scientific reports

OPEN

Optimized car parts detection with advanced feature fusion and attention modules

Raghuveer Chandaluri[1], Ponduri Vasanthi[2], Lakshmi Prasanna Kothala[1]✉, Y. Chakrapani[3] & Akula Rajesh[3]

Accurate detection of car parts is essential for applications in intelligent transportation systems, automated vehicle inspection, and maintenance planning. However, varying object scales, background clutter, and occlusions still hinder reliable real-time detection. To address these challenges, this paper presents an enhanced YOLO-based architecture that integrates task-specific feature refinement and improved supervision strategies for fine-grained car-part detection. The framework employs a modified C2fCIB block for enriched cross-channel feature interaction and multi-scale representation, an improved PSA module with progressive selective filtering for discriminative spatial–channel attention, and an SPPF layer for efficient multi-receptive field context extraction. In addition, two newly introduced components such as SCDown, a spatial–channel downsampling module designed to retain semantic richness during resolution reduction, and a Dual Assignment Head combining One-to-One and One-to-Many label assignment to further enhance the small-part sensitivity, localization robustness, and recall. Experimental results on a car-parts dataset demonstrate that the proposed model achieves a precision of 63.3%, recall of 81.6%, mAP of 73.7%, and an inference speed of 111 FPS, outperforming baseline detectors including Faster R-CNN, SSD, YOLOv4, YOLOv5, YOLOv7, and YOLOv8. The findings confirm that the proposed architecture delivers an effective balance of accuracy and efficiency, making it suitable for real-world automotive inspection and intelligent vehicle applications.

Keywords Enhanced C2fCIB module, YOLO, Improved PSA, Dual assignment head, Multi-scale feature fusion

The segmentation and identification of car parts play a critical role in automotive industries, particularly in applications such as automated manufacturing, autonomous driving, vehicle inspection, car damage assessment, and safety analysis. Traditional machine learning and handcrafted feature-based methods often struggle in real-world scenarios due to issues such as illumination changes, occlusion, and background clutter, leading to limited robustness and poor generalization. With the advancement of deep learning frameworks particularly those based on convolutional neural networks have significantly improved the ability to localize and classify automotive components in 2D images[1,2]. Among these, the YOLO family of detectors has gained prominence due to its balance of speed and detection accuracy, making it suitable for real-time scenarios. Despite their effectiveness, existing YOLO-based models still face challenges when applied to fine-grained car-part detection, where objects may appear at different scales, partially occluded, visually similar, or tightly packed within complex scenes.

In recent years, researchers have increasingly focused on object detection performance and semantic segmentation using attention mechanisms, multi-scale feature fusion, lightweight backbone strategies. However, several limitations remain, including insufficient cross-channel feature interaction, weak spatial–channel representation during downsampling, and suboptimal label assignment for small or overlapping car parts. These gaps indicate the need for a more robust detection architecture that can efficiently extract multi-level contextual features and adaptively focus on both spatial and channel-specific information. Motivated by these limitations, this work proposes an enhanced YOLO-based detection framework designed specifically for reliable and accurate car-part detection under challenging visual conditions. The framework introduces improved feature extraction, adaptive attention, and refined prediction mechanisms to better handle multi-scale and occluded automotive components.

[1]Vignan's Foundation for Science Technology and Research, Guntur, Andhra Pradesh, India. [2]Eswar College of Engineering, Palnadu, Andhra Pradesh 522237, India. [3]ACE Engineering College, Ankushapur, Telangana 501301, India. ✉email: prasanna.kothala@aceec.ac.in

Literature survey

Early studies on automotive car part segmentation primarily relied on classical image processing techniques. Baird (1977) introduced one of the earliest methods for locating automotive parts on conveyor belts using traditional segmentation, but such approaches were limited by their sensitivity to noise, lighting variations, and non-rigid part shapes[3]. Later, Huang et al. (2013) explored image segmentation through the CAR/CAD joint session though the work remained conceptual without robust automotive datasets[4]. Lu et al. (2014) presented a graphical model based on segment appearance consistency for semantic part parsing of cars, demonstrating success on controlled datasets but facing challenges in real-world complex backgrounds[5]. Patil et al. (2017), Zhang et al. (2018) and Singh et al. (2019) shifted the research focus towards car damage classification using deep learning for insurance claims, but these CNN models lacked fine-grained segmentation for individual car components[6–8]. Dhieb et al. (2019) further used transfer learning for damage location, yet robustness across diverse vehicle types and damage conditions remained limited[9].

