Imaging and Radiation Research (2023) Volume 6 Issue 2
doi: 10.24294/irr.v6i2.5757

ORIGINAL RESEARCH ARTICLE

Performance evaluation of YOLOv5 and YOLOv8 models in car detection

Fatma Nur Kılıçkaya[1,*], Murat Taşyürek[2], Celal Öztürk[3]

[1] *Department of Computer Technologies-Computer Programming, Technical Sciences Vocational School, University of Kayseri, Kayseri 38280, Turkey*

[2] *Department of Computer Engineering, University of Kayseri, Kayseri 38280, Turkey*

[3] *Department of Software Engineering, University of Erciyes, Kayseri 38039, Turkey*

*** Corresponding author:** Fatma Nur Kılıçkaya, fnurkilickaya@kayseri.edu.tr

ABSTRACT

Vehicle detection stands out as a rapidly developing technology today and is further strengthened by deep learning algorithms. This technology is critical in traffic management, automated driving systems, security, urban planning, environmental impacts, transportation, and emergency response applications. Vehicle detection, which is used in many application areas such as monitoring traffic flow, assessing density, increasing security, and vehicle detection in automatic driving systems, makes an effective contribution to a wide range of areas, from urban planning to security measures. Moreover, the integration of this technology represents an important step for the development of smart cities and sustainable urban life. Deep learning models, especially algorithms such as You Only Look Once version 5 (YOLOv5) and You Only Look Once version 8 (YOLOv8), show effective vehicle detection results with satellite image data. According to the comparisons, the precision and recall values of the YOLOv5 model are 1.63% and 2.49% higher, respectively, than the YOLOv8 model. The reason for this difference is that the YOLOv8 model makes more sensitive vehicle detection than the YOLOv5. In the comparison based on the F1 score, the F1 score of YOLOv5 was measured as 0.958, while the F1 score of YOLOv8 was measured as 0.938. Ignoring sensitivity amounts, the increase in F1 score of YOLOv8 compared to YOLOv5 was found to be 0.06%.

Keywords: large volume satellite image; object detection; YOLOv5; YOLOv8

ARTICLE INFO

Received: 11 April 2024
Accepted: 24 May 2024
Available online: 1 July 2024

1. Introduction

Vehicle detection is a critical technology that finds many important applications today. It is getting stronger day by day with this technology and deep learning algorithms, and it is expanding its application area in many sectors. With the influence of technological advances in recent years, vehicle detection has an important place in traffic management, automatic driving systems, security, urban planning, environmental impacts, transportation, and emergency response applications. In this context, it is primarily used to monitor traffic flow, evaluate traffic density, and increase security. This information can be integrated into cities' traffic management systems through traffic signals and security cameras. Bautista et al evaluated the performance of a CNN model in detecting and classifying vehicle entries captured by low-quality traffic cameras. Although typical models typically consider high-resolution inputs that provide some useful pattern information, applying low-resolution inputs for such purposes also performs effectively[1,2]. In addition, it plays an important role in automatic driving systems (ADAS). These systems detect other vehicles around the vehicles and provide drivers with safety and comfort features such as collision prevention, lane following, and automatic parking. Tsai et al.[3] observed the emergence of the Advanced Driver Assistance System (ASDS), an application for driver guidance that resulted from the evolution of image sensors within the Driver Assistance System (SDS). A Collision Warning System (CUS) has been proposed to prevent vehicle collisions. For these applications to work, vehicles on the road must be detected[1]. It is also used for vehicle detection, security, and monitoring purposes in many areas such as public areas, parking lots, and private properties. This information can be used to detect potential threats and implement security measures. Javed et al.[4] proposed that future smart cities can harness several technological advancements including deep learning, the Internet of Things (IoT), Industry 5.0, 6G networks, robotic systems, and cyber security integration to address the challenges posed by sustainable urban development, population growth, and citizen demands. They have created a safe and sustainable standard of living. Additionally, it can be used to assess environmental impacts. For example, analyzing vehicle traffic in a particular area can help assess air quality and noise levels. Badach et al.[5] developed a method aimed at enhancing airflow in city design, managing pollution distribution, and reducing people's exposure to air pollution. This method has been implemented in the cities of Antwerp, Belgium, and Gdansk, Poland. Its objective is to determine urban air quality management zones using geographical information systems. It is also used to understand traffic flow and improve urban transportation infrastructure. This information can help urban planners optimize infrastructure and create solutions to traffic problems.

Kashyap et al.[6] emphasized the importance of traffic flow prediction within specific regions at predetermined time intervals, advocating for precise forecasting of this traffic flow. In their study, conducted in the context where traditional models are no longer adequate, they particularly focused on deep models, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), Restricted Boltzmann Machines (RBM), and Stacked Autoencoder (SAE). They addressed this technological evolution by examining learning architectures. In this context, they investigated how vehicle detection can contribute to making transportation systems more efficient and seamless by monitoring and analyzing traffic flow. This can reduce journey times and optimize energy consumption. Finally, detecting vehicles in traffic can help emergency responders get to the scene quickly. It is important to provide a faster response in case of an accident or emergency. On the other hand, recent developments in remote sensing technologies have offered new research opportunities, especially in the field of analysis and interpretation of satellite images. The main difficulty is the small size and large number of objects in satellite images[7]. This can cause information loss during convolution operations and make it difficult to distinguish features due to the viewing angle. Particular difficulties arise due to the different resolution levels for satellite images[7]. To overcome such difficulties, large volumes of satellite images are published in WMS format with GeoServer software, which has an open-source strategy that enables spatial data to be presented in geographic information systems. Then, satellite images presented in WMS format produced by GeoServer with the TileCache system, developed in Python and licensed under BSD, are saved in tile format by creating small images of 256 x 256 dimensions. It is emphasized that vehicle detection is made with YOLOv5 and YOLOv8 technologies using the data obtained from these satellite images recorded in Tile format. YOLO (You Only Look Once) algorithms are deep learning methods that can detect objects in a single pass and exhibit high-speed performance. A comparison of YOLOv5 and YOLOv8 reveals the advantages of both algorithms in terms of vehicle detection performance in aerial images. This comparison aims to evaluate the strengths of both versions and the features that may be more effective in certain scenarios.

