Vehicle and Parking Space Detection for Smart Parking Systems Using the YOLOv5 Method

Aditya Eka Saputra[a], Bernat Kristian S. Giawa[b] and Rajes Khana[c]

Department of Electrical Engineering, Universitas 17 Agustus 1945 Jakarta, Jl. Sunter Permai Raya, Sunter Agung, Tanjung Priok, Jakarta Utara, Indonesia

Informatics Department, Faculty of Engineering and Informatics, Universitas 17 Agustus 1945, Jakarta, Indonesia

Keywords: Vehicle Detection, Parking Lot, YOLOv5 Method, Parking Efficiency, Smart Parking System.

Abstract: The YOLOv5 network architecture has the advantage of fast and accurate object detection speed and has high real-time object detection capabilities. This research utilizes the YOLOv5 (You Only Look Once version 5) method to detect vehicles and parking spaces in a smart parking system. The main aim of this research is to increase the efficiency of parking space use. The research involved collecting and processing image data from a variety of different parking locations, which was used to train the YOLOv5 model. The proposed network is trained and evaluated on the Parking Lot dataset. The results of the YOLOv5s_Ghost experiment with a car vehicle detection confidence value of 93.0% and available space detection confidence of 94.0%. Using the best weights from YOLOv5s_Ghost increases the mean Average Precision (mAP) value to 94.9%, slightly above YOLOv5s which reaches 94.7%. The YOLOv5s_Ghost architecture shows a high level of accuracy in vehicle and parking space detection, even in various lighting conditions from morning to evening in the smart parking system. YOLOv5s_Ghost uses the GhostNet module, can be transferred to other classic models with comparable performance while reducing the number of parameters, optimizing computing resources, and increasing mAP and reducing loss.

1 INTRODUCTION

In this modern era, with population growth and an increase in the number of cars, the number of parking spaces is also increasing in all big cities in the world. In most parking locations, ground sensors are used to monitor the condition of the various parking spaces. Traditional methods for detecting parking spaces involve ultrasonic technology (Shao et al., 2018), geomagnetics (Zhou & Li, 2014), and infrared ray (H. C. Chen et al., 2017; Li & Lin, 2019). This requires the installation and maintenance of sensors in each parking area, especially in parking lots with a large number of spaces, which may result in high costs. Although this method can produce a higher level of accuracy, it is relatively expensive.

To overcome these challenges, intelligent parking space detection technology has emerged as an innovative solution. One of the methods proposed in this paper uses deep learning-based object recognition technology, specifically the You Only Look Once (YOLO) model version 5. This model is very efficient in detecting objects in images and videos in real-time.

This research focuses on implementing the YOLOv5 method for detecting available parking spaces and cars in the parking area. Using this technology, it can accurately identify empty parking spaces and parked vehicles, providing real-time information to parking users. This research seeks to increase the efficiency of parking space use and reduce losses due to time wasted looking for a suitable parking space.

2 RELATED WORK

Object detection is a technique in image or video

[a] https://orcid.org/0009-0009-2430-9030
[b] https://orcid.org/0009-0001-5897-4102
[c] https://orcid.org/0000-0001-6062-6492

Saputra, A., Giawa, B. and Khana, R.
Vehicle and Parking Space Detection for Smart Parking Systems Using the YOLOv5 Method.
DOI: 10.5220/0012584700003821

In *Proceedings of the 4th International Seminar and Call for Paper (ISCP UTA '45 JAKARTA 2023)*, pages 458-466
ISBN: 978-989-758-691-0; ISSN: 2828-853X

processing that allows the recognition and determination of the location of objects in an image or video. Essentially, the concept of object detection involves scanning the entire image area to identify parts that contain objects and parts that are the background (Salim, 2020).

In recent years, deep learning-based object detection methods have become increasingly significant. K.Simonyan and A.Zisserman (Chung et al., 2018) developed a very deep CNN convolutional network, known as VGG, for object classification. Research shows that VGG models can generalize well across a wide range of tasks and datasets, matching or outperforming more complex recognition pipelines built on less deep image representations. These results emphasize the importance of depth in visual representation.

RCNN (Girshick et al., 2014) (Region Proposal-Convolutional Neural Network) is an object detection method that combines Region Proposal with Convolutional Networks. This is the first time that deep learning has been used in a conventional object detection task. The best performing systems are complex ensembles that combine multiple low-level image features with high-level context from object detectors and scene classifiers. This research presents a simple and scalable object detection algorithm that provides a relative improvement of 30% compared to the previous best results in PASCAL VOC 2012. This research achieves performance through two insights. The first is to apply a high-capacity convolutional neural network to bottom-up region proposals to localize and segment objects. The second is a paradigm for training large CNNs when labeled training data is scarce. This research shows that it is very effective to first train a network with supervision for an additional task with a lot of data (image classification) and then fine-tune the network for a target task where data is scarce (detection). It then conjectured that the "supervised pre-training/domain-specific refinement" paradigm would be highly effective for a variety of data-deficient vision problems.

