Communication in Physical Sciences. 2025. 12(2) 562-570

Optimized Fast R-CNN for Automated Parking Space Detection: Evaluating Efficiency with MiniFasterRCNN

Olumide Oni and Kenechukwu Francis Iloeje

Received: 12 December 2024/Accepted: 24 January 2025/Published: 05 February 2025
https://dx.doi.org/10.4314/cps.v12i2.21

Abstract*: Automated parking space detection is a crucial application of computer vision in intelligent transportation systems. In this study, we developed a Fast R-CNN-based model for classifying and localizing parking spaces into empty and occupied categories. The model architecture consists of a pre-trained CNN backbone (ResNet50) for feature extraction, a Region Proposal Network (RPN) for generating potential bounding boxes, and Region-of-Interest (RoI) pooling for feature refinement. The classification head utilizes a softmax activation function with cross-entropy loss, while the bounding box coordinates are refined using smooth L1 loss. To facilitate training, we employed Roboflow for dataset annotation, creating ground truth bounding boxes for parking spaces. The model was fine-tuned using transfer learning, leveraging knowledge from the COCO dataset. Training involved hyperparameter optimization, including learning rate scheduling and weight decay, to enhance convergence. Model selection was based on validation loss and accuracy to ensure generalization to unseen data. The model was deployed using Gradio, allowing real-time parking space detection from uploaded images. Despite achieving a final loss of 0.8280, the model exhibited some background noise distortions, impacting detection accuracy. To address this limitation, we explored a lightweight alternative, MiniFasterRCNN, optimized for efficiency with a simpler architecture. The MiniFasterRCNN was trained on a three-class dataset (empty, occupied, background), achieving a validation accuracy of 77.78%. However, attempts to achieve 100% accuracy proved inefficient, highlighting the need for further improvements, such as segmentation techniques (Masked R-CNN). This research demonstrates the feasibility of Fast R-CNN-based models for parking space detection while emphasizing the importance of architectural optimizations and hyperparameter tuning for improved accuracy and robustness in real-world applications.*

Keywords*: Parking Space Detection, fast R-CNN, transfer learning, computer vision, object detection*

Olumide Oni
897 First Avenue, West Haven,
Connecticut. 06516
Email: Ooni1@unh.newhaven.edu
Orcid id: 0009-0001-6113-3312

Kenechukwu Francis Iloeje
54 Allan St, West Haven, CT 06516
Email: kiloe1@unh.newhaven.edu

1.0 Introduction

The rapid urbanization and continuous rise in vehicle ownership have contributed to increasing traffic congestion and parking challenges in metropolitan areas (Shoup, 2017). Efficient parking management has become a crucial component of urban planning, particularly in institutions such as universities, where a high density of vehicles requires strategic space allocation (Chou & Lin, 2019). As universities serve as hubs for students, faculty, and visitors, managing available parking spaces efficiently can significantly impact accessibility and convenience.

Traditional parking management approaches, such as manual monitoring and ticketing systems, are often inefficient, labor-intensive, and prone to human error (Caicedo et al., 2012). Moreover, these systems contribute to unnecessary vehicular movement as drivers search for available parking spots, leading to increased fuel consumption, air pollution, and driver frustration (Geng & Cassandras, 2013). To mitigate these issues, automated parking space detection systems leveraging artificial intelligence (AI) and computer vision have emerged as a promising solution (Amato et al., 2017).

Recent advancements in deep learning, particularly convolutional neural networks (CNNs), have significantly improved the accuracy and efficiency of object detection tasks, including parking occupancy detection (Zhou et al., 2019). CNN-based models, such as Fast Region-based Convolutional Neural Networks (Fast R-CNN), provide robust solutions for real-time image-based parking space classification (Ren et al., 2015). By integrating AI-driven image analysis, parking detection systems can provide real-time updates on occupancy status, reducing the time spent searching for available spots and improving overall urban mobility (Ye et al., 2020).

This study aims to develop a vehicle parking space detection system using Fast R-CNN trained on the Common Objects in Context (COCO) dataset. By analyzing image data collected from parking lots at the University of New Haven, the proposed system seeks to enhance real-time parking monitoring, reduce traffic congestion, and contribute to more efficient parking space utilization. The findings from this research can be instrumental in guiding the development of smart parking systems and urban mobility strategies.