2. Related works

Recent developments in remote sensing technologies have offered new research opportunities, especially in the field of analysis and interpretation of satellite images[8]. Various techniques have been proposed in the literature for vehicle detection and similar object detection in satellite images. In this work, the latest and relevant work in the field of object detection for satellite images is presented. Object detection or counting at the location obtained from a specific satellite image has become a highly demanded task that has become widespread with the innovations of technology in recent years[9]. A typical practice, as reported in the study of Liao et al.[9], is to estimate a retailer's revenue by counting vehicles in parking lots daily or weekly. Since the number of cars in the parking lot correlates with the retailer's revenue, it is possible to accurately detect and count vehicles in parking lots on a global scale and combine this data with other sources to predict the retailer's profits before the annual/quarterly report is published. In addition, various AI applications, such as estimating retailer profits using Komissarov[10], satellite imagery, and predicting agricultural yields and prices, provide valuable tools for monitoring economic activity, estimating energy resources, and identifying poverty. They also estimated public health by tracking timely changes in the number of vehicles parked at hospitals[11]. Another study found at the Federal Polytechnic Oko in Nigeria, demand for inadequate parking spaces is a common problem in university campuses, polytechnics, monotechnics, and colleges of education[12]. These studies are made from high-resolution satellite images. Special difficulties may arise due to different resolution levels for satellite images[7]. The bird's eye perspective of satellite photographs causes inherent differences in the scale and orientation of objects. Moreover, the background of satellite images is complex and the object area is small; therefore, small objects tend to be missing due to the difficulty of feature extraction. Overlays and aliasing of dense objects also affect detection performance. Although the self-sufficient attention mechanism has been introduced to detect small objects, the computational complexity has increased with the resolution of the image[13]. To overcome such difficulties, large volumes of satellite images are published in WMS format with GeoServer software, which has an open-source strategy that enables spatial data to be presented in geographic information systems. Then, satellite images presented in WMS format produced by GeoServer with the TileCache system, developed in Python and licensed under BSD, are saved in tile format by creating small images of 256 × 256 dimensions. It is emphasized that vehicle detection is made with YOLOv5 and YOLOv8 technologies using the data obtained from these satellite images recorded in Tile format. YOLO algorithms are deep learning methods that can detect objects in a single pass and exhibit high-speed performance. In recent years, developments in remote sensing technologies and the design of satellite sensors have led to the emergence of new research topics in the field of remote sensing.

Patel and Meduri[14] aimed to develop and test a CNN model for detecting people and vehicles using the YOLO framework. It provides a method that helps with automatic parking area detection. The method produces detection results that require no human intervention but are verifiable by humans and can significantly reduce the time spent tagging thousands of parking spaces. This approach can achieve an 86% reduction in human effort. Additionally, accuracy can be further improved with a higher frame rate and vehicle tracking algorithm improvements. In the future, studies can be conducted on combining this method with traditional low-level thresholding techniques, thus increasing the overall performance. In conclusion, this work provides a scientific contribution to the process of detecting parking spaces and has the highest performance, especially in a busy parking area, which can achieve up to a 90% reduction in human effort. Ammar et al.[7] discussed the comparison of Faster R-CNN, YOLOv3, and YOLOv4 methods for vehicle detection on aerial images. For the Stanford dataset; Faster R-CNN achieved a 42% F1 score, YOLOv4 achieved a 32% F1 score and YOLOv3 achieved a 34% F1 score. For the PSU dataset; Faster R-CNN achieved an 84% F1 score, YOLOv4 achieved a 95% F1 score, and YOLOv3 achieved a 96% F1 score. He found that the performance of YOLOv3 deteriorates as the difference in image resolution between training and testing data increases. Putra et al.[15] presented a system that provides real-time person and vehicle detection. This modified YOLO-based system is capable of detecting small objects, offering good detection accuracy and real-time operation. Modifications using 11 × 11 grid cells make it possible to accurately detect small-sized people and vehicles. Arruda et al.[16] emphasized the use of deep learning techniques to address object detection tasks, highlighting that these techniques relied on the training dataset. However, since the training dataset must be similar to the images in the target task, the process of acquiring the dataset often involves labeling the images, which is a time-consuming and costly process. Wang et al.[17] aimed to detect ridesharing vehicles among other vehicles based on the movement traces of these vehicles by utilizing the transfer learning method. CoTrans offering a framework that can be adapted to transfer learning tasks that involve a two-stage structure. In the first stage, using taxi and bus data, a random forest (RF) classifier is learned based on the projection features shared by these vehicle types. This RF classifier is then used to label all candidate vehicles. In the second stage, a convolutional neural network (CNN) classifier is learned to identify ride-sharing vehicles using high-confidence labels. RF and CNN classifiers are developed iteratively through the learning process with the feature set. As a result, ridesharing vehicles are identified in the candidate vehicle pool using the combined results of RF and CNN. Kashyap et al.[18] developed an artificial intelligence study for car brand and model prediction. In this study, BDCNN (Binary Dense Convolutional Neural Network), a model used in the field of Computer Vision, was used. BDCNN is a type of convolutional neural network that has achieved high success, especially in binary classification tasks. Research has shown that Convolutional Neural Network achieves higher accuracy than other networks on automobile datasets, and the main reason for this success is the extraction of binary descriptive features using edge detection before training the CNN. Xu et al.[19] focused on vehicle detection in the field of computer vision, especially through camera images. They stated that while vehicle detection systems can achieve good detection performance, reliable detection is challenging due to factors such as vehicle proximity, changing light conditions, and environmental visibility. To address these problems, a model called CDCDMA (Cross-Domain Car Detection Model with integrated convolutional block Attention mechanism) was introduced, and it was shown that this model increased the cross-domain vehicle detection performance by 40% and surpassed other models in cross-domain vehicle recognition. Although vehicle detection in satellite images has a significant area in the literature, Yolo comparison from high-resolution satellite images is rare. This study introduced a new approach to accurately detecting vehicles and their spatial location in high-resolution satellite images and achieved a high success rate.