S. Ren et al (Ren et al., 2017) introduced a faster R- CNN, which is more efficient compared to RCNN. Faster R-CNN eliminates the selective search step of RCNN by introducing RPN networks. RPN allows region proposal, classification, and regression to share common convolutional features, thus speeding up the detection process. Nevertheless, Faster R-CNN still involves two stages: first, determining the presence of targets within the frame area, then identifying those targets. This research has presented RPN for making efficient and accurate regional proposals. By sharing convolutional features with downstream detection networks, the region proposal step is almost cost-free. This method allows a unified deep learning-based object detection system to run at 5-17 fps. The learned RPN also improves the quality of region proposals and overall object detection accuracy.

YOLO (Redmon et al., 2016) combines object discrimination and object recognition into one step, which improves the detection speed. YOLOv5 (You Only Look Once version 5) is a real-time object recognition algorithm based on deep learning. The YOLOv5 algorithm has advantages in terms of speed and accuracy in object detection. YOLOv5 hasfast performance in detecting objects. This means it is capable of real-time detection, even on devices with limited resources. Even though it is fast, YOLOv5 also maintains a good level of accuracy in detecting objects (Iskandar Mulyana & Rofik, 2022).

In managing parking lots, identifying whether a parking space is empty or not is an additional challenge besides detecting vehicles. A number of studies have been carried out to overcome this problem. M.Ahrnbom et al. (Ahrnbom et al., 2016) took features such as color and gradient size in LUV space, then trained an SVM-based classifier to classify the status of parking lots, whether they are empty or occupied. Giuseppe Amato et al. (Amato et al., 2016) used a CNN (convolutional neural network) to train a detector capable of detecting parking lots and their status based on LBP features. Tom Thomas et al. (Thomas & Bhatt, 2018) developed a binary classifier convolutional neural network to determine whether a parking space is occupied or not. Meanwhile, Cheng-Fang Peng (Peng et al., 2018) takes three new features from each parking space, namely vehicle color characteristics, local gray scale variation features, and corner features, to assess occupancy status. They trained a deep neural network to determine the occupancy status of each parking space based on these three features. Another system (Amato et al., 2016) periodically captures images of several parking lots, and for each parking lot, the occupancy status is determined using a pre-trained CNN. However, in this method, the image captured by the camera must be filtered through a mask that identifies different parking spots. However, making these masks must be done manually by humans, which means it is necessary to make manual masks for each parking space at different parking locations.

3 MODEL METHOD

3.1 YOLO v5 Algorithm

The YOLO object detection algorithm is a one- stage object detection algorithm first proposed by Redmon J. This algorithm eliminates the candidate box extraction step present in the two- stage algorithm, and combines bounding boxes and classification into a single regression problem. The process of the YOLO algorithm is as follows: first, the image is divided into an S×S grid. Each grid is responsible for predicting the presence of targets and determining where the actual center point is within the grid. From each of these grids, several bounding boxes S × S × B are generated. Each bounding box has five parameters: the coordinates of the target center point, the dimensions of the width and height of the target (x, y, w, h), and the confidence whether the target is there or not. Each S × S grid also predicts the probability of the possible target categories within it. The confidence of the predicted bounding boxes and the category probabilities are then multiplied to obtain a category score for each predicted box. These prediction boxes are then filtered using the non-maximum suppression (NMS) method to obtain the final prediction results. The YOLO series algorithm has experienced rapid development in recent years. In 2020, two versions of YOLO appeared successively, namely YOLO v4 and YOLO v5. YOLO v5 successfully achieves a precision accuracy of nearly 50 mAP in the COCO dataset (Lin et al., n.d.) while maintaining operating speed. In the context of vehicle detection in a highway monitoring environment, this chapter selects a small version of YOLO v5 as the reference network model, with the aim of improving the accuracy of the detection algorithm.

YOLO v5 is the most advanced version of the YOLO object detection algorithm. Based on the YOLO v3 and YOLO v4 algorithms, there is innovation in set arithmetic to increase detection speed. YOLO v5 adopts the anchor box concept to improve the efficiency of the R-CNN algorithm, and the manual selection approach of anchor boxes is abandoned. K-means clustering is carried out on the bounding box dimensions to obtain more optimal prior values. In 2020, Glenn Jocher introduced YOLO v5. This network structure consists of input, backbone, neck, and prediction, as seen in Figure 1.