With the rise of deep learning, researchers began addressing car part recognition and segmentation using more advanced neural models. Khanal et al. (2020) applied pre-trained deep neural networks for classifying car parts, improving recognition accuracy but failed in tackling segmentation tasks[10]. Pasupa et al. (2021) evaluated U-Net, DeepLabv3+, and SegNet for semantic car part segmentation and found DeepLabv3 + superior, though all models struggled with fine-grained part boundaries and visually similar parts[11]. Lin et al. (2021) proposed automated part segmentation and texture generation using the DeLTA framework, enhancing visual realism but demanding high computational resources[12]. Shaik (2023) proposed a YOLOv9-based model for car parts detection and segmentation with improved multi-scale feature extraction, yet performance on small or occluded parts remained a challenge[13]. Jurado-Rodríguez et al. (2022) extended segmentation in UAV-based images, introducing a new perspective but increasing complexity and processing time[14]. Lin et al. (2022) worked on dataset augmentation for 2D networks car scenes, improving robustness but limited to synthetic and controlled data[15]. Yusuf et al. (2022) adopted Mask R-CNN for real-time vehicle part identification but faced latency issues in high-resolution scenarios[16].

More recent works explored instance segmentation and real-time optimization for automotive applications. ACM (2023) focused on vehicle part identification using instance segmentation to improve labeling accuracy, while Aldawsari et al. (2023) enhanced real-time performance by refining segmentation pipelines for varying conditions such as illumination and camera angle changes[17,18]. Biomedical imaging advancements have also influenced automotive segmentation. Kothala et al. (2023) introduced a Ghost-Convolution-based YOLO model for medical image localization, demonstrating lightweight computational efficiency adaptable to automotive segmentation[19]. Vasanthi and Mohan (2023) proposed a transformer-based detection model for extremely small and dense objects, showcasing the power of attention mechanisms but at the cost of increased model complexity[20]. Anupama et al. (2024) contributed a comparative analysis of deep learning models for car part segmentation, highlighting the need for improved accuracy in fine-scale part boundaries[21].

Further enhancements have been proposed to address multi-scale and small-object detection challenges. Kothala and Guntur (2024) introduced an ensemble learning and test-time augmentation model for localization of small-scale objects, demonstrating robustness that can transfer to automotive part segmentation tasks[22]. Vasanthi and Mohan (2024) developed a Multi-Head-Self-Attention YOLOv5x-Transformer variant to improve multi-scale object detection accuracy, showing potential for car part detection but requiring heavy computational resources[23]. Panboonyuen (2025) proposed ALBERT, a transformer-based architecture for automotive damage and part segmentation, improving contextual reasoning but still lacking integration with lightweight real-time models[24]. VigneshArjunRaj introduced MA-Net, a GitHub-based implementation of a multi-scale attention architecture for car parts and damage segmentation, but it remains experimental with limited large-scale validation[25]. Earlier work by Liu et al. (2016), employing perceptual hashing for segmentation, offered lightweight performance but lacked modern deep learning precision[26]. Dwivedi et al. (2020) reinforced the industrial need for automated car damage assessment using deep learning but did not integrate part segmentation within the framework[27].

Base on the literature we observed the following critical gaps that motivate our study:

1. Poor Detection of Small, Occluded, and Visually Similar Car Parts: Existing models struggle to accurately detect fine-grained components such as indicators, emblems, grills, and door handles, particularly under occlusion, cluttered backgrounds, or low-contrast conditions.
2. Insufficient Multi-Scale Feature Representation for Automotive Parts: Most prior deep learning-based approaches lack effective multi-scale feature fusion tailored for car-part detection, resulting in weak localization performance for parts of varying sizes across different vehicle models.
3. High Model Complexity Affecting Real-Time Deployment: Attention-based and transformer-enhanced object detectors show improved performance but introduce heavy computational load, making them unsuitable for real-time applications.
4. Lack of Detection-Specific Attention Mechanisms Optimized for Automotive Datasets: While attention and transformer modules have been explored in general object detection, there is limited work on lightweight, task-oriented attention mechanisms specifically designed to enhance part-level feature discrimination in automotive environments.

To overcome these gaps, we proposed a novel object detection model. The key contributions are as follows:

1. Car-Part Adaptive Backbone with Modified C2fCIB and SCDown: We redesign the C2fCIB and SCDown modules to better handle fine-grained car-part features. The enhanced C2fCIB introduces lightweight chan-

nel-interaction and feature recalibration, while the SCDown module preserves semantic information during downsampling through spatial–channel selective refinement.

2. Improved PSA for Fine-Grained feature Enhancement: A modified PSA is integrated to sequentially refine spatial and channel attention, enabling the network to focus on subtle and visually similar car-part regions even under occlusion, illumination changes, and background clutter.
3. Dual Assignment Head for Robust Car-Part Detection: We develop a Dual Assignment Head combining One-to-One and One-to-Many label assignment. This hybrid strategy ensures precise and stable localization (O2O) while improving recall for small and occluded parts (O2M), achieving a superior precision–recall balance.
4. Optimized Depth–Width Scaling: The model employs increased depth and width multipliers over the YOLO baseline to enhance feature extraction capacity and representation strength, resulting in better small-part discrimination with efficient training and inference performance.