3. Methods and material

Within the scope of this study, a comparison of GeoServer, TileCache, and Yolo technologies, which concern different disciplines, was made to detect building objects from large-volume satellite images together with their real locations on the earth. Therefore, in this section, information about GeoServer and TileCache is first presented, and then YOLO techniques are discussed.

3.1. GeoServer

GeoServer is an open-source map server software for geographic information systems (GIS) applications. Being fully compliant with Open Geospatial Consortium (OGC) standards, it provides an ideal platform to access different geospatial data sources and distribute this data through standard OGC services. As a result, it offers users a simple and flexible framework for creating and sharing cartographic representations[20].

3.2. TileCache

TileCache system is used to increase the performance of large data sets in map systems. WMS servers such as GeoServer regenerate relevant fields with each request, which may cause performance loss. The TileCache system converts the data published as WMS into images of predetermined sizes and saves them to disk. These images can be presented quickly using the zoom, latitude, and longitude information of the area requested by the user. The TileCache system provides performance advantages, especially when presenting large raster data sets[21].

3.3. You only look once (YOLO)

The growth of computer vision is crucial for the development of contemporary information technologies[22]. Object recognition is an important issue in the field of computer vision, and various machine and deep learning models are used to achieve higher performance. Two-stage object detection detectors, which were common in the past, have been left behind with the development of single-stage detectors. The Yolo algorithm stands out by showing superior performance compared to similar two-stage algorithms, especially in the field of object detection. The ability to simultaneously predict bounding boxes in the image and classes of objects in these boxes using a single convolutional network is an important feature that distinguishes the Yolo algorithm from other models. The architecture of YOLO is shown in **Figure 1**. This approach, unlike sliding window or region recommendation based models, provides the advantage of seeing and processing the entire image during the training and testing stages. Its simple structure and high processing speed have made the YOLO algorithm a preferred solution, making it a breakthrough in object detection[23].

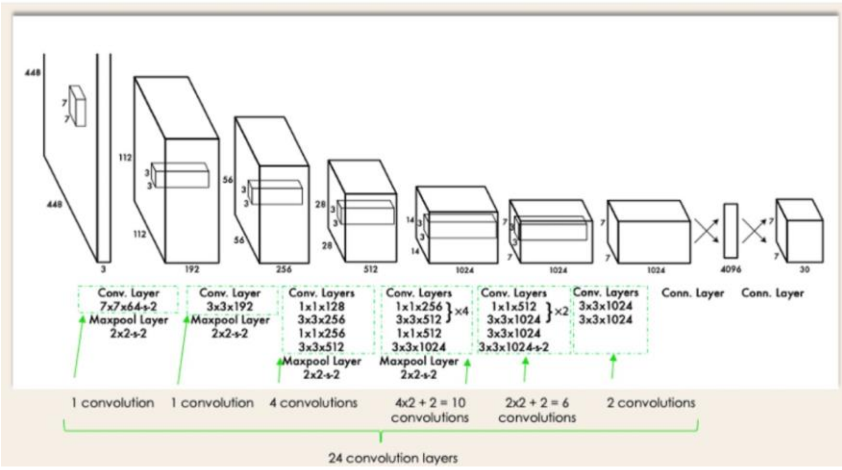

Figure 1. The structure of YOLO[24].

3.3.1. YOLOv5

YOLOv5's model layout has undergone significant changes to its standard architecture. The new layout includes three main components: the backbone, neck, and head. Darknet 53 is used as the main body of YOLOv5. This architecture performs feature extraction by focusing on small filter windows and redundant connections[25,26]. The neck acts as a connecting element between the main body and the head and is designed to collect and refine the features extracted by the main body. This process focuses on the increase of spatial and semantic information at various scales[27]. The head of YOLOv5 consists of three branches that predict features at different scales. Each heading produces bounding boxes, class probabilities, and confidence scores. Additionally, the network uses the Non-maximum Suppression (NMS) algorithm to filter out overlapping bounding boxes[27].