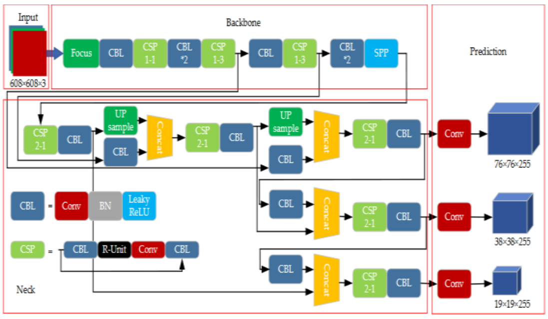


Figure 1: YOLO v5 network structure.

1) The input is the vehicle image input link, which is divided into three parts: Data enhancement (De, n.d.), image size processing (Shorten & Khoshgoftaar, 2019), and anchor frame automatic adaptation (Devkota et al., 2022). In traditional YOLO v5, mosaic data enhancement technique is used to combine inputs by randomly zooming, cropping, arranging and merging images, with the aim of improving small target detection capabilities. When training a dataset, the size of the input images is adjusted to a uniform size and then fed into the model for analysis. The initial size of the dataset is set to 460 × 460 × 30. The initial anchor frames for YOLO v5 are (116, 90, 156, 198, 373, 326).
2) The backbone network consists of two structures, namely the Focus structure (Yang et al., 2018) and the CSP structure (Guo et al., 2022). The Focus structure is tasked with cropping the image before it enters the main part of the network. As shown in Figure 10, the original image with size 608 × 608 × 3 is divided into small chunks. With this, a feature map of size 304 × 304 × 12 is generated, and then through a convolution operation with kernel 32, a new feature map is formed. The Focus operation can reduce the dimensions of the input sample without using additional parameters, making it possible to retain as much information as possible from the original image. The CSP structure, on the other hand, imposes transitions on the input features by using two 1*1 convolutions. This approach helps improve the learning capabilities of CNNs (Y. Chen & Yuan, 2020), overcome computational bottlenecks, and reduce the required memory load.
3) The Neck is a network layer that integrates image features and passes them to the prediction layer. In YOLO v5, the Neck uses the FPN+PAN structure. FPN takes high-level feature information and combines it from top to bottom to form a feature map that is used in the prediction process. Meanwhile, PAN is a basic pyramid that transmits position characteristics strongly from bottom to top (Dong & Xing, 2018). Thus, this structure allows efficient

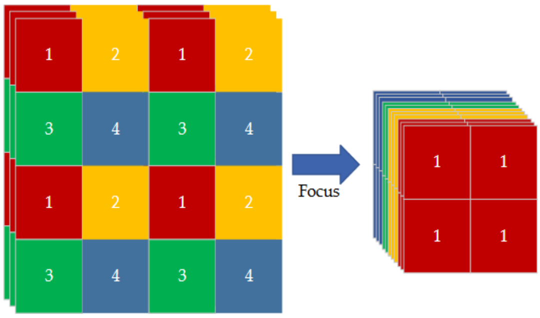


Figure 2: Processing flow of Focus module.

communication and integration of feature information in order to prepare predictions.

4) The prediction layer is tasked with processing image features and generating bounding boxes to predict categories. In YOLO v5, GIOU_Loss is used as the loss function to determine the box boundaries. In overlapping object detection situations, GIOU_NMS is more efficient compared with traditional non- maximum suppression (NMS) methods.

3.2 Research Flowchart

Researchers used Google Colab to implement YOLOV5, which is the latest development of the YOLO network designed to detect objects in images (Tan et al., 2021). In essence, the aim of object detection is to identify the location of objects in the image and classify their type. In other words, the process involves using images as input, followed by creating bounding box vectors and predicting object classes in the output (Wei et al., 2020).

Figure 3 explains the flow of the research stages as follows:

a. Dataset Collection

In this research, researchers used a custom dataset that was collected personally. Dataset collection is the process of collecting images in the form of pictures or images obtained from video recordings taken in UTA'45 Jakarta car parks. The video recording results were then extracted using Roboflow software.

b. Dataset Labeling Process

The labeling process is the stage where all images in the dataset are labeled so they can contain the image name. Labeling is done by creating a bounding box for the object you want to mark in the image. Make sure the bounding box surrounds the object correctly