Proposed model

The proposed architecture provided is a detailed layer-by-layer representation for car part detection is shown in Fig. 1. This model incorporates various modern modules such as Conv, C2f, SCDown, C2fCIB, SPPF, PSA, and v10Detect, each playing a specific role in feature extraction, downsampling, attention, and detection, where input images are processed through multiple convolutional layers for feature extraction before passing to detection heads that predict object classes and bounding boxes. The process begins with an RGB input image represented as I∈$R^{H\times W\times 3}$. Below is an explanation of each component with relevant mathematical formulations and functional roles.

Conv layer

A convolutional (Conv) layer, denoted as Conv ([in_c, out_c, k, s]), is responsible for extracting features through a convolution operation. The output spatial size of this layer is calculated

$$Output\ size = \left[\frac{W-k+2p}{s}\right]+1 \tag{1}$$

Where W = input size, k = kernel size, p = padding, s = stride. The convolution is followed by applying a non-linear activation function, such as LeakyReLU, and then batch normalization (BN) to stabilize learning.

$$Y = BN\left(LeaKyReLU\left(X*W+b\right)\right) \tag{2}$$

C2f: cross stage partial block

This improves gradient flow and feature representation by splitting the input tensor and progressively merging processed features. Let the given input is $X \in\ R^{C\times\ H\times\ W}$. It is divided into two parts, where each F_i typically represents a residual or bottleneck block. The outputs are concatenated as

$$split\ X = [X_1, X_2] \tag{3}$$

$$Y = concat\left(X_1, F_1\left(X_2\right), F_2\left(F_1\left(X_2\right)\right), \ldots\right) \tag{4}$$

Allowing the model to retain lower-level features while still generating higher-level representations. This design reduces computation while preserving important spatial details. For instance, Layer 2: uses three bottleneck modules within the C2f block to achieve this efficient feature extraction and fusion.

SCDown: spatial-channel downsampling module

In the proposed model, we integrate a scaling-aware SCDown module in the backbone to mitigate information loss during down-sampling. Unlike traditional YOLO downsampling (which uses only stride-2 convolution or MaxPool + Conv), SCDown introduces a dual-stage spatial–channel refinement mechanism. First, spatial reduction captures compressed structural patterns, and then the 1 × 1 convolution selectively recalibrates channel responses. This preserves semantic richness, which is not done in standard YOLO downsampling. It reduces the spatial resolution of feature maps while maintaining rich channel information. It typically combines a stride-s convolution for spatial downsampling with a 1 × 1 convolution for channel adjustment. The SCDown operation can be expressed as:

$$Y = Conv_{1\times\ 1}\left(Conv_{k\times\ k,s}\left(X\right)\right) \tag{5}$$

Where $Conv_{k\times k,\ s}$ applies a convolution with kernel size k, stride s (s > 1 for downsampling), and appropriate padding p. The output spatial size is (H' , W'). This module reduces computational load while retaining semantically strong features, making it efficient for deeper layers.

C2fCIB: cross stage partial with channel interaction block

C2fCIB extends the C2f block by adding a Channel Interaction Block (CIB), which applies channel-wise attention to enhance inter-channel dependencies. Here we will process X_2 through n bottleneck modules F_i. Then, we apply channel attention via a squeeze-and-excitation mechanism:

$$S = \sigma\ \left(W_2\cdot\ ReLU\left(W_1\cdot\ GAP\left(Z\right)\right)\right) \tag{6}$$