The architecture of YOLOv5 is as shown in **Figure 2**. It includes modular structures such as the main body and neck, CBL module, and Bottle Neck CSP module. The neck amplifies three feature layers extracted from the main body, transmits semantic information from higher layers to the lower layer, and combines with the localization information from the lower layer, performing bottom-up feature fusion; This process creates a double feature pyramid. YOLOv5 creates a Combined Head (CH) using a classifier and a regressor and simultaneously predicts classes, object presence, and connectors through dimensional convolution layers[28].

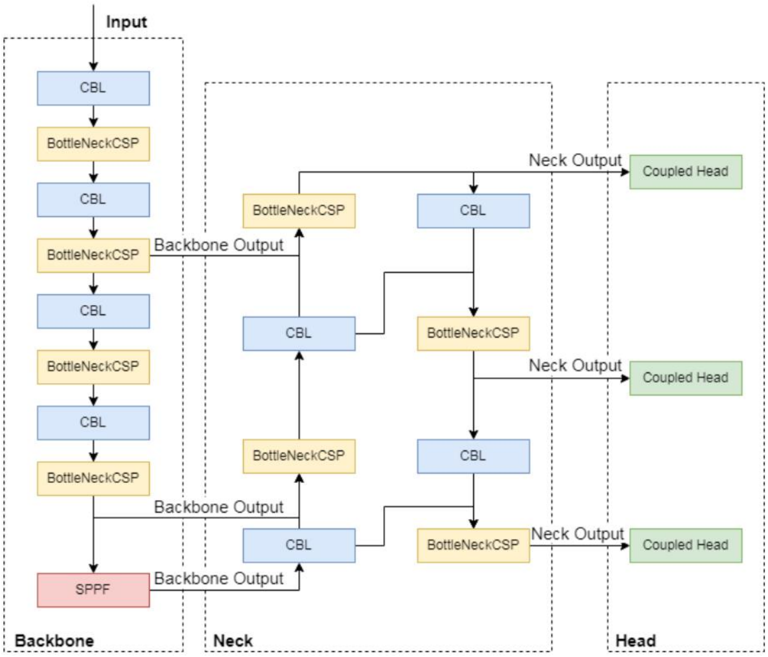

Figure 2. The structure of YOLOv5[27].

3.3.2. YOLOv8

YOLOv8 is the latest model in object detection model architectures that replaces YOLOv5 and includes various improvements. Developed by the company Ultralytics, YOLOv8 differs from YOLOv5 by introducing two new neural networks a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN). It also introduces a new tagging tool to simplify the tagging process. This tool includes useful features like auto-tagging, shortcut tagging, and customizable hotkeys, making tagging images easy[26]. FPN works by increasing the number of feature channels while reducing the spatial resolution of the input image. This allows the creation of a feature map capable of detecting objects at different scales and resolutions. PAN architecture, on the other hand, can combine features from different network levels through hop connections. As a result, the network can more effectively capture features at different scales and resolutions, which is critical for accurately detecting objects of different sizes and shapes. YOLOv8 includes five different versions: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) and YOLOv8x (extra large). Each version offers different scales of parameter counts and computational power to suit a variety of applications.

The architecture of YOLOv8 is as shown in **Figure 3**. YOLOv8 supports various imaging tasks such as object detection, segmentation, pose estimation, tracking, and classification. In the output layer of the model, the sigmoid function is used for the object presence score, and the softmax function is used for class probabilities. YOLOv8 uses CIoU and DFL loss functions for bounding box losses and binary cross-entropy for classification losses. These loss functions improve object detection performance, especially when dealing with small objects[23,25]. YOLOv8 also includes a semantic segmentation model called YOLOv8-Seg. This model has a CSPDarknet53 feature-extracting main body and uses the C2f module instead of the traditional YOLO neck architecture. YOLOv8-Seg maintains high speed and efficiency while achieving superior results across a variety of object detection and semantic segmentation metrics[25].

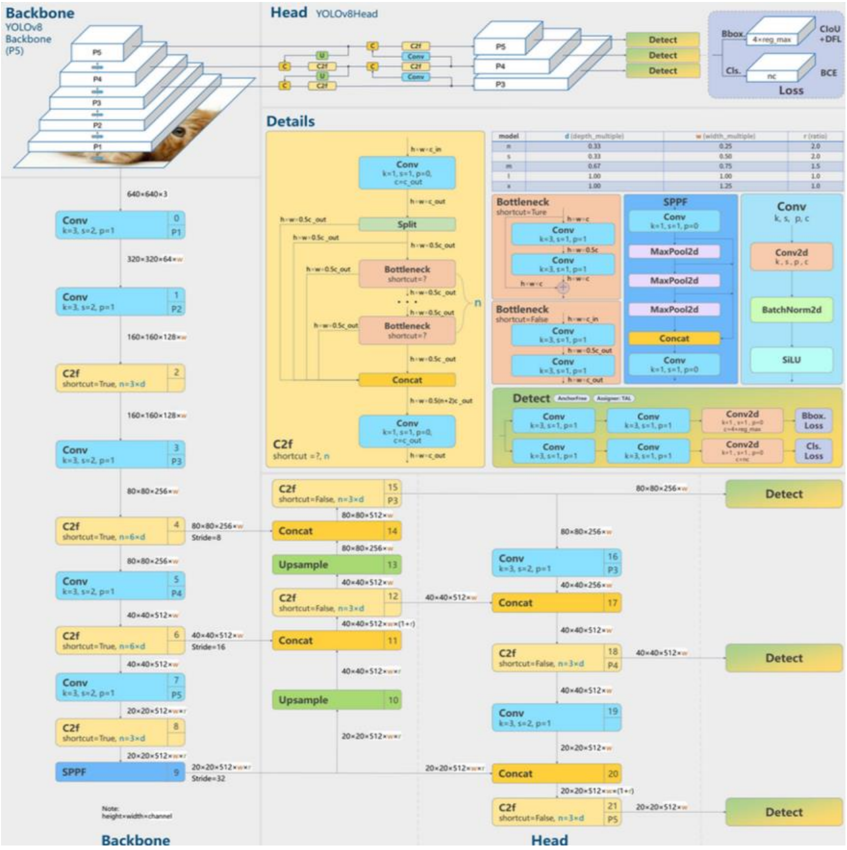

Figure 3. The structure of YOLOv8[29].

