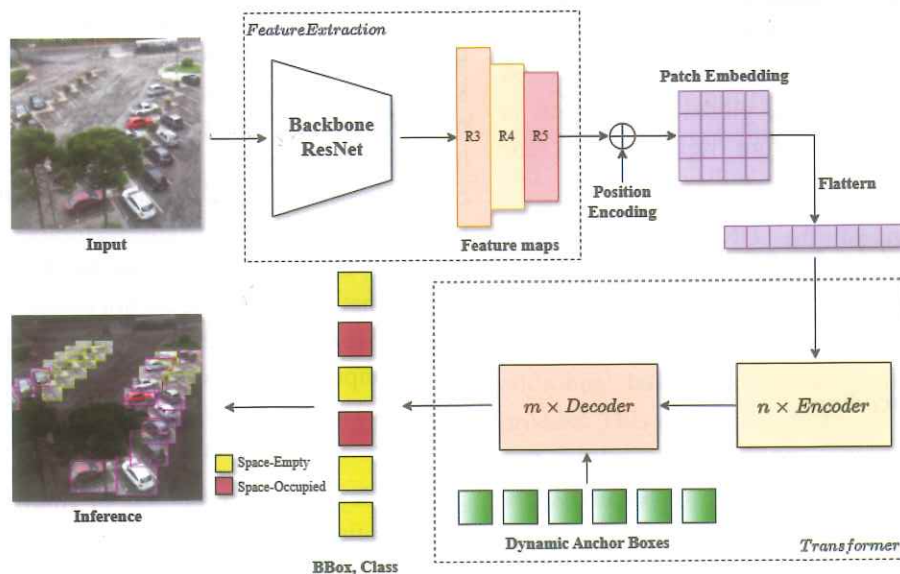


Fig. 1. The proposed model.

encoder-decoder layers. The backbone is responsible for extracting spatial representations from the input image, while the Transformer encoders refine these features to capture global contextual relationships. The decoder then generates object-level predictions through iterative interaction with the encoded features. Finally, prediction heads output class labels and bounding box coordinates, enabling accurate localisation and classification of parking spaces. Model optimisation is carried out using a combination of classification and regression losses, with parameters updated through the AdamW optimizer. Fig. 1 provides an overview of the architecture.

3.2. Feature extraction

In the proposed model, the feature extraction process is responsible for generating spatial features from the input image, which are then used for subsequent object detection tasks. This process consists of two main components:

Backbone: A CNN is used as the backbone, specifically the ResNet architecture pretrained on the ImageNet dataset. The spatial features from the input image are extracted by the backbone, capturing important details such as textures, edges, and shapes. Features from different layers of the backbone provide varying levels of abstraction, which are crucial for detecting objects at different scales and complexities.

Transformer encoders: The extracted features are refined and enhanced by Transformer encoders. These encoders use the self-attention mechanism to model long-range dependencies and contextual relationships within the image. As a result, global information is captured, improving the feature representation for downstream tasks.

The combination of the CNN backbone and Transformer encoders ensures that the extracted features are both spatially rich and contextually aware, forming a strong input for the decoder and prediction heads.

3.3. Positional encoding

In the proposed model, positional encodings are used to incorporate the positional information of objects within the image. Since the Transformer architecture is permutation-invariant, it does not inherently capture the spatial relationships between objects. To overcome this,

positional encodings are added to the input features to provide the model with information about the relative or absolute positions of objects in the image.

In this model, sinusoidal position encoding (SinCos) is used. These embeddings are based on sine and cosine functions, allowing the model to capture the relative position of each object in the input image. The positional encodings are computed as follows (see Eq. (1)):

$$PE(pos, i) = \begin{cases} \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) & \text{if } i \text{ is even,} \\ \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) & \text{if } i \text{ is odd.} \end{cases} \quad (1)$$

where pos is the position of the token in the sequence, i is the dimension of the position encoding, d is the dimensionality of the embedding.

These sine and cosine functions allow the model to represent both local and global spatial information, and the embeddings are added directly to the input features before being passed into the Transformer encoder. This helps the model learn the content of both the objects and their locations, improving its ability to detect and localise objects in the image.