Several studies have delved into the realm of parking space detection, each offering unique insights and methodologies. Notably, the work of Lin et al. (2014) introduced the Common Objects in Context (COCO) dataset, a comprehensive benchmark for object detection tasks that has significantly contributed to the development and evaluation of models like Fast R-CNN (Ren et al., 2015). The COCO dataset encompasses a diverse range of object categories, including those relevant to our project, such as cars and parking spaces, making it an essential resource for training deep learning models in automated parking detection.

Deep learning techniques have been widely adopted for parking space detection due to their robustness in handling complex visual patterns. Zhang et al. (2018) demonstrated the efficacy of Faster R-CNN in detecting vacant and occupied parking spaces with high precision, showcasing its applicability in real-world scenarios. Similarly, Amato et al. (2017) explored a decentralized approach to parking lot occupancy detection using convolutional neural networks (CNNs), highlighting the scalability of AI-driven methods. The integration of real-time object detection models, such as You Only Look Once (YOLO) (Redmon et al., 2016) and Single Shot MultiBox Detector (SSD) (Liu et al., 2016), has further advanced the field by enabling faster and more efficient parking monitoring systems. Furthermore, hybrid approaches combining deep learning with IoT-based solutions have shown promise in enhancing detection accuracy. Chou and Lin (2019) developed a smart parking system integrating deep learning and IoT sensors, allowing for real-time updates and predictive analytics. Similarly, Ye et al. (2020) applied surveillance camera-based deep learning techniques to improve parking occupancy detection under varying lighting and environmental conditions.

By drawing inspiration from these studies, our research aims to optimize parking space detection in a university setting, leveraging the strengths of deep learning methodologies while addressing challenges such as occlusions, environmental variability, and real-time processing constraints. The implementation of Fast R-CNN on COCO-trained models will enable a scalable and adaptable solution for improving parking efficiency on campus.

2.0 Data Processing

2.1 Data Collection

To develop a robust and accurate parking space detection model, we curated a dataset consisting of 200 high-resolution images of parking lots within the University of New Haven environs. These images were captured at various times of the day and under diverse weather conditions to ensure variability in lighting, shadows, and environmental factors. To mitigate potential biases and enhance the generalizability of our model, we maintained a balanced dataset composition, with 100 images depicting empty parking lots and the remaining 100 showcasing occupied parking spaces.

To improve the quality and representativeness of the dataset, images were sourced from multiple angles and perspectives, including aerial views, ground-level shots, and different vantage points commonly used in surveillance and monitoring systems. This approach aimed to simulate real-world parking scenarios, capturing challenges such as occlusions, varying vehicle sizes, and environmental noise. Additionally, images were collected across different days to account for temporal variations in parking patterns.

For image acquisition, we utilized high-resolution cameras with fixed focal lengths to ensure consistency in image clarity and resolution. Furthermore, we incorporated images from publicly available datasets, such as the PKLot dataset (De Almeida et al., 2015), to enhance the dataset's diversity and allow for cross-validation against existing benchmarks in the field.

The dataset underwent a thorough pre-processing stage to remove low-quality images, such as those affected by excessive glare, blurriness, or obstructions. Image augmentation techniques, including contrast adjustments, rotation, scaling, and Gaussian noise addition, were employed to enhance the dataset's robustness and improve model generalization to new parking scenarios.

By meticulously curating a diverse and high-quality dataset, we aimed to establish a strong foundation for training and evaluating our machine learning model, ensuring its effectiveness in real-world parking space detection applications.

Fig. 1: The images below reveals two samples of our data sets

To facilitate the training of our Fast R-CNN model, we utilized Roboflow for dataset annotation. This involved the creation of ground truth bounding boxes for both empty and occupied parking spaces. The annotated dataset served as a crucial component for training the model to accurately identify and classify parking spaces in new, unseen images.

2.2 Model Implementation

The implementation of the Fast R-CNN model for parking space detection leveraged the COCO dataset, which includes parking space annotations among its diverse object categories (Lin et al., 2014). Utilizing this dataset provided a solid foundation for our specific task while benefiting from its extensive and varied real-world scenarios. Fast R-CNN, introduced by Girshick (2015), is an efficient deep learning model for object detection that improves upon earlier approaches by integrating region proposal generation and classification within a single network, reducing computational complexity.