3.4. Experimental evaluation

In this research, YOLOv5 and YOLOv8 architectures, which are methods involving deep learning, were used. Deep learning will be applied to vehicle detection in satellite images. This study focuses on comparing the performance of both architectures.

3.4.1. Data set and preprocessing

Within the scope of this research, large-scale satellite images were utilized. A total of 1614 satellite images were used, with 60% of them, equivalent to 1000 images, allocated for labeling as training and validation data. From these, 80% (793 images) were designated for training, while the remaining 20% (207 images) were set aside for validation. The remaining 40%, equivalent to 614 images, were designated as test data for running predictions with the network. Compressed in ECW format, image occupied 20 GB of storage space on computer networks. Subsequently, the configuration details used to convert the ECW view, where GeoServer was integrated, into a tile layout are shown in **Table 1**.

Table 1. TileCache configuration.

Type	**WMS**
URL	http//lacalhost:8082/geoserver/wms?layers=2020uydu_3857
extension	png
size	256 × 256
srs	EPSG: 3857
levels	22
bbox	−20,037,508.34, −20,037,508.34, 20,037,508.34, 20,037,508.34
maxResolution	156,543.0339
extent_type	loose

Table 1 is a list of the layer names that GeoServer supplied. It should now be possible to identify which particular GeoServer layer is being accessed. The term "type" refers to information related to GeoServer's service, wherein GeoServer provides data as a Web Map Service (WMS). This service's data is transformed into pre-sized images via the TileCache system. The server location where the WMS service is hosted is included in the URL. The term "localhost" is often used to refer to the address of the local machine, since GeoServer is usually installed on a local computer. The tiling layout will be saved in the computer's disk system in the format chosen by the extension argument; commonly, jpeg and png formats are used in this regard. While JPEG format is more frequently used for satellite photos with less ideal backdrops, PNG format is recommended, particularly for crisp photographs. JPEG format was therefore chosen for this investigation. The value option establishes the tiled image's dimensions; 256 is the most widely utilized value in this case. One of the most often used projection systems for two-dimensional maps is EPSG:3857, which is identified by the Srs parameter as the projection server for the tiled view. The Web Mercator projection is another name for this projection organization. The cylindrical map projection known as the Web Mercator projection in academic contexts is based on an equator-aligned tangent spheroid that is calculated using formulas. The zoom level at which the tiling process will take place is determined by the Levels option. The variable z's maximum possible value is specified by this argument. To enable universal accessibility for tile pictures in EPSG:3857, standard global restrictions are followed for the bounding box parameter. In the meantime, the highest level of resolution is defined by the max resolution parameter. The srs and bbox parameters' limitations should be compared in order to determine this value. To establish this value as common practice, it is usually used as the default setting. The Extent_type parameter is used when the desired limits are empty of data; setting it to 'loose' guarantees that blank pictures will be created in the event that the WMS source is unable to provide data for the designated limits.

3.4.2. Object detection performance of the methods

Object detection algorithms have an important place in the field of computer vision, and evaluating the performance of various models in this field is a critical step in developing effective and reliable solutions for real-world applications. In this context, the comparison of prominent models such as YOLOv5 and YOLOv8 has become an important research topic. In this study, the F1 score was used to evaluate the object detection abilities of both models.

An overview of the differences in object detection performance between the YOLOv5 and YOLOv8 algorithms is shown in **Table 2**. At first glance, it is clear that YOLOv5 attains a higher accuracy of 0.928, suggesting that the model yields findings that are generally more correct. On the other hand, YOLOv8 does rather well, with an accuracy of 0.894, which is little less than that of YOLOv5. The focus then shifts to measurements related to recall and precision. Recall shows the percentage of true positives found, whereas precision shows the ratio of genuine positives to all positive predictions. The results show that YOLOv5 achieves better recall (0.969) and precision (0.947), indicating fewer missed detections and false positives. Despite YOLOv8's comparatively high recall (0.944) and precision (0.931) results, it is important to recognize the advantage YOLOv5 has in these measurements, as shown in **Figure 4**.

Table 2. YOLOV5 and YOLOV8 model performance.

Approach	Total	TP	FP	FN	TN	Accuracy	Precision	Recall	F1-Score
YOLOv5	1139	1105	139	61	35	0.928	0.947	0.969	0.958
YOLOv8	1154	1086	129	80	64	0.894	0.931	0.944	0.937
YOLOv5 (not FP)	1139	1105	139	0	35	0.972	1	0.969	0.984
YOLOv8 (not FP)	1154	1086	129	0	64	0.949	1	0.944	0.971

TP = true positive; FP = false positive; FN = false negative; TN = true negative; accuracy = TP + TN/TP + TN + FP + FN; precision = TP/TP + FP; recall = TP/TP + FN; F1 score = 2* precision * recall/precision + recall.