3.4. Transformer

The decoder in the proposed model plays a central role in object detection by transforming input queries into predictions for object bounding boxes and labels. It operates on the refined features generated by the Transformer encoder and leverages two types of queries to guide the detection process. These are dual queries, consisting of positional and content queries. Positional queries represent predefined spatial locations, also known as anchor boxes, which act as guides for potential object locations in the image. On the other hand, content queries are learnable embeddings that capture object patterns and are responsible for probing features corresponding to specific object characteristics. Together, these dual queries enable the model to locate objects that align with the anchor boxes while exhibiting patterns similar to those captured by the content queries. As the input data moves through the decoder layers, these queries are progressively updated to better align with ground-truth objects.

The decoder is composed of multiple layers, and its operation involves layer-by-layer refinement. At each layer, the dual queries interact with the refined feature maps generated by the encoder using attention

mechanisms. This interaction allows for the integration and further refinement of positional and content information. The outputs from these interactions are iteratively adjusted, enabling the predicted bounding boxes and labels to narrow down progressively and align more closely with the actual objects in the image. This step-by-step refinement process ensures that the model captures both the spatial and semantic details necessary for accurate object detection.

After processing through all the decoder layers, the final layer outputs predictions for bounding boxes and object labels. These predictions are passed to specialised prediction heads, which perform classification and box regression tasks. To compute the final loss, the predictions are matched with ground-truth objects using a bipartite graph matching algorithm. The iterative refinement of queries and the decoder's ability to fuse spatial and content information make it particularly effective in detecting objects with varying scales and complexities, achieving accurate detection across a range of real-world scenarios.

3.5. Loss function

The loss function in the model is designed to optimise two primary tasks: classification and bounding box regression. It combines three loss components: *classification loss*, *L1 loss*, and *Generalised IoU (GloU) loss*, each contributing to a specific aspect of object detection performance. The final loss is computed as shown in Eq. (2):

$$\text{Loss} = \lambda_{\text{cls}} \cdot \mathcal{L}_{\text{cls}} + \lambda_{\text{L1}} \cdot \mathcal{L}_{\text{L1}} + \lambda_{\text{GloU}} \cdot \mathcal{L}_{\text{GloU}} \quad (2)$$

Classification loss: The classification loss is computed using Focal loss, which addresses the imbalance between positive and negative samples by applying adjustable weights. It is defined as in Eq. (3):

$$\mathcal{L}_{\text{cls}} = -\alpha \cdot (1 - p_t)^\gamma \cdot \log(p_t) \quad (3)$$

- p_t : The predicted probability for the correct class.
- $\alpha = 0.25$: Balances the weight between positive and negative samples.
- $\gamma = 2$: Reduces the weight for well-classified samples to focus on hard examples.

This component ensures accurate prediction of object classes.

L1 Loss: The L1 loss calculates the absolute difference between the predicted and ground-truth bounding box coordinates (x, y, w, h) as described in Eq. (4):

$$\mathcal{L}_{\text{L1}} = \|b - b^*\|_1 \quad (4)$$

- b : Predicted bounding box.
- b^* : Ground-truth bounding box.

L1 loss ensures the precise localization of the bounding boxes.

Generalized IoU Loss. The GloU loss [38] extends IoU by penalising both overlap mismatches and redundant areas in the predicted bounding boxes, as shown in Equation 5:

$$\mathcal{L}_{\text{GloU}} = 1 - \text{IoU} + \frac{|C \setminus (b \cup b^*)|}{|C|} \quad (5)$$

- C : The smallest enclosing box containing both b and b^* .
- $|C \setminus (b \cup b^*)|$: The area outside the union of the predicted and ground-truth boxes.

This component enhances the alignment of predicted boxes with the ground-truth, even when there is no overlap.

4. Experiment and findings

4.1. Dataset

The PKLot dataset [39], developed by the Federal University of Paraná, Brazil, is a crucial resource for parking spot detection research. It contains over 12,000 images captured from three different parking lots (PUC, UFPR04, and UFPR05), with more than 700,000 labelled parking spaces categorised as Space-Empty or Space-Occupied. These images, stored in JPEG format with a resolution of 1280x720, closely resemble real-world parking scenarios, encompassing both outdoor environments (PUC, UFPR05) and an indoor setting (UFPR04).

PKLot stands out for its diversity, covering a wide range of lighting conditions (e.g., sunny, rainy, overcast) and viewpoints, making it well-suited for training machine learning models. Cameras are positioned at varying heights, further enhancing the dataset's generalisation capability across different environments.