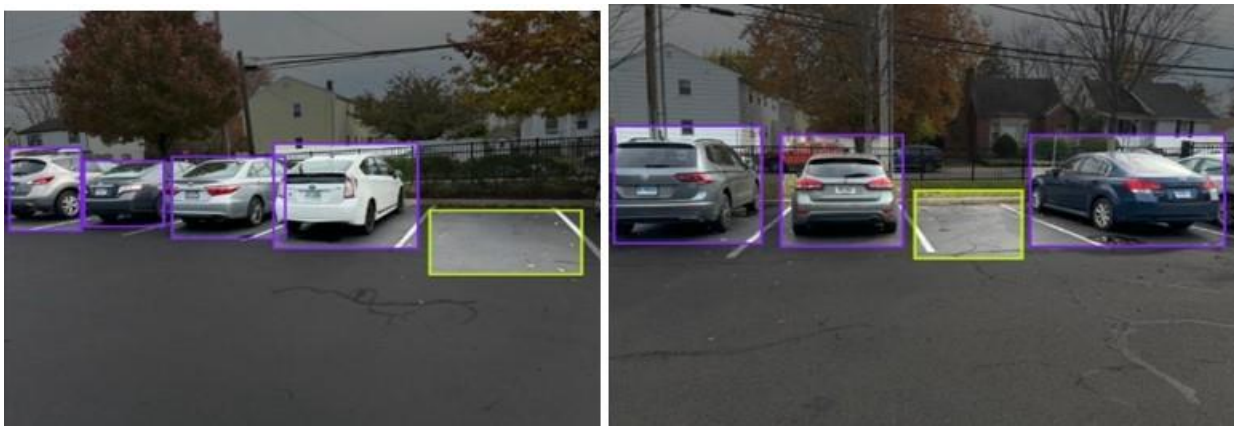

Fig. 2: Image of annotated vehicle parking spaces

2.3 Data Transformation and Augmentation

To enhance model performance, we applied a series of data transformation and augmentation techniques. Before augmentation, images were resized to 640×640 pixels using the Roboflow annotation tool to ensure uniformity and prevent scaling distortions that could lead to inaccurate bounding box placements. The dataset was then converted to a Tensor format, facilitating seamless integration with deep learning frameworks like TensorFlow and PyTorch.

Data normalization was performed to standardize pixel values, ensuring consistent input ranges for the neural network and improving convergence during training. Various augmentation techniques were employed to enhance model robustness and generalization, including:

(i) **Horizontal flipping** – Simulating different viewpoints of parking spaces.
(ii) **Random cropping** – Introducing variability in object positioning.
(iii) **Brightness and contrast adjustments** – Accounting for variations in lighting conditions.
(iv) **Gaussian noise addition** – Making the model more resistant to environmental noise.

The dataset was partitioned into 160 images for training, 20 for validation, and 20 for testing**,** ensuring an appropriate balance for evaluating model performance. Data augmentation significantly increased the effective dataset size, reducing the risk of overfitting and improving the model's ability to generalize across different parking environments (Shorten & Khoshgoftaar, 2019).

2.4 Transfer Learning and Model Architecture

Transfer learning played a pivotal role in our approach, enabling the Fast R-CNN model to leverage pre-trained weights and improve detection accuracy with fewer training samples. Transfer learning involves utilizing knowledge gained from a source task to enhance learning in a related target task. In our implementation, we initialized the Fast R-CNN model with weights pre-trained on the COCO dataset, benefiting from the extensive object annotations, including vehicles and parking spaces (Lin et al., 2014).

The Fast R-CNN architecture consists of the following key components:

(i) **Feature Extraction Layer** – Uses a pre-trained backbone, such as ResNet-50 (He et al., 2016), to extract spatial features from input images.
(ii) **Region Proposal Network (RPN)** – Identifies potential parking space regions for classification.
(iii) **ROI Pooling Layer** – Converts varying-sized regions into fixed-size feature maps.
(iv) **Fully Connected Layers** – Perform classification and bounding box

regression to accurately detect parking spaces.

Fine-tuning was conducted by unfreezing the last few layers of the feature extractor, allowing the model to adapt specifically to university parking environments. This adaptation process enhances the model's ability to distinguish between occupied and empty parking spaces under varying conditions, including occlusions and changes in lighting.