An overview of the differences in object detection performance between the YOLOv5 and YOLOv8 algorithms is shown in **Table 2**. At first glance, it is clear that YOLOv5 attains a higher accuracy of 0.928, suggesting that the model yields findings that are generally more correct. On the other hand, YOLOv8 does rather well, with an accuracy of 0.894, which is little less than that of YOLOv5. The focus then shifts to measurements related to recall and precision. Recall shows the percentage of true positives found, whereas precision shows the ratio of genuine positives to all positive predictions. The results show that YOLOv5 achieves better recall (0.969) and precision (0.947), indicating fewer missed detections and false positives. Despite YOLOv8's comparatively high recall (0.944) and precision (0.931) results, it is important to recognize the advantage YOLOv5 has in these measurements, as shown in **Figure 4**.

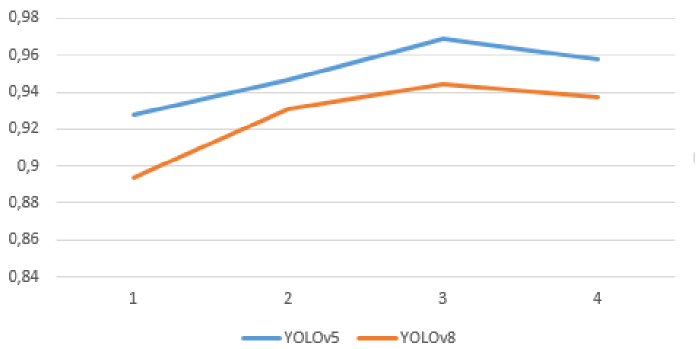

Figure 4. YOLOv5 vs. YOLOv8: Object detection performance comparison.

The performance values of the YOLOv5 and YOLOv8 models are summarized in **Table 2**. The YOLOv5 model is 1.63% greater in precision than YOLOv8. The difference in the recall value of the YOLOv5 model is larger than that of YOLOv8 and is 2.49%. The obtained experimental results offer an interesting perspective when compared with the F1 scores measuring the performance of YOLOv5 and YOLOv8 models. It has been observed that the F1 score of your YOLOv5 model is higher than that of your YOLOv8 model. The F1-score value of the YOLOv5 model is 2.05% larger than that of YOLOv8. In our research, it was determined that the higher sensitivity of YOLOv8 was the source of the difference between F1 scores. For this reason, the precision values of the objects that were not included in each image and those labeled due to sensitivity were recalculated, and these values are given in **Table 2**. We found that when the sensitivity factor was eliminated, the F1 score difference between YOLOv5 and YOLOv8 decreased from 0.020 to 0.013. This analysis shows that YOLOv8 provides greater precision than YOLOv5. YOLO object detection algorithms are generally effective in real-time object detection tasks. The F1 score stands out as a metric that evaluates both the accuracy and precision of the model. A high F1 score reflects the model's ability to successfully detect both true positive predictions and true positive examples. Comparison of YOLOv5 and YOLOv8 models in examining technological approaches in object detection is given in **Table 3**.

Table 3. Comparison of YOLOv5 and YOLOv8 models: Examination of technological approaches in object detection.

	YOLOv5	**YOLOv8**
Phases	Train, Validation, Test	Train, Validation, Test
Neural network type	CNN	CNN
Backbone feature extractor	Darknet	CSPDarknet53
Location detection	Grid-based	Grid-based
Number of anchors boxes	3	3
Default anchors sizes	Various sizes	Various sizes
IoU thresholds	0.5, 0.6, 0.7, 0.8, 0.9	0.5, 0.6, 0.7, 0.8, 0.9
Loss function	YOLOv5 Loss Function	YOLOv8 Loss Function

Table 3. (*Continued*).

	YOLOv5	**YOLOv8**
Input size	Various sizes	Various sizes
Momentum	0.937	0.937
Weight decay	0.0005	0.0005
Batch size	64	64
Precision	0.947	0.931
Recall	0.969	0.944
F1 score	0.958	0.937

The comparison between YOLOv5 and YOLOv8 models showcases advanced technological approaches in the field of object detection. Both models embrace deep learning-based Convolutional Neural Network (CNN) architectures and follow the same training, validation, and testing procedures. They employ a grid-based method for localization and use the same number of anchors, albeit with different default anchor sizes. Additionally, they operate with similar IoU thresholds and share the same weight decay and momentum values. However, they exhibit differences in performance; for instance, YOLOv5 demonstrates higher precision and recall values. This comparison contributes to understanding the decisive factors in the model selection process when evaluating advancements in object detection.

Figure 5. YOLOv5 detection results.

The results of the satellite image training applied to YOLOv5 and YOLOv8 models are given in **Figures 5** and **6**. **Figure 5** shows the detection results of YOLOv5. YOLOv5 was used to detect cars and the scores obtained reflect the high accuracy rates of the model in object recognition.

Figure 6, shows the detection results of YOLOv8. The model was used to detect cars using YOLOv8 and the scores obtained indicate the high accuracy levels of the model in recognizing objects.

Figure 6. YOLOv8 detection results.