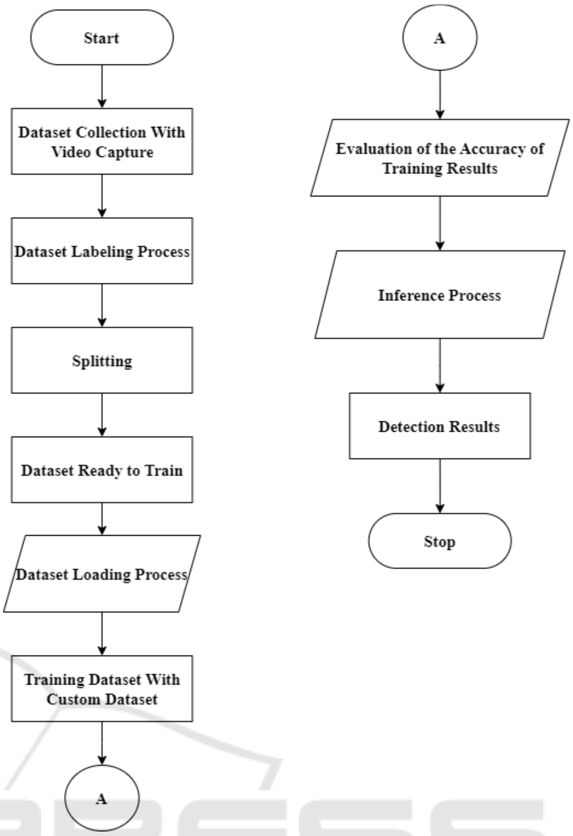


Figure 3: Research Flowchart.

and precisely so that it covers all the objects in question. After adding a bounding box to the object, add a class name or label that corresponds to the marked object. These could be labels like "car" and "space available." After adding the bounding box and giving the object a class name, make sure to save the annotation or label. Roboflow will store information about the locations, object types, and labels that you have added to the dataset. After labeling the entire dataset, make sure to save the labeled dataset with appropriate annotations.

c. Splitting

The Data Splitting process is the process of dividing a dataset into different subsets for use in certain stages in machine learning or model evaluation. This division is generally carried out for the purposes of testing, validation and model training. A common split is 70-80% for training datasets, 10-15% for validation, and 10-15% for test, but these proportions can vary depending on the size of the dataset and project needs. The dataset used in this research has a total of 650 images. To be adaptive to the training process, this work reduces the image size to 640 × 640 pixels and converts the standard dataset format to YOLOv5 format. Based on the 650 image dataset, it is divided into 3 parts, namely Train, Val and Test. The distribution of the dataset can be seen in the table 1 below:

Table 1: Dataset Splitting.

Distribution	Percentage	Total Image
Train	70%	454 images
Val	20%	132 images
Test	10%	64 images

d. Dataset Ready to Train

A dataset that is ready for training is a collection of images data that has gone through a previous process where each image has been given an annotation or label that explains what objects are in it. This annotation usually takes the form of a bounding box that marks the location of the object, as well as a class or label that identifies the type of object.

This dataset has been prepared to be used on computing platforms such as Google Colab, which is a development environment that can be accessed online. Apart from that, this dataset will be utilized by applying the YOLOV5 method. YOLOV5 is an approach or technique in developing object detection models that makes it possible to detect objects quickly and accurately in images.

e. Dataset Loading Process

The dataset input stage is the step where the collection of available car and space datasets that have gone through the roboflow process are uploaded to Google Colab. This process has great importance because the quality of the dataset must be prepared as best as possible to ensure object detection has stability and a high level of accuracy. The dataset used is a collection of images of cars and available spaces in parking lots which have been annotated with labels on each image.

f. Training Dataset with Custom Dataset

After the dataset created for training is fulfilled. The next step is to train the data using the Google Colab system. In YOLOv5 training involves cloning data from the YOLOv5 ultralytic GitHub repository and using the YOLOv5s and YOLOv5s_Ghost models. YOLOv5 is characterized by 213 layers containing a total of 7,225,885 parameters. Batch size can be adjusted to a range of 16, 24, and 40, and training can last for 100, 300, or 500 epochs. The YOLOv5 algorithm uses technologies known as IOU (Intersection Over Union) and Non-max Suppression. This technology is used to measure the ratio between the bounding boxes of predicted objects and the base annotation, where IOU > 0.5-0.9 is considered acceptable. In this context, if the object's confidence value is more than 0.5, a bounding box will be assigned to the object. However, if the object's confidence value is less than 0.5, the object is considered as background or an area that has no detected objects.

We use a dataset of cars and available spaces in parking lots that we have created ourselves using roboflow to pre-train the network, we use this dataset to fine-tune the network to detect vehicles and available spaces in parking locations. The network parameters are refined by using the training set images in the collection on the smart parking dataset, so that the detection effect of the entire network is optimized. Several experimental parameters were set as shown in Table 2.

Table 2: Description of network parameters.