2.5 Model Architecture

[[[[

Our Fast R-CNN architecture is composed of several critical components that work together to efficiently detect parking spaces within images. These components include a convolutional neural network (CNN) for feature extraction, a region proposal network (RPN) for identifying potential object regions, and region-of-interest (RoI) pooling layers for extracting relevant features. The final classification and localization are handled by fully connected layers that predict both the object category and bounding box coordinates.

2.6 Feature Extraction (Backbone Network)

The foundation of our Fast R-CNN model is a pre-trained convolutional neural network (CNN), which serves as the feature extractor. This backbone network, typically based on architectures such as ResNet-50 (He et al., 2016) or VGG-16 (Simonyan & Zisserman, 2015), captures rich hierarchical representations from input images, enabling the model to distinguish between occupied and empty parking spaces effectively.

2.7 Region Proposal Network (RPN)

The Region Proposal Network (RPN) plays a crucial role in object detection by scanning the feature maps generated by the backbone and identifying potential regions of interest. The RPN:

- Uses anchor boxes to propose candidate bounding boxes.
- Predicts adjustments to these boxes to refine localization.
- Generates proposals with objectness scores, indicating the likelihood of containing a relevant object (i.e., an occupied or empty parking space).

This process significantly reduces the number of candidate regions, improving computational efficiency.

2.8 Region-of-Interest (RoI) Pooling

After obtaining region proposals, RoI pooling is applied to extract fixed-size feature maps from each proposed region. Since the bounding boxes generated by the RPN vary in size, RoI pooling ensures that all extracted regions are resized to a consistent spatial dimension, allowing for uniform processing in subsequent layers. This step maintains spatial consistency while preserving important feature information.

2.9 Fully Connected Layers (Classification and Regression)

The extracted RoI features are then passed through a series of fully connected layers, which are responsible for:

(i) **Classification** – Determining whether the detected region contains an occupied or empty parking space.

(ii) **Bounding Box Regression** – Refining the predicted bounding box coordinates to improve localization accuracy.

This final stage ensures that the model accurately classifies parking spaces and precisely locates them within the image (Fig. 3)

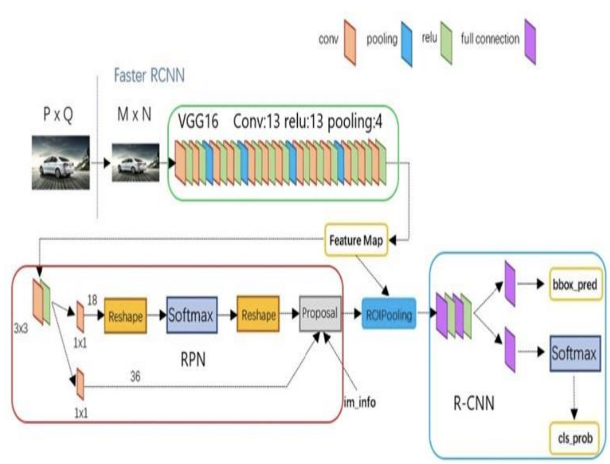

Fig.3: Architecture of faster R-CNN

2.10 Objective Function

The objective function of our Fast R-CNN model comprises two main components: classification and bounding box regression.

Classification Loss: This component utilizes a softmax activation function to assign probabilities to each region proposal being either an empty or occupied parking space. The cross-entropy loss measures the dissimilarity between predicted and ground truth class probabilities.

Bounding Box Regression Loss: To refine the predicted bounding box coordinates, a smooth L1 loss is employed. This loss function minimizes the difference between predicted and ground truth bounding box parameters.

2.11 Parameters and Configuration

The model's parameters include the weights learned during transfer learning, which are fine-tuned during training on our specific parking space dataset. The backbone architecture's parameters, such as the number of filters in each convolutional layer, influence feature extraction. Configuration parameters include anchor box scales and aspect ratios for the RPN, affecting region proposal generation.

2.11 Input and Output Sizes

The input size of our model corresponds to the dimensions of the images in our dataset, typically represented as height × width × channels. The output consists of class probabilities (empty or occupied) for each region proposal and refined bounding box coordinates.

In summary, our transfer learning approach enhances the Fast R-CNN model's capability to detect parking spaces by leveraging knowledge gained from the COCO dataset. The architecture, objective function, and parameter configurations collectively contribute to the model's ability to accurately classify and locate parking spaces in the University of New Haven parking lots.

3.0 Transfer of Experimental Results

The model was optimized for the COCO-format dataset. In order to produce the required number of classes, the last layers of the network had to be adjusted. The model was then trained using the target dataset.