To improve detection performance, we introduce an additional background class during the experimental process. This class is automatically assigned to regions that do not contain parking spaces, ensuring that the model can differentiate between relevant and irrelevant areas. By incorporating the background class, the model can better handle ambiguous detections, reducing false positives and improving overall reliability (Fig. 2).



Fig. 2. An image from the training set.

However, PKLot also presents challenges. Occlusions caused by trees, poles, or other vehicles complicate detection, while the dynamic nature of parking lots adds variability. Additionally, image noise, including shadows, reflections, and variations in camera quality, can impact model accuracy. Another limitation is the dataset's uneven distribution, with significantly fewer samples in UFPR04 compared to PUC, which may affect model performance.

4.2. Data augmentation

Data augmentation is an essential step in improving the robustness and generalisation of object detection models, particularly in scenarios involving limited training data or diverse environmental conditions. In this study, a diverse set of data augmentation techniques was applied to simulate various conditions and improve model training. The pipeline includes a combination of geometric transformations, resizing, cropping, and normalisation, as described below:

- Horizontal flip: Applied horizontal flipping with a probability of 50%, simulating mirrored environments.
- Random transformation selection: Randomly selected between two paths: Path 1: Resized the image to predefined scales, Path 2: Applied a sequence of transformations: resizing, cropping, and resizing again.
- Normalisation: Normalised pixel values based on the dataset's mean and standard deviation.

4.3. Training parameters

Table 1. Experimental parameters and environment.

Parameter	Value
Epochs	50
Optimizer	AdamW
Weight decay	0.0001
Learning rate	0.0001
Loss class	Focal loss
Focal loss	$\alpha=0.25, \gamma=2$
Loss BBox	L1
Numbers of encoder	6
Numbers of decoder	6
Heads of attention	8
Object queries	300
Attention of encoder	Deformable attention
Attention of decoder	Cross attention
Backbone	ResNet
Backbone-Pretrained	ImageNet
Model-Pretrained	COCO

The model is trained using the AdamW optimizer, along with a learning rate scheduler to effectively adjust the learning rate during training. To enhance generalisation and mitigate overfitting, various data augmentation techniques are utilised. These augmentations help the model perform robustly on unseen data by simulating diverse real-world conditions. A detailed summary of the experimental parameters and configurations is provided in Table 1, offering a comprehensive overview of the training setup.

4.4. Evaluation metrics

In the parking spot detection task, the performance of the model is evaluated using several standard metrics, such as mean average precision (mAP) and Intersection over Union (IoU). These metrics not only help compare the accuracy of models but also provide insights into the complexity and computational efficiency of the methods.

a) Mean average precision (mAP)

mAP is the primary metric for evaluating the accuracy of object detection models. It is calculated as follows:

1) Precision (PREC) and Recall (REC):

$$\text{PREC} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \text{REC} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

where:

- TP (True Positive): Correct predictions.
- FP (False Positive): Incorrect predictions.
- FN (False Negative): Missed detections.

2) Mean average precision (mAP): With N classes in the detection task, the mAP is computed as:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^N \text{AP}_i \quad (7)$$

where AP_i is the Average Precision of class i .

To further analyse detection performance across different object scales, the mAP can be divided into:

- mAP_s: Evaluates the detection performance for small objects (typically with areas less than 32×32 pixels).
- mAP_m: Assesses the performance for medium-sized objects (with areas between 32×32 and 96×96 pixels).
- mAP_l: Measures the precision for large objects (with areas greater than 96×96 pixels).

These metrics provide insight into how well the model performs across varying object sizes, which is crucial for applications where object scale significantly impacts detection accuracy.

b) Intersection over Union (IoU):

IoU measures the overlap between the predicted bounding box and the ground truth bounding box. The formula is:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \quad (8)$$

where:

- Area of Intersection: The area of overlap between the two boxes.
- Area of Union: The total area of both boxes minus the intersection.

IoU is typically used to decide if a prediction is correct (True Positive) if it exceeds a predefined threshold, such as 0.5 or 0.75.