Parameter Name	Parameter Value
Learning Rate	0.005
Epoch	2000
Batch Size	16
Img Size	640
Momentum	0.937
Weight Decay	0.0005

g. Evaluation of the Accuracy of Training Results

The accuracy evaluation process is a step to assess the level of accuracy of model training on the dataset. This stage has an important role in object detection because detection stability requires a high level of accuracy. Therefore, assessing the accuracy value in object detection is very important to make object detection more stable in its accuracy value in images or videos. Several metrics have been used to assess the performance of deep learning detection models. Precision (P) is the proportion of True Positives among all detected Positives (Padilla Carrasco et al., 2023):

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)}$$

True Positive (TP) is a result obtained from a machine prediction which states that this is the correct answer (true). Meanwhile, False Positive (FP) is the result of an answer obtained from a machine prediction which states it is correct but is a wrong answer. Recall is a matrix used to measure how good the model that has been created is. The Recall matrix equation can be written as follows:

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)}$$

False Negatives (FN) describes the number of positive objects that are present in the dataset, but the model incorrectly detects them as negatives or fails to recognize them. The recall matrix is a marker of how well the model performs when the data categories are imbalanced. Therefore, in calculating False Negatives (FN) for recalls, FN is part of the denominator used to calculate the proportion of positive objects that failed to be detected by the model. In other words, False Negatives reflect model errors in identifying existing positive objects. This is a common standard practice in object detection applications (Redmon et al., 2016).

In this context, mAP.5 and mAP.95 reflect the average Average Precision of all detections with an Intersect of Union (IoU) of 50% and 95%, respectively. IoU is the result of the intersection of two bounding boxes, namely those detected by the model and the ground truth, which are then normalized by the combination of the two. Average precision (AP) is calculated for each class-specific detection with an IoU greater than 50% or 95%. Finally, Average Average Precision (mAP) is calculated using the average of all classes.

Additionally, in the validation process, different types of errors related to bounding boxes (Box), (Obj), and (Cls) are calculated, as seen in Table 2. Box Error is measured using the Index of Similarity (IoU), which is the result of from the intersection of model predictions and ground truth, which is then normalized by the combined area of both. Obj error refers to the objectivity score, which is used to estimate the probability that a bounding box is an actual object. Meanwhile, the Cls error is related to the multi- classification score. Obj and Cls error calculations use the Focal Loss function, which is an extension of the cross-entropy loss function. Focal Loss is used to reduce the impact of easy examples and redirect training to more difficult negative cases. There are also other metrics that assess model efficiency, such as inference speed which is often measured in frames per second (FPS), and the number of parameters which generally indicate good model complexity.

h. Inference Process

The next step in entering images or videos is the process where the images or videos that will be tested for object detection are entered into the Google Colab system. The images used in this step involve various cars and available spaces in the parking lot and videos taken in the parking lot environment.

i. Detection Results

The detection results stage is the result of applying object detection to an image or video using the YOLOV5 method. These results show the car objects and available space that were successfully detected in the image or video, along with the detection accuracy value.

4 EXPERIMENT RESULTS

YOLOv5 is the latest version of the superior YOLO object detection algorithm with high detection capability, fast accuracy and good real- time performance. YOLOv5 presents five different models, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, where YOLOv5s has the smallest model size (Tian & Liao, 2021). This research uses 2 YOLOv5 architectures, namely YOLOv5s and YOLOv5s_Ghost. The comparison results of car object detection and available space of both models (YOLOv5s model and YOLOv5s_Ghost model) are shown in Table 3 and Figure 4. In addition, this table shows the Precision, Recall, F-1 score, and mAP of the YOLOv5s and YOLOv5s_Ghost architectural models. We compare based on the value of the best 2000 epoch results. To evaluate the model performance objectively, the mAP (Mean average precision) values were compared. The mAP value of the YOLOv5s model is 94.7%, and that of YOLOv5s_Ghost is 94.9%. Overall, it can be seen that the YOLOv5s_Ghost model has advantages over the YOLOv5s model.

Table 3: Comparison table of best performance by models.

Model	Precision	Recall	F-1 Score	mAP (0.5)
YOLOv5s	98.3	96.0	93.0	94.7
YOLOv5s-Ghost	98.9	96.0	93.0	94.9

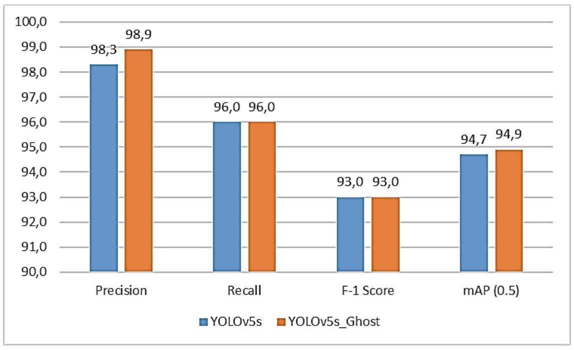


Figure 4: Comparison graph of result values for Original YOLOv5s and YOLOv5s_Ghost model.