Choosing Hyperparameters

The learning rate, momentum, and weight decay were chosen at random and eventually updated via a step scheduler which basically adjusts the learning rate decreasingly by a specific value every 3 epochs of the entire 20 epochs of the training loop. So based on the policy, we adopted bigger steps when the model was far from the minimum and fewer ones when it was getting close, this improved the model's ability to converge.

Model Selection and Evaluation

In order to make sure the selected model generalized well to unobserved data, it was chosen based on validation loss and accuracy. To provide an unbiased evaluation of the best model's performance, it was tested on a different test set.

The model was deployed using Gradio. A random parking lot image was uploaded and an output image showed the bounding box detections which depicts the detected parking spaces.

```
Epoch 0 - Training loss: 3.489413261413574
Epoch 1 - Training loss: 1.3457867354154587
Epoch 2 - Training loss: 0.808091327548027
Epoch 3 - Training loss: 1.0471802353858948
Epoch 4 - Training loss: 1.0082941502332687
Epoch 5 - Training loss: 0.9243813753128052
Epoch 6 - Training loss: 0.9018815457820892
Epoch 7 - Training loss: 0.8797635287046432
Epoch 8 - Training loss: 0.8477801233530045
Epoch 9 - Training loss: 0.8227691799402237
Epoch 10 - Training loss: 0.8345062285661697
Epoch 11 - Training loss: 0.8144939988851547
Epoch 12 - Training loss: 0.8380615264177322
Epoch 13 - Training loss: 0.825903907418251
Epoch 14 - Training loss: 0.8210687786340714
Epoch 15 - Training loss: 0.8193602114915848
Epoch 16 - Training loss: 0.8153988569974899
Epoch 17 - Training loss: 0.84190833568573
Epoch 18 - Training loss: 0.8154646903276443
Epoch 19 - Training loss: 0.8280725032091141
```

3.1 Final Result

The faster r-cnn resnet50 pretrained model was fine tuned for our task. We trained the model through 20 epochs with variable learning rate and obtained a final loss of 0.8280. The model was only able to detect some bounding box on the test images but was picking up background noises which caused some distortion. This leaves the model somewhat efficient but lacks segmentation techniques which the Masked RCNN solves.

3.2 Mini Network - Mode Architecture

The MiniFasterRCNN is a lightweight version of the Faster R-CNN model designed for classification and object detection tasks. It is a smaller and less complex alternative to the original Faster R-CNN while still retaining its core functionalities.

The model incorporates two convolutional stages. Each utilizes a 3x3 filter, with a stride of 1 and padding of 1, scaling the channel depth from 3 (corresponding to RGB input) to 16, and subsequently to 32 channels.

Weights are initialized following the Kaiming method, while biases are set to zero at the start.

Pooling Stages

Sequentially following each convolutional stage are two max pooling stages. These employ a 2x2 filter and a stride of 2, effectively halving the spatial dimensions of the input.

Flattening Stage

Post the second pooling stage, a flattening operation is executed to transform the 2D feature map into a linear array.

Fully-Connected Stages

The network includes a pair of dense layers. The initial dense layer transitions from an input size determined by the product of 32, final_conv_height, and final_conv_width (a computation derived from the post-pooling feature map dimensions) to an output size of 8 neurons. The final layer accepts the 8-neuron input and projects it onto a space corresponding to the number of classification categories (num_classes). Weights are assigned through a normal distribution, and biases are initialized to zero.

The model was developed to be trained on datasets with a predefined number of classes (3 classes - empty, occupied, and background). For fine-tuning, the network was adapted by setting the num_classes parameter to match the specific dataset's classes.

Hyperparameter Selection

Hyperparameters such as learning rate, momentum, and weight decay are crucial for the training process. These were selected through a systematic process using Ray Tune, allowing for an efficient search through the hyperparameter space.