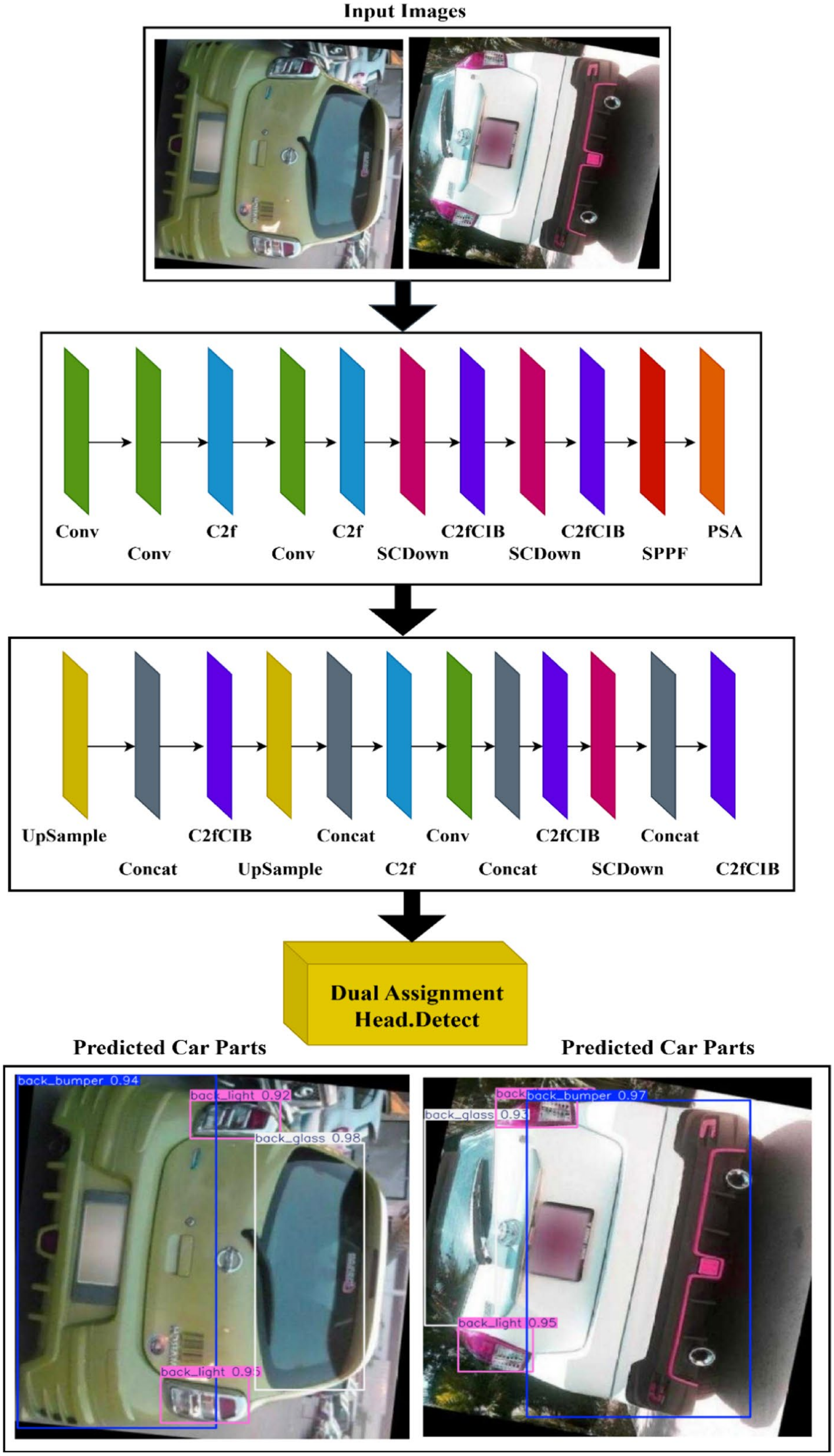


Fig. 1. Proposed model architecture.

$$Y = Z \odot S \tag{7}$$

Where GAP(·) is global average pooling and ⊙ is element-wise multiplication.

This module improves gradient flow and enhances important feature channels for better detection accuracy. Additionally, we emphasize that the proposed implementations of C2fCIB is not direct replications of their original form, but include task-oriented architectural refinements to fine-grained car-part detection. In the enhanced C2fCIB module, we redesigned the cross-interaction bottleneck by integrating a lightweight inter-channel attention mechanism and feature recalibration layer, enabling more discriminative and fine-grained feature aggregation across channels. This adaptation strengthens the module's ability to capture subtle and inter-part dependencies, which are crucial for differentiating visually similar car components.

SPPF: spatial pyramid pooling – fast

SPPF aggregates multi-scale context features using multiple max-pooling operations of different kernel sizes, enabling the model to handle objects at varying scales efficiently. Apply sequential pooling with kernel size k (k = 5, stride = 1) to get $P_1 = MP_k(X)$, $P_2 = MP_k(P_1)$, and $P_3 = MP_k(P_2)$ values. Then to get the final output we will concatenate all.

$$Y = Concat(X, P_1, P_2, P_3) \tag{8}$$

SPPF increases the receptive field without increasing computational complexity, preserving efficiency in real-time detection.

PSA: pyramid split attention module

PSA splits feature maps into multiple groups, processes each group with convolutions of different kernel sizes, and applies attention to weight the importance of each group adaptively.