Part of Figure 5 shows a graph of the metrics curve as training progresses. After evaluation, the YOLOv5s model has a validation precision score of 98.3%, a recall score of 96.0%, an F1 score of 93.0%, and a mAP score of 94.7%.

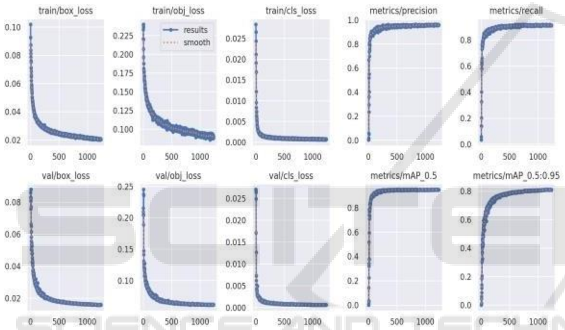


Figure 5: Graph of result values changes in key indicators according to the epochs of training for YOLOv5s model.

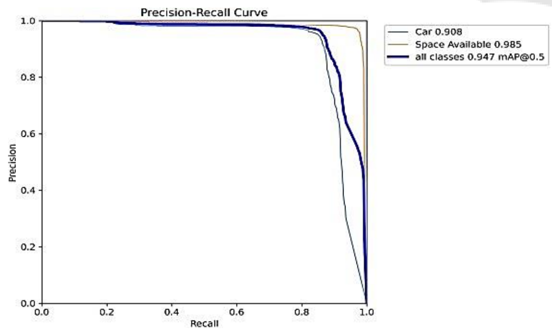


Figure 6: Graph of result values Precision–Recall curve for YOLOv5s model.

As a result of the training and validation process, we found that the YOLOv5s_Ghost model was the best. Thus, the final prediction is made based on the weights obtained from the trained YOLOv5s_Ghost model, which is considered to have the best performance. Part of Figure 7 shows a graph of the metrics curve as training progresses. After evaluation, the YOLOv5s_Ghost model has a validation precision score of 98.9%, a recall score of 96.0%, an F1 score of 93.0%, and an mAP score of 94.9%. These results confirm the effectiveness of our approach in correctly predicting experiments conducted in multiple environments.

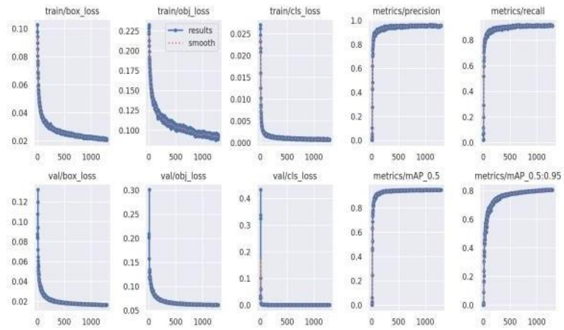


Figure 7: Graph of result values changes in key indicators according to the epochs of training for YOLOv5s_Ghost model.

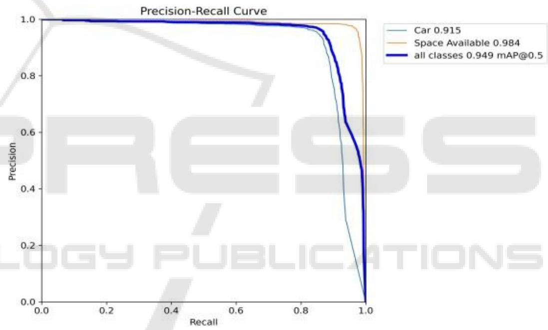
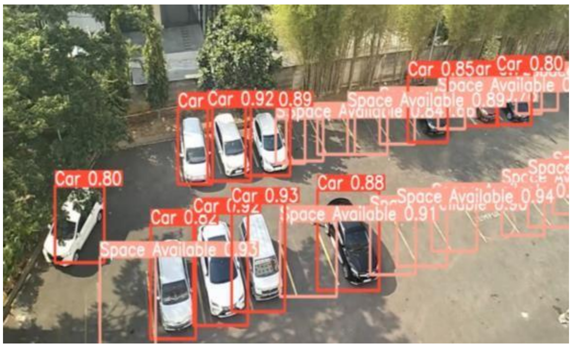


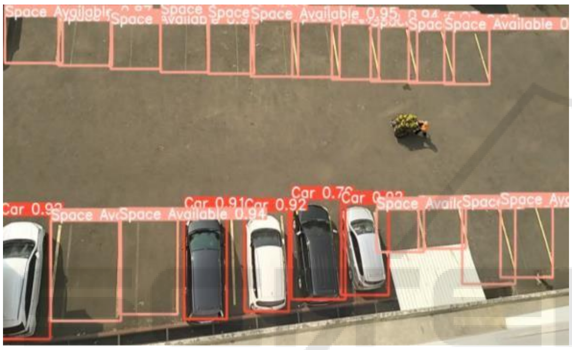
Figure 8: Graph of result values Precision–Recall curve for YOLOv5s_Ghost model.