4.5. Experimental results

To evaluate the effectiveness of the proposed model, the experimental process follows a two-phase

approach: Pre-training and Fine-tuning. In the first phase (Pre-training), the model is trained on a large-scale dataset to learn general feature representations. In the second phase (Fine-tuning), the pre-trained model is further optimised on a more specific dataset to adapt to the target domain. Similarly, we conducted the same procedure on several previous models to provide a fair comparison with the proposed model. Below are the details of these phases:

a) Training on the COCO dataset

In the initial phase, the model is trained on the COCO dataset [40], a widely used dataset containing 80 object classes with over 200,000 annotated images. This dataset is divided into a training set (train2017) and a validation set (val2017). These experiments are conducted using the MMDetection library [41]. The results achieved on the COCO dataset are presented in Table 2.

Table 2. Experimental results on several models using the COCO dataset.

Model	Epoch	Param	GFlop	mAP
SSD300 [25]	120	25M	341G	25.5
RetinaNet [42]	25	36M	128G	36.5
FasterRCNN [12]	12	41M	134G	37.4
DETR [3]	150	41M	60.5G	39.9
DETR-PKLot	50	44M	65G	42.3

From Table 2, it can be observed that the proposed model DETR-PKLot achieves the highest mAP (42.3). Compared to DETR, DETR-PKLot maintains a similar parameter count (44M vs 41M) while slightly increasing computational cost (65GFlops vs 60.5GFlops). However, the number of training epochs significantly reduces from 150 to 50, showcasing improved training efficiency and higher accuracy.

b) Training on the PKLot Dataset

To thoroughly evaluate the efficacy of the proposed DETR-PKLot model for parking slot detection, we conducted a comprehensive assessment on the PKLot dataset, with results analysed through both quantitative metrics and qualitative visualisations. A transfer learning approach was employed, where models were

first pre-trained on the COCO dataset and subsequently fine-tuned on the PKLot dataset for 50 epochs, as described in the Dataset section. The performance of DETR-PKLot and baseline models is summarised in Table 3, which reports metrics including mean Average Precision (mAP), mAP at IoU thresholds of 0.5 and 0.75 (mAP 0.5, mAP 0.75), and mAP for small and medium objects (mAPs, mAPm). The proposed DETR-PKLot model achieves an outstanding mAP of 98.1%, with mAP0.5 and mAP0.75 scores of 98.1%, and mAPs and mAPm scores of 98.9% and 98.3%, respectively. These results markedly outperform other models, including RetinaNet (mAP 50.5%), Faster R-CNN (mAP 59.7%), DETR (mAP 93.7%), and SSD300 (mAP 97.9%). Moreover, DETR-PKLot maintains a moderate parameter count (44M) and computational complexity (65 GFLOPs) compared to SSD300's significantly higher 341 GFLOPs, underscoring its balance between accuracy and efficiency for real-world applications.

Table 3. Experimental results on several models using transfer learning on the PKLot dataset.

Model	Param	GFlop	mAP0.5	mAP	mAPs	mAPm
RetinaNet	36M	128G	85.9	50.5	49.2	55.2
FasterRCNN	41M	134G	93.1	59.7	58.4	63.9
DETR	41M	60.5G	98.0	93.7	93.2	95.7
SSD300	25M	341G	98.0	97.9	98.5	96.3
DETR-PKLot	44M	65G	98.1	98.1	98.9	98.3

Complementing the quantitative results, visualisations provide further insight into the model's performance. The confusion matrix (Fig. 3) demonstrates high classification accuracy for space-empty (99.56%) and space-occupied (99.78%) labels, with only minor misclassifications: 0.27% of space-empty samples misclassified as space-occupied and 0.17% as background, and 0.14% of space-occupied samples misclassified as space-empty and 0.09% as background. Additionally, predicted images from the validation set (Fig. 4) illustrate the model's ability to accurately detect and classify parking spaces under diverse conditions. These visualisations validate the model's high accuracy and reinforce its practical applicability in parking management systems. Comparative analysis highlights the weaknesses of other models: RetinaNet struggles with small and medium-sized objects (mAPs

49.2%, mAPm 55.2%), while DETR shows notable improvements (mAP 93.7%). DETR-PKLot, however, achieves superior performance across all metrics, particularly in mAPs (98.9%) and mAPm (98.3%), enabling robust detection of both small and medium objects in complex urban environments with challenges such as occlusion and lighting variations.