Compute attention weights:

$$A_i = \frac{exp(GAP(F_i))}{\sum {}_{j=1}^{g} exp(GAP(F_j))} \tag{9}$$

Fuse the outputs.

$$Y = \sum {}_{i=1}^{g} A_i \cdot F_i \tag{10}$$

Deep features (from later layers) carry strong semantic information but lower spatial resolution. Shallow features (from earlier layers) preserve fine spatial details but have weaker semantics. By upsampling the deep features and then concatenating them with shallow features:

$$F_{fused} = Concat(Upsample(F_{deep}), F_{Shallow}) \tag{11}$$

The model merges coarse and fine information. This Feature Pyramid Network (FPN) design helps the detector handle objects of all scales, improving accuracy for both small and large objects. The proposed PSA module will replace the conventional attention flow with a progressive selective filtering strategy, allowing spatial and channel attention to interact sequentially and adaptively. This modification enhances the model's capability to emphasize part-specific salient regions under complex variations such as illumination changes, occlusion, and viewpoint shifts. Collectively, these internal upgrades contribute to a more robust and context-aware detection framework.

Dual assignment head

The Head in an object detection network is the final processing stage that transforms the high-level feature maps from the backbone and neck into meaningful predictions. In the proposed model, we used a novel a Dual Assignment Head, which adopts a hybrid supervision strategy by integrating both One-to-One (O2O) and One-to-Many (O2M) label assignment to significantly enhance car-parts detection performance. The O2O branch assigns each ground truth instance to a single, high-quality prediction through a strict matching mechanism, enabling precise localization[28,29]. This promotes stable optimization, sharper boundary regression, and improved discrimination among visually similar car components. In contrast, the O2M branch allocates multiple positive predictions to the same ground truth instance, thereby enriching gradient propagation and improving recall, especially for small, partially occluded, or low-visibility parts such as indicators, door handles, and emblems. By jointly leveraging the strengths of both assignment strategies, the Dual Assignment Head maintains an optimal balance between precision and recall. Consequently, the fusion of O2O and O2M pathways results in robust part-level detection across diverse vehicle models and challenging automotive imaging conditions involving scale variation, background clutter, and occlusion. For each car part, the proposed head predicts bounding box offsets, Where, (P_x, Py) are offsets relative to anchor boxes. (P_w, P_h) define object size[31]. These offsets are transformed into final coordinates

$$B = P_{x\_coord}, P_{y\_coord}, P_{weight}, P_{height} \tag{12}$$

$$x\_coord = \sigma\ (P_x) + c_x \tag{13}$$

$$y\_coord = \sigma\ (P_y) + c_y \tag{14}$$

$$weight = P_w e^{tw} \tag{15}$$

$$height = P_h e^{th} \tag{16}$$

Experimental outcomes

The car parts detection model was trained and evaluated using a well-structured dataset comprising 3,156 labeled training images with 116 background images and 401 labeled validation images with 12 backgrounds, all free from corruption. The data loading process ensured fast image access, with an average read speed of 1261.7 ± 426.4 MB/s for training images and 1007.0 ± 586.4 MB/s for validation, and new cache files were created to optimize access speed. Data augmentation was applied using Albumentations, including Blur and MedianBlur with a probability of 0.01, grayscale conversion via weighted averaging, and CLAHE (Contrast Limited Adaptive Histogram Equalization) with specified clip limits and tile grid sizes to improve model robustness under varying image conditions. The model was trained for 35 epochs using input image sizes of 640 × 640 for both training and validation, with eight dataloader workers enabling efficient data feeding. The training process selected the AdamW optimizer with a lr of 0.00037 and momentum of 0.9, optimizing 185 weight parameters without decay, 198 with decay, and 197 bias parameters. This setup ensured a comprehensive and efficient training pipeline for high-accuracy car parts detection.

Dataset

The bar chart illustrates the number of detected instances for various car parts in a car parts detection dataset is shown in Fig. 2. The x-axis lists the car part categories such as bumpers, doors, lights, mirrors, hood, trunk, tailgate, and wheels, while the y-axis represents the count of instances for each category. Among all parts, the hood has the highest occurrence with over 2500 instances, followed by front_left_light, front_bumper, front_right_light, and front_door, each with more than 1500 detections. Parts-like back_left_light, back_door, and wheel have comparatively fewer occurrences, under 500 instances. This variation indicates that certain parts, especially those at the front of the vehicle, are more frequently present or annotated in the dataset, which may influence the model's learning bias toward those parts. While the Ultralytics CarParts-Seg dataset is originally a segmentation dataset, we performed a structured conversion of segmentation masks into bounding-box annotations for object detection. Each mask was transformed into a minimum enclosing rectangle to generate precise bounding boxes. The converted labels were saved in YOLO format and validated to maintain annotation consistency. To ensure reliability, 10% of samples were manually cross-checked against the original masks. This approach enabled more accurate localization than conventional bounding-box annotations.