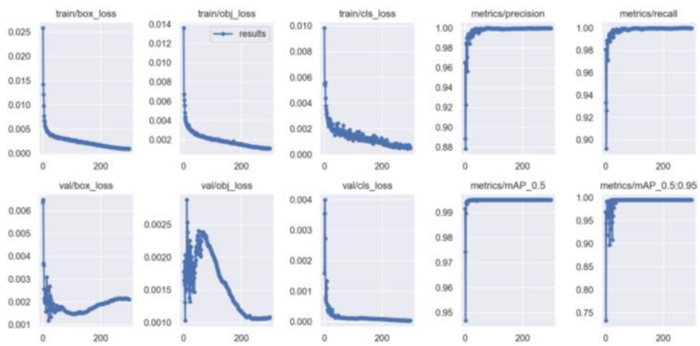

Figure 7. YOLOv5 performance results.

Figure 7, demonstrates that the YOLOv5 model exhibits successful performance during its training process, characterized by decreasing loss values, improving metrics, and a declining learning rate. The analysis focuses on examining the loss values on both the training and validation sets, which serve as indicators of the model's performance. Decreasing loss values signify an improvement in the model's learning process. The graph illustrates a progressive decrease in loss values over time for both training and validation sets, indicating enhanced learning efficiency. In **Figure 7**, following an initial increase in the val/obj_loss value, there is a subsequent decrease over time, indicating an augmentation in the learning rate.

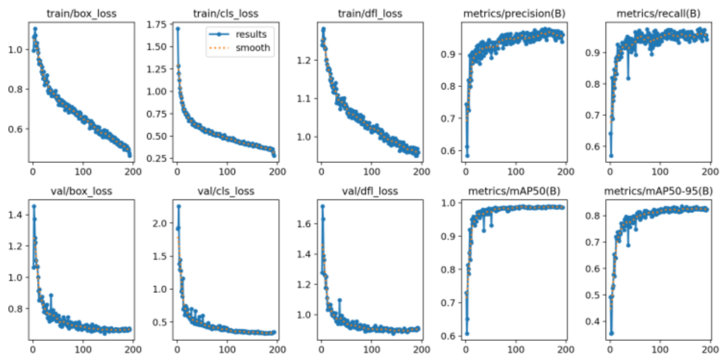

Figure 8. YOLOv8 performance results.

Figure 8 demonstrates that the YOLOv8 model exhibits successful performance during its training process. This is characterized by decreasing loss values, improving metrics, and a declining learning rate. The decreasing loss values compared to YOLOv5 indicate that the model performs the learning process better. In **Figure 8**, the val/obj_loss value shows a consistent decrease, indicating that the learning rate is better compared to YOLOv5. Precision, Recall, and other performance results are shown graphically in **Figures 6** and **8**. It is known that the performance values of YOLOv5 and YOLOv8 models increase until the last epoch in the graphs. When the characteristics of the graphs in **Figures 6** and **8** are examined, it can be seen that the training process does not lead to overfitting and underfitting situations. This shows that the deep learning process occurs smoothly.

4. Results

Based on the research results, YOLOv5 and YOLOv8 models are known to successfully detect cars in satellite images. There are differences in performance values when it comes to car detection. Deep learning models, especially algorithms such as YOLOv5 and YOLOv8, show effective vehicle detection results with satellite image data. According to the comparisons, the precision and recall values of the YOLOv5 model are 1.63% and 2.49% higher, respectively, than the YOLOv8 model. The reason for this difference is that the YOLOv8 model makes more sensitive vehicle detection than the YOLOv5. In the comparison based on the F1 score, the F1 score of YOLOv5 was measured as 0.958, while the F1 score of YOLOv8 was measured as 0.938. Ignoring sensitivity amounts, the increase in the F1 score of YOLOv8 compared to YOLOv5 was found to be 0.06%. In conclusion, while YOLOv5 and YOLOv8 models demonstrate high performance in terms of F1 score, there is a noticeable difference in the precision criterion, favoring YOLOv8. This observation suggests that YOLOv8 achieves object detection with higher precision, leading to a lower rate of detection in the F1 score compared to YOLOv5. The highlights of this study include the wide range of applications of vehicle detection technology, the effective results obtained through the use of deep learning models, and the contributions of this technology in the fields of urban planning, traffic management, and security.

Author contributions

Conceptualization, FNK, MT and CÖ; methodology, FNK, MT and CÖ; software, FNK and MT; validation, FNK and MT; writing FNK, MT and CÖ; supervision, CÖ; project administration, MT; funding acquisition, MT and CÖ. All authors have read and agreed to the published version of the manuscript.

Funding

This study is funded by Project No. 122E315 of the Career Development Program (3501) of Turkey's Scientific and Technological Research Council.

Conflict of interest

The authors declare no conflict of interest.

References

1. Maity S, Bhattacharyya A, Singh PK, et al. Last Decade in Vehicle Detection and Classification: A Comprehensive Survey. Archives of Computational Methods in Engineering. 2022; 29(7): 5259-5296. doi: 10.1007/s11831-022-09764-1
2. Bautista CM, Dy CA, Manalac MI, et al. Convolutional neural network for vehicle detection in low resolution traffic videos. In: Proceedings of the 2016 IEEE Region 10 Symposium (TENSYMP).
3. Tsai YM, Huang KY, Tsai CC, et al. Learning-Based Vehicle Detection Using Up-Scaling Schemes and Predictive Frame Pipeline Structures. In: Proceedings of the 2010 20th International Conference on Pattern Recognition.
4. Javed AR, Shahzad F, Rehman S ur, et al. Future smart cities: requirements, emerging technologies, applications, challenges, and future aspects. Cities. 2022; 129: 103794. doi: 10.1016/j.cities.2022.103794
5. Badach J, Voordeckers D, Nyka L, et al. A framework for Air Quality Management Zones - Useful GIS-based tool