The Precision–Recall curve is a method of evaluating the performance of an object detector due to changes in the confidence level threshold value. The confidence level is a value that tells the user how confident the algorithm is about the detection. In other words, the closer the number is to 1, the more confident the model is in detecting the target object. Part of Figure 8 is a graph of the Precision–Recall curve of the YOLOv5s_Ghost model. It can be seen that the space available value is 98.4% which is quite high.

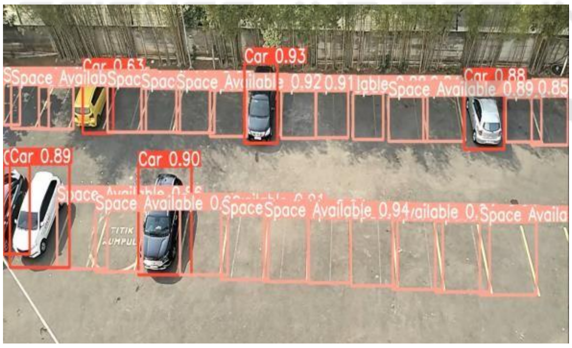
The object detection results of the YOLOv5s_Ghost model can be seen in Table 4. Among the detected objects, available space with a confidence of 94.0%. Available space detection is calculated as 96.6% for Precision, 97.0% for Recall, 96.8% for F1-Score, and finally 98.4% for mAP. This means that the available space detection rate is quite high. Meanwhile, the car detection value has a confidence of 93.0%. Car detection is calculated as 95.0% for Precision, 85.2% for Recall, 89.8% for F1-Score, and finally 91.5% for mAP value.



(a)



(b)



(c)

Figure 9: (a,b and c) Detection results of car and space available using YOLOv5s_Ghost model.

Table 4: Key indicators of YOLOv5s_Ghost model.

Parameter	Car	Space Available	Total
Precision / %	95.0	96.6	95.8
Recall / %	85.2	97.0	91.1
F1-Score / %	89.8	96.8	93.3
mAP (0.5) / %	91.5	98.4	94.9

5 CONCLUSIONS

This paper proposes a car and parking space detection method based on the YOLOv5 algorithm. In this research, we successfully detected car and available space using two models and carried out a comparison between the YOLOv5s and YOLOv5s_Ghost models, and through training, the model was selected for the YOLOv5s_Ghost model with good performance. Then the best weights obtained through validation are applied to the YOLOv5s_Ghost model and tested. As a result, we find that the mAP has increased to 94.9% compared to the YOLOv5s model with an mAP value of 94.7% and the difference in the increase of the YOLOv5s_Ghost model is slight. In the car and available space detection test, the highest confidence value was obtained, namely the car was 93.0% and the available space confidence value was 94.0%. In YOLOv5s_Ghost there is a GhostNet module, which is a plug-and-play module that is easy to transfer to other classic models while maintaining comparable performance. Adding the GhostNet module by reducing the number of parameters in the model, so it requires less computing resources and adding only the head is enough for the detection task on the embedding device to produce higher mAp and lower loss. If there is enough memory to embed the device, it is still necessary to consider accuracy and parameters. However, during the test there were several cars and the available space was not detected because the dataset used in this research was 650 images. The author suggests increasing the dataset size and variation, as well as conducting more experiments at the training stage in order to achieve more optimal model results and a higher level of accuracy. Furthermore, the authors suggest using the YOLOv8 model architecture for further research.

REFERENCES

Ahrnbom, M., Astrom, K., & Nilsson, M. (2016). Fast Classification of Empty and Occupied Parking Spaces Using Integral Channel Features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1609–1615. https://doi.org/10.1109/CVPRW.2016.200

Amato, G., Carrara, F., Falchi, F., Gennaro, C., & Vairo, C. (2016). Car parking occupancy detection using smart camera networks and Deep Learning. *Proceedings - IEEE Symposium on Computers and Communications*, *2016- Augus* (Dl), 1212–1217. https://doi.org/10.1109/ISCC.2016.7543901

Chen, H. C., Huang, C. J., & Lu, K. H. (2017). Design of a non-processor OBU device for parking system based on infrared communication. *2017 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-TW 2017*, 297–298. https://doi.org/10.1109/ICCE-China.2017.7991113

Chen, Y., & Yuan, L. (2020). *Dynamic Convolution : Attention over Convolution Kernels*. 11027– 11036. https://doi.org/10.1109/CVPR42600.2020.0110 4

Chung, C., Patel, S., Lee, R., Fu, L., Reilly, S., Ho, T., Lionetti, J., George, M. D., & Taylor, P. (2018). Published as a conference paper at ICLR 2015 Very Deep Convolutional Networks For Large-Scale Image Recognition

Karen. *American Journal of Health-System Pharmacy*, *75*(6), 398–406.