Results and discussion

A mosaic augmented training image used for car parts detection is shown in Fig. 3. Mosaic augmentation combined four images into a single composite image, allowing the model to learn from varied contexts, scales, and object placements in one shot. In this example, multiple car images are stitched together, each containing annotated bounding boxes with labels for specific car parts such as doors, lights, bumpers, mirrors, hood, and wheels. The bounding boxes are color-coded, and each box is associated with a label ID indicating the car part category. This augmentation technique not only increases the diversity of the training dataset but also helps the

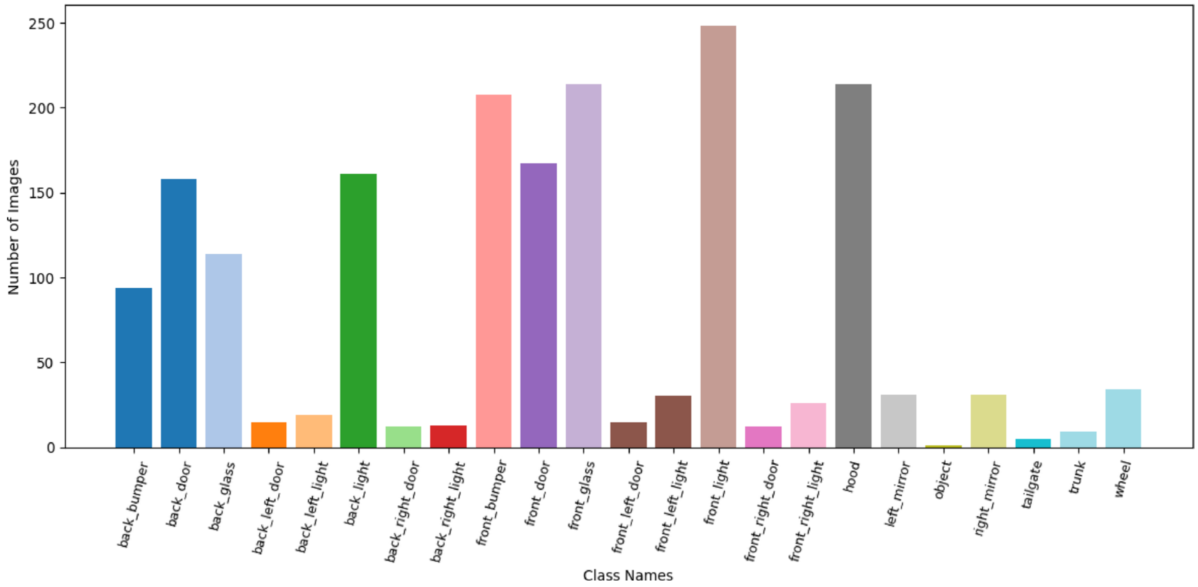


Fig. 2. Bar plot representation of the employed dataset.