for urban planning: Case studies in Antwerp and Gdańsk. Building and Environment. 2020; 174: 106743. doi: 10.1016/j.buildenv.2020.106743
6. Kashyap AA, Raviraj S, Devarakonda A, et al. Traffic flow prediction models – A review of deep learning techniques. Cogent Engineering. 2021; 9(1). doi: 10.1080/23311916.2021.2010510
7. Ammar A, Koubaa A, Ahmed M, et al. Aerial images processing for car detection using convolutional neural networks: Comparison between faster r-cnn and yolov3. arXiv. 2019; arXiv:1910.07234.
8. ölkesen IC, Yomralıo˘glu T. Use of worldview-2 satellite imagery and ancillary data for mapping land cover and use (Turkish). Harita Dergisi. 2014; 152(2): 12-24.
9. Liao L, Xiao J, Yang Y, et al. High temporal frequency vehicle counting from low-resolution satellite images. ISPRS Journal of Photogrammetry and Remote Sensing. 2023; 198: 45-59. doi: 10.1016/j.isprsjprs.2023.02.006
10. Komissarov V. AI applications for satellite imagery and satellite data. Available online: https://emerj.com/ai-sector-overviews/ai-applications-for-satellite-imagery-and-data/ (accessed on 23 May 2024).
11. Association AH. Social determinants of health series: Transportation and the role of hospitals. Available online: https://www.aha.org/ahahret-guides/2017-11-15-social-determinants-health-series-transportation-and-role-hospitals (accessed on 23 May 2024).
12. Vivian EO. Car park suitability mapping in federal polytechnic Oko, Anambra state, using geospatial techniques. Intemational Journal of Innovative Science and Research Technology. 2023; 8(2).
13. Gong H, Mu T, Li Q, et al. Swin-Transformer-Enabled YOLOv5 with Attention Mechanism for Small Object Detection on Satellite Images. Remote Sensing. 2022; 14(12): 2861. doi: 10.3390/rs14122861
14. Patel R, Meduri P. Car Detection Based Algorithm for Automatic Parking Space Detection. In: Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA).
15. Putra M, Yussof Z, Lim K, Salim S. Convolutional neural network for person and car detection using yolo framework, Journal of Telecommunication, Electronic and Computer Engineering. 2018; 10(1-7): 67-71.
16. Arruda VF, Paixao TM, Berriel RF, et al. Cross-Domain Car Detection Using Unsupervised Image-to-Image Translation: From Day to Night. In: Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN). Published online July 2019. doi: 10.1109/ijcnn.2019.8852008
17. Wang L, Geng X, Ma X, et al. Ridesharing car detection by transfer learning. Artificial Intelligence. 2019; 273: 1-18. doi: 10.1016/j.artint.2018.12.008
18. Kashyap K, Miri R. An Optimal BDCNN ML Architecture for Car Make Model Prediction. Journal of Artificial Intelligence and Technology. Published online August 28, 2023. doi: 10.37965/jait.2023.0268
19. Xu H, Lai S, Li X, et al. Cross-domain car detection model with integrated convolutional block attention mechanism. Image and Vision Computing. 2023; 140: 104834. doi: 10.1016/j.imavis.2023.104834
20. Citra DH, Jazman M, Afdal M, et al. Vector tile server in geographic information system in bapenda pekanbaru city, KLIK: Kajian Ilmiah Informatika dan Komputer. 2023; 3(6): 861-869.
21. Taşyürek M, Türkdamar MU, Öztürk C. DSHFS: a new hybrid approach that detects structures with their spatial location from large volume satellite images using CNN, GeoServer and TileCache. Neural Computing and Applications. 2023; 36(3): 1237-1259. doi: 10.1007/s00521-023-09092-w
22. Herzog DJ, Herzog NJ. Innovative frontiers: Post-quantum perspectives in healthcare and medical imaging. Imaging and Radiation Research. 2024; 6(1): 3852. doi: 10.24294/irr.v6i1.3852
23. Ortataş FN, Kaya M. Performance Evaluation of YOLOv5, YOLOv7, and YOLOv8 Models in Traffic Sign Detection. 2023 8th International Conference on Computer Science and Engineering (UBMK). Published online September 13, 2023. doi: 10.1109/ubmk59864.2023.10286611
24. Science D. Yolo object detection explained. Available online: https://www.datacamp.com/blog/yolo-object-detection-explained (accessed on 23 May 2024).
25. Terven J, Cordova-Esparza D. A comprehensive review of yolo: From yolov1 and beyond. arXiv. 2023; arXiv:2304.00501.
26. Sary IP, Andromeda S, Armin EU. Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images. Ultima Computing: Jurnal Sistem Komputer. 2023; 15(1): 8-13.
27. Reis D, Kupec J, Hong J, Daoudi A. Real-time flying object detection with yolov8. arXiv. 2023; arXiv:2305.09972.
28. Liang H, Chen J, Xie W, et al. Defect detection of injection-molded parts based on improved-YOLOv5. Journal of Physics: Conference Series. 2022; 2390(1): 012049. doi: 10.1088/1742-6596/2390/1/012049
29. King R. Brief summary of YOLOv8 model structure. Available online: https://github.com/ultralytics/ultralytics/issues/189 (accessed on 23 May 2024).