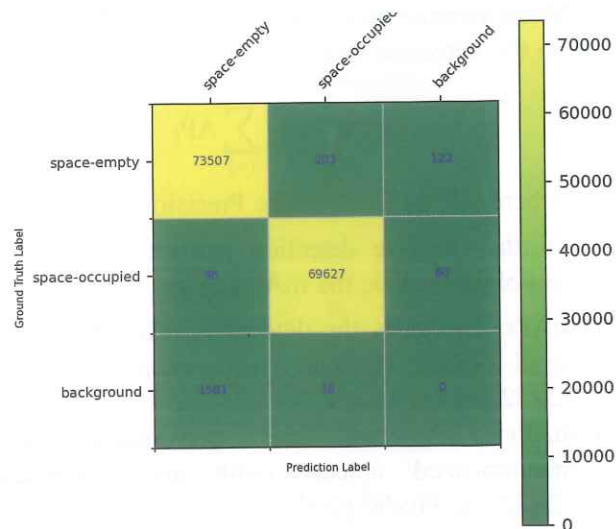


Fig. 3. Confusion matrix.

In the experimental setup, the model was deployed on a system equipped with an NVIDIA GeForce RTX 4090 GPU (24GB VRAM), using PyTorch 1.10.0 and CUDA 11.1. The batch size was set to 1 to directly measure inference time per image. Results indicate that the average processing time per image was approximately 16.6 ms, of which 0.7 ms was attributed to data loading and preprocessing, while 15.9 ms corresponded to the actual model inference. This translates to a throughput of about 63 frames per second (FPS), which is sufficient for real-time deployment scenarios.

To further verify the model's generalisation ability, we tested it on the CNRPark+EXT dataset [43, 44], which differs in environment, viewpoints, and lighting conditions. The model maintained high performance, achieving an mAP of 93.4%, demonstrating that it not only learns robust features from PKLot but also adapts well to different parking lot scenarios (Fig. 4). This reinforces the stability and effectiveness of the proposed approach, while confirming that the diversity of environmental conditions within PKLot contributes significantly to enhancing the model's generalisation



Fig. 4. Some predicted images from the validation set.

capability. In future work, we aim to further optimise computational demands and integrate multi-source data to improve robustness under diverse real-world conditions.

5. Conclusions

This study introduces a novel parking slot detection framework that capitalises on recent advancements in deep learning, with a particular focus on Transformer-based architectures. Based on an analysis of the strengths and limitations of existing detection methods, this work emphasises the importance of robust datasets, such as PKLot, in the development of effective algorithms. The proposed approach addresses key challenges, including occlusion, varying lighting conditions, and data imbalance, to achieve an mAP of 98.1%, exceeding the performance of existing state-of-the-art methods. Designed for enhanced accuracy and generalisability across diverse real-world scenarios, the framework demonstrates accuracy and efficiency for practical applications. Future research will explore strategies to

further improve stability under varying environmental conditions and investigate the integration of multi-source data to enhance detection robustness and reliability.

CRediT author statement

Dung Nguyen: Methodology, Formal analysis, Original draft preparation, Visualisation; Van-Dung Hoang: Supervision, Validation, Writing - Reviewing and Editing; Van-Tuong-Lan Le: Conceptualisation, Data curation, Investigation.