Learning Rate Decay Policy

The learning rate decay policy applied was a step decay, which reduces the learning rate by a certain factor after a specified number of epochs. This policy aids in stabilizing the training as the model approaches the minimum of the loss function.

```
# Validation
validation_loss, validation_accuracy = validate(model, valid_loader, criterion)
print(f'Epoch [{epoch + 1}/{num_epochs}], Validation Loss: {validation_loss:.4f

Epoch [1/5], Training Loss: 285475.8479
Epoch [1/5], Validation Loss: 1.0819, Validation Accuracy: 77.78%
Epoch [2/5], Training Loss: 1.0707
Epoch [2/5], Validation Loss: 1.0544, Validation Accuracy: 77.78%
Epoch [3/5], Training Loss: 1.0459
Epoch [3/5], Validation Loss: 1.0259, Validation Accuracy: 77.78%
Epoch [4/5], Training Loss: 1.0208
Epoch [4/5], Validation Loss: 1.0003, Validation Accuracy: 77.78%
Epoch [5/5], Training Loss: 0.9974
Epoch [5/5], Validation Loss: 0.9782, Validation Accuracy: 77.78%
```

Model Selection and Evaluation

Model selection was based on the performance metrics obtained from the validation dataset. The best-performing model was then evaluated on a separate test set. Evaluation metrics such as accuracy, precision, and F1-score provided a comprehensive assessment of the model's predictive capabilities.

3.3.1 Final Result

An attempt to overfit the model to obtain a

100% accuracy proved inefficient. An accuracy of 73% was achieved. However, after training the model on the hyperparameters we obtained a validation accuracy of 77.78%.

4.0 Conclusion

In summary, the development and evaluation of the MiniFasterRCNN model provided valuable insights into model optimization and performance assessment. The model selection, based on validation loss and accuracy, led to a well-generalizing model, achieving a validation accuracy of 77.78%. Initial attempts to achieve perfect accuracy resulted in a 73% accuracy rate, highlighting the challenges and limitations in model training.

The deployment using Gradio demonstrated the model's practical application in detecting parking spaces, although it encountered issues with background noise. This limitation was also observed in the fine-tuned Faster R-CNN ResNet50 model, which, despite a promising final loss of 0.8280, showed a need for enhanced segmentation capabilities.

5.0 References:

Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., & Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, *72*, pp.327-334.

Caicedo, J., Blazquez, C., & Camacho, T. (2012). Evaluation of parking-garage capacity expansion: A case study. *Transport Policy*, *24*, pp.131-137.

Chou, C. C., & Lin, C. J. (2019). A smart parking service based on deep learning and IoT. *Journal of Advanced Transportation*, *2019*, pp.1-10.

De Almeida, P. R., Oliveira, L. S., Britto, A. S., Silva, E. J., & Koerich, A. L. (2015). PKLot–A robust dataset for parking lot classification. *Expert Systems with Applications*, *42*, *11, pp.* 4937-4949.

Geng, Y., & Cassandras, C. G. (2013). A new “smart parking” system based on optimal resource allocation and reservations. *IEEE Transactions on Intelligent Transportation Systems, 14, 3, pp.* 1129-1139.

Girshick, R. (2015). *Fast R-CNN*. In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440-1448. https://doi.org/10.1109/ICCV.2015.169

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. https://doi.org/10.1109/CVPR.2016.90.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. https://doi.org/10.1109/CVPR.2016.90

https://drive.google.com/drive/folders/1MXQaWOkAg1iI6Uo1WP12Ig7eP7sZNN7N?usp=shari ng

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision (ECCV)*, pp.740-755.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *arXiv preprint arXiv:1405.0312.*

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision (ECCV)*, pp.21-37.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6517-6525). Honolulu, HI, USA. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.779-788.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 91-99.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems (NeurIPS).

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0.

Shoup, D. C. (2017). *The high cost of free parking*. Routledge.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*

Ye, H., Deng, Y., Huang, X., Song, Y., & Duan, Y. (2020). Parking occupancy detection based on deep learning with surveillance cameras. *IEEE Transactions on Industrial Informatics*, *16*, 7, pp. 4524-4533. Zhang, H., Lin, C., Zhang, D., & Zhang, W. (2018). Parking space detection using Faster R-CNN. *IEEE International Conference on Artificial Intelligence and Big Data*, pp. 289-294.

Zhou, X., Yu, R., Wen, J., & Jin, R. (2019). Parking space detection based on convolutional neural networks. *IEEE Access*, *7*, pp. 42392-42400.

Compliance with Ethical Standards Declaration

Ethical Approval

Not Applicable

Competing interests

The authors declare that they have no known competing financial interests

Funding

The authors declared no source of funding

Authors' Contribution

Both author were involved in all stages of the projects that produce this manuscript and in the writing and corrections at all stages