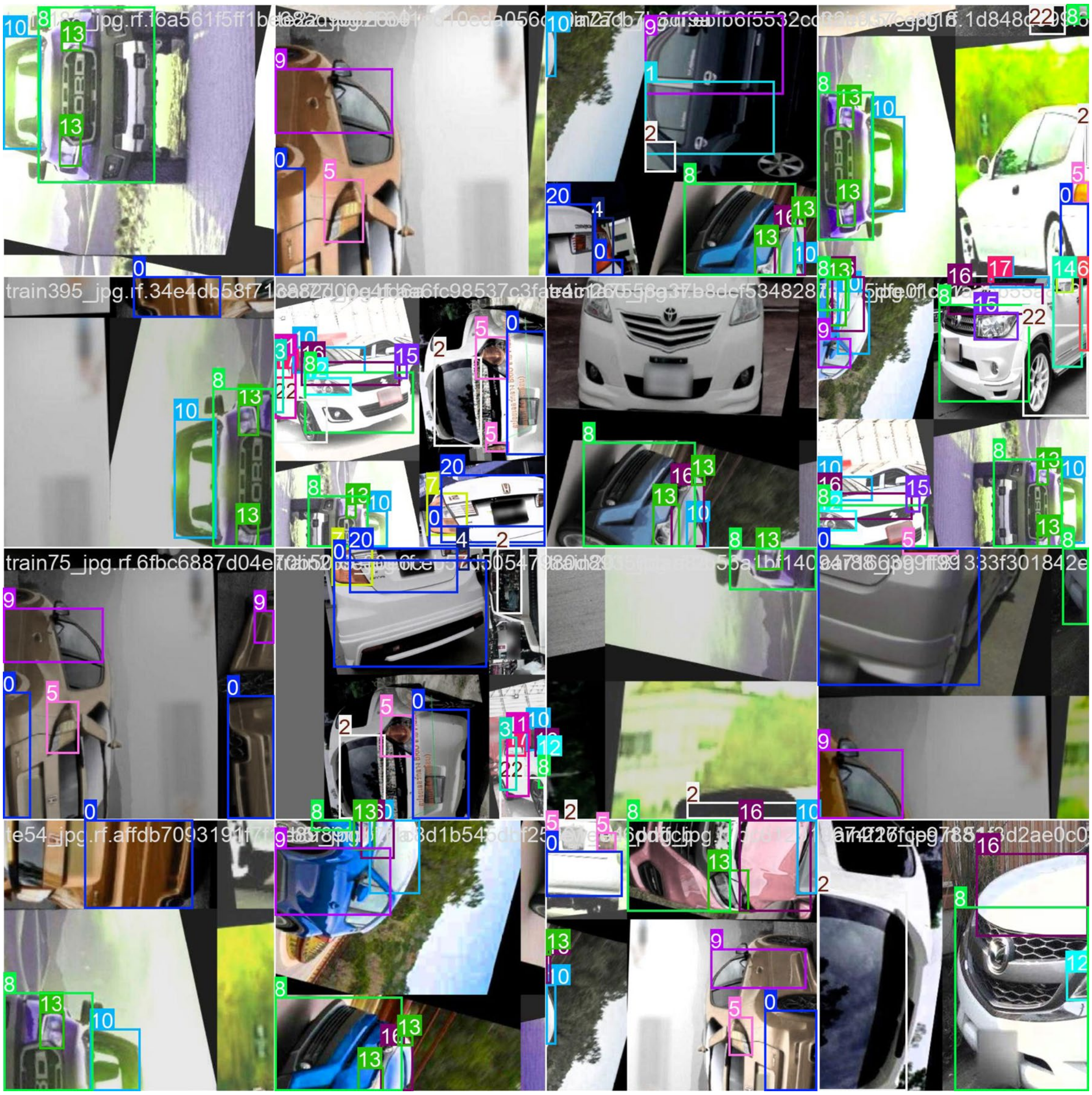


Fig. 3. Mosaic augmented image.

model handle objects appearing in different positions, orientations, and lighting conditions, thereby improving its generalization and detection accuracy.

In Fig. 4, each row represents the predicted labels, while each column corresponds to the actual (true) labels. The diagonal values indicate correct predictions, with darker blue shades showing higher accuracy. For example, classes like front_bumper (0.97), hood (0.96), front_door (0.94), and back_glass (0.91) show high prediction accuracy. However, some parts, such as front_left_door and front_right_light, exhibit noticeable misclassifications, with values spread across other categories, indicating the model sometimes confuses visually similar parts (e.g., different doors or lights). The "background" class also shows some false positives where car parts are mistakenly predicted as background. Overall, the confusion matrix reveals that while the model performs well on distinct parts, it struggles with differentiating between parts that have similar shapes, positions, or visual features.

The training and validation performance curves for a car parts detection model over multiple epochs are shown in Fig. 5. The precision and recall metrics also improve across epochs, with precision reaching above 0.80 and recall peaking near 0.83, suggesting the model is correctly identifying most objects with relatively few false positives. The second row displays validation performance, where val/box_loss, val/cls_loss, and val/dfl_loss similarly decrease, showing good generalization to unseen data. The mAP@50 and mAP@50–95 metrics, which measure overall detection accuracy, improve consistently, stabilizing above 0.70 and 0.60 respectively, indicating strong detection capability across IoU thresholds. Overall, the curves suggest that the model is converging well, with both training and validation metrics improving in parallel, reflecting effective learning without major signs of overfitting.