ACKNOWLEDGEMENTS

This research is funded by University of Sciences, Hue University under the grant number DHKH2026A-013. The authors sincerely thank Hue University for supporting this research.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] Y. LeCun, L. Bottou, Y. Bengio, et al. (1998), "Gradient-based learning applied to document recognition", *Proceedings of The IEEE*, **86**(11), pp.2278-2324, DOI: 10.1109/5.726791.
- [2] A. Vaswani, N. Shazeer, N. Parmar, et al. (2017), "Attention is all you need", *Advances in Neural Information Processing Systems*, **30**, pp.6000-6010, DOI: 10.48550/arXiv.1706.03762.
- [3] N. Carion, F. Massa, G. Synnaeve, et al. (2020), "End-to-end object detection with transformers", *European Conference on Computer Vision*, pp.213-229, DOI: 10.48550/arXiv.2005.12872.
- [4] X. Zhu, W. Su, L. Lu, et al. (2020), "Deformable DETR: Deformable transformers for end-to-end object detection", *ICLR 2021*, DOI: 10.48550/arXiv.2010.04159.
- [5] S. Liu, F. Li, H. Zhang, et al. (2022), "DAB-DETR: Dynamic anchor boxes are better queries for DETR", *arXiv*, DOI: 10.48550/arXiv.2201.12329.
- [6] D. Meng, X. Chen, Z. Fan, et al. (2021), "Conditional DETR for fast training convergence", *Proceedings of The IEEE/CVF International Conference on Computer Vision*, pp.3651-3660.
- [7] Y. Zhao, W. Lv, S. Xu, et al. (2024), "DETRs beat YOLOs on real-time object detection", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.16965-16974, DOI: 10.48550/arXiv.2304.08069.
- [8] N. Dalal, B. Triggs (2005), "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, **1**, pp.886-893.
- [9] T. Lindeberg (2012), "Scale invariant feature transform", *Scholarpedia*, **7**, DOI: 10.4249/scholarpedia.10491.
- [10] R. Girshick, J. Donahue, T. Darrell, et al. (2014), "Rich feature hierarchies for accurate object detection and semantic segmentation", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.580-587, DOI: 10.48550/arXiv.1311.2524.
- [11] R. Girshick (2015), "Fast R-CNN", *IEEE International Conference on Computer Vision (ICCV)*, pp.1440-1448, DOI: 10.1109/ICCV.2015.169.
- [12] S. Ren, K. He, R. Girshick, et al. (2017), "Faster R-CNN: Towards real-time object detection with region proposal networks", *IEEE Transactions on Pattern Analysis Machine Intelligence*, **39**(6), pp.1137-1149.
- [13] J. Redmon, S. Divvala, R. Girshick, et al. (2016), "You only look once: Unified, real-time object detection", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.779-788.
- [14] J. Redmon, A. Farhadi (2017), "YOLO9000: Better, faster, stronger", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.6517-6525, DOI: 10.48550/arXiv.1612.08242.
- [15] J. Redmon (2018), "YOLOv3: An incremental improvement", *arXiv*, DOI: 10.48550/arXiv.1804.02767.
- [16] A. Bochkovskiy, C.Y. Wang, H.Y.M. Liao (2020), "YOLOv4: Optimal speed and accuracy of object detection", <https://arxiv.org/pdf/2004.10934>, accessed 5 July 2025.
- [17] G. Jocher (2020), "Ultralytics YOLOv5", <https://github.com/ultralytics/yolov5>, accessed 10 May 2025.
- [18] K. Zhu, H. Lyu, Y. Qin (2025), "Enhanced detection of small and occluded road vehicle targets using improved YOLOv5", *Signal, Image and Video Processing*, **19**(1), pp.1-15, DOI: 10.1007/s11760-024-03756-3.
- [19] C. Li, L. Li, Y. Geng, et al. (2023), "YOLOv6 v3.0: A full-scale reloading", *arXiv*, DOI: 10.48550/arXiv.2301.05586.
- [20] C.Y. Wang, A. Bochkovskiy, H.Y.M. Liao (2023), "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.7464-7475, DOI: 10.48550/arXiv.2207.02696.
- [21] G. Jocher, A. Chaurasia, J. Qiu (2023), "Ultralytics YOLOv8", <https://github.com/ultralytics/ultralytics>, accessed 10 May 2025.
- [22] C.-Y. Wang, I.H. Yeh, H.Y.M. Liao (2024), "YOLOv9: Learning what you want to learn using programmable gradient information", *Computer Vision - ECCV 2024, Lecture Notes in Computer Science*, Springer, pp.1-21, DOI: 10.1007/978-3-031-72751-1_1.
- [23] A. Wang, H. Chen, L. Liu, et al. (2024), "YOLOv10: Real-time end-to-end object detection", *Advances in Neural Information Processing Systems*, **37**, pp.107984-108011, DOI: 10.48550/arXiv.2405.14458.
- [24] G. Jocher, J. Qiu (2024), "Ultralytics YOLO11", <https://github.com/ultralytics/ultralytics>, accessed 10 May 2025.
- [25] W. Liu, D. Anguelov, D. Erhan, et al. (2016), "SSD: Single shot multibox detector", *Computer Vision - ECCV 2016: 14th European Conference, Amsterdam, The Netherlands*, **14**, pp.21-37.
- [26] Z. Liu, Y. Lin, Y. Cao, et al. (2021), "Swin Transformer: Hierarchical vision Transformer using shifted windows", *Proceedings of The IEEE/CVF International Conference on Computer Vision*, pp.10012-10022.
- [27] Z. Liu, H. Hu, Y. Lin, et al. (2022), "Swin Transformer v2: Scaling up capacity and resolution", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.12009-12019, DOI: 10.48550/arXiv.2103.14030.
- [28] P. Shi, Y. Pan, A. Yang (2025), "SS-BEV: Multi-camera BEV object detection based on multi-scale spatial structure understanding", *Signal, Image and Video Processing*, **19**(2), pp.1-13, DOI: 10.1007/s11760-024-03762-5.