De, E. (n.d.). *Deep-Learning-Based Image Reconstruction and Enhancement in Optical Microscopy*. 1–21. https://doi.org/10.1109/JPROC.2019.2949575

Devkota, P., Manda, P., Devkota, P., Mohanty, S. D., & Manda, P. (2022). *Deep learning architectures for recognizing ontology concepts from scienti c literature Deep learning architectures for recognizing ontology concepts from scientific literature*.

Dong, N., & Xing, E. P. (2018). *Few-Shot Semantic Segmentation with Prototype Learning*. 1–13.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 580–587. https://doi.org/ 10.1109/CVPR.2014.81

Guo, Y., Zeng, Y., Gao, F., Qiu, Y. I., Zhou, X., Zhong, L., & Zhan, C. (2022). Improved YOLOV4-CSP Algorithm for Detection of Bamboo Surface Sliver Defects With Extreme Aspect Ratio. *IEEE Access*, *10*, 29810–29820. https://doi.org/10.1109/ACCESS.202 2.3152552

Iskandar Mulyana, D., & Rofik, M. A. (2022). Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5. *Jurnal Pendidikan Tambusai*, *6*(3), 13971–13982. https://doi.org/10.31004/jptam.v6i3.4825

Li, Y., & Lin, G. (2019). Design of intelligent parking lot based on Arduino. *IOP Conference Series: Materials Science and Engineering*, *490*(4), 3596–3601. https://doi.org/10.1088/1757-899X/490/4/042010

Lin, T., Zitnick, C. L., & Doll, P. (n.d.). *Microsoft COCO : Common Objects in Context*. 1–15.

Padilla Carrasco, D., Rashwan, H. A., Garcia, M. A., & Puig, D. (2023). T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks. *IEEE Access*, *11*(March), 22430–22440. https://doi.org/10.1109/ACCESS.2021.3137638

Peng, C. F., Hsieh, J. W., Leu, S. W., & Chuang, C. H. (2018). Drone-based vacant parking space detection. *Proceedings-32nd IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2018*, *2018-Janua*, 618–622. https://doi.org/10.1109/WAINA.2018.00155

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/ 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.25 77031

Salim, A. (2020). *Object Detection (Case: Plat Detection)*. https://medium.com/bisa-ai/object-detection-case-plat-detection-7cb5f53682ae

Shao, Y., Chen, P., & Cao, T. (2018). A grid projection method based on ultrasonic sensor for parking space detection. *International Geoscience and Remote Sensing Symposium (IGARSS)*, *2018-July*, 3378–3381. https://doi.org/10.1109/IGARSS.2018.8519022 Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*. https://doi.org/10.1186/s40537-019-0197-0 Tan, S., Lu, G., Jiang, Z., & Huang, L. (2021).

Improved YOLOv5 network model and application in safety helmet detection. *ISR 2021 - 2021 IEEE International Conference on Intelligence and Safety for Robotics*, 330–333. https://doi.org/10.1109/ ISR50024.2021.941956 1

Thomas, T., & Bhatt, T. (2018). Smart Car Parking System Using Convolutional Neural Network. *Proceedings of the International Conference on Inventive Research in Computing Applications, ICIRCA 2018*, *Icirca*, 172–174. https://doi.org/10.11 09/ICIRCA.2018.8597227

Tian, M., & Liao, Z. (2021). Research on Flower Image Classification Method Based on YOLOv5. *Journal of Physics: Conference Series*, *2024*(1), 012022. https://doi.org/10.1088/1742- 6596/2024/1/012022

Wei, R., He, N., & Lu, K. (2020). YOLO-mini-tiger: Amur tiger detection. *ICMR 2020 - Proceedings of the 2020 International Conference on Multimedia Retrieval*, 517–524. https://doi.org/10.1145/33722 78.3390710

Yang, S. J., Berndl, M., Ando, D. M., Barch, M., Narayanaswamy, A., Christiansen, E., Hoyer, S., Roat, C., Hung, J., Rueden, C. T., Shankar, A., Finkbeiner, S., & Nelson, P. (2018). *Assessing microscope image focus quality with deep learning*. 1–9.

Zhou, F., & Li, Q. (2014). Parking guidance system based on zigbee and geomagnetic sensor technology. *Proceedings - 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, DCABES 2014*, 268–271. https://doi.org/10.1109/DCABES.2014.58