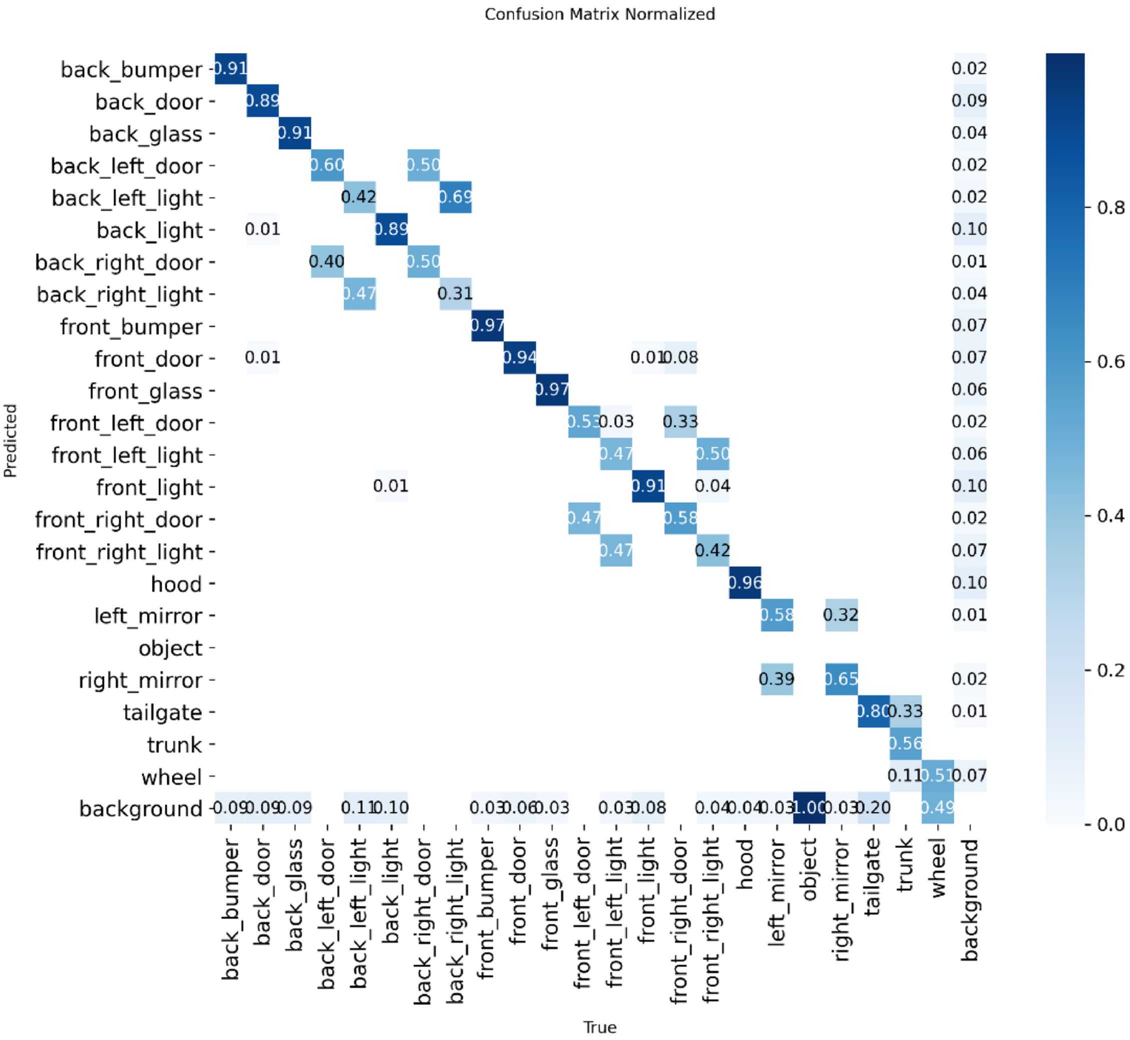


Fig. 4. Confusion matrix.

For the car parts detection task, the proposed model demonstrates strong and balanced performance across precision, recall, mAP@0.5, and F1-score metrics over varying confidence thresholds as shown in Fig. 6. At lower confidence scores, recall remains high, ensuring most car parts are detected, while precision gradually improves as the confidence increases, indicating fewer false positives. The mAP@0.5 curve remains consistently high, reflecting the model's ability to localize and classify car parts accurately across multiple categories such as headlights, bumpers, wheels, and mirrors. The F1-score curve identifies the optimal trade-off point between precision and recall, ensuring reliable detection without sacrificing accuracy.

These trends confirm that the model is well-suited for real-world automotive applications, where both detection completeness and accuracy are critical for tasks such as automated inspection, damage assessment, and inventory management.

Table 1 presents a detailed comparison of seven different object detection methods—Faster-RCNN, SSD, YOLOv4, YOLOv5, YOLOv7, YOLOv8, and a proposed model across four evaluation metrics: precision (%), recall (%), mAP (%), and FPS. precision measures how many of the detected objects are correct. Here, YOLOv7 achieves the highest precision at 68.3%, meaning it produces fewer false positives compared to others. The proposed model has 63.3% precision, which is competitive and slightly better than Faster-RCNN (63.1%), SSD (62.5%), YOLOv4 (61.5%), and YOLOv5 (60.7%), but lower than YOLOv7 and YOLOv8 (66.1%). recall measures how many of the actual objects are detected, reflecting the ability to avoid false negatives. The proposed model stands out here with 81.6% recall, Far surpassing all other methods the second highest being YOLOv8 at 68.3%. This indicates that the proposed model is much better at capturing all relevant objects, making it highly effective for comprehensive detection.

The proposed model again leads with 73.7%, outperforming YOLOv8 (71.7%) and YOLOv7 (68%). This means it consistently performs well across various overlap criteria between predicted and ground truth boxes. FPS represents the processing speed. YOLOv8 is by far the fastest at 126 FPS, followed by YOLOv5 (