- [29] C. Wang, Z. Cheng, A. Akram, et al. (2025), "Gogglesdet: Goggles detection with clustering transformer and a new dataset", *Signal, Image and Video Processing*, **19**(188), DOI: 10.1007/s11760-024-03778-x.
- [30] H. Lin, S. Ming, Y. Liu, X. Qu (2025), "AEFusion: An attention-based ensemble learning approach for BEV fusion perception in autonomous modular buses", *IEEE Transactions on Intelligent Vehicles*, **10**(5), pp.3468-3480, DOI: 10.1109/TIV.2024.3454288.
- [31] C. Hao, Z. Yu, X. Liu, et al. (2025), "A simple yet effective network based on vision Transformer for camouflaged object and salient object detection", *IEEE Transactions on Image Processing*, DOI: 10.48550/arXiv.2402.18922.
- [32] L. Zhang, J. Huang, X. Li, et al. (2018), "Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset", *IEEE Transactions on Image Processing*, **27**(11), pp.5350-5364, DOI:10.1109/TIP.2018.2857407.
- [33] H. Do, J.Y. Choi (2020), "Context-based parking slot detection with a realistic dataset", *IEEE Access*, **8**, pp.171551-171559, DOI: 10.1109/ACCESS.2020.3024668.
- [34] W. Li, L. Cao, L. Yan, et al. (2020), "Vacant parking slot detection in the around view image based on deep learning", *Sensors*, **20**(7), 22pp, DOI: 10.3390/s20072138.
- [35] J.K. Suhr, H.G. Jung (2021), "End-to-end trainable one-stage parking slot detection integrating global and local information", *IEEE Transactions on Intelligent Transportation Systems*, **23**(5), pp.4570-4582, DOI: 10.48550/arXiv.2003.02445.
- [36] Y. Wang, Y. Guan, R. Cao (2022), "DetPS: A fully convolutional end-to-end parking slot detector", *IEEE 17th Conference on Industrial Electronics and Applications (ICIEA)*, pp.1051-1056, DOI: 10.1109/ICIEA54703.2022.10005941.
- [37] J. Deng, W. Dong, R. Socher, et al. (2009), "ImageNet: A large-scale hierarchical image database", *IEEE Conference on Computer Vision and Pattern Recognition*, pp.248-255, DOI: 10.1109/CVPR.2009.5206848.
- [38] Z. Zheng, P. Wang, W. Liu, et al. (2020), "Distance-IoU loss: Faster and better learning for bounding box regression", *Proceedings of The AAAI Conference on Artificial Intelligence*, **34**(7), pp.12993-13000, DOI: 10.48550/arXiv.1911.08287.
- [39] P.R.D. Almeida, L.S. Oliveira, A.S.B. Jr, et al. (2015), "PKLot - A robust dataset for parking lot classification", *Expert Systems with Applications*, **42**(11), pp.4937-4949, DOI: 10.1016/j.eswa.2015.02.009.
- [40] T.Y. Lin, M. Maire, S. Belongie, et al. (2014), "Microsoft COCO: Common objects in context", *Computer Vision - ECCV 2014: 13th European Conference, Zurich, Switzerland*, pp.740-755, DOI: 10.48550/arXiv.1405.0312.
- [41] K. Chen, J. Wang, J. Pang, et al. (2019), "MMDetection: Open MMLab detection toolbox and benchmark", *arXiv*, DOI: 10.48550/arXiv.1906.07155.
- [42] T.Y. Ross, G. Dollar (2017), "Focal loss for dense object detection", *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp.2980-2988.
- [43] G. Amato, F. Carrara, F. Falchi, et al. (2017), "Deep learning for decentralized parking lot occupancy detection", *Expert Systems with Applications*, **72**, pp.327-334, DOI: 10.1016/j.eswa.2016.10.055.
- [44] G. Amato, F. Carrara, F. Falchi, et al. (2016), "Car parking occupancy detection using smart camera networks and deep learning", *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp.1212-1217, DOI: 10.1109/ISCC.2016.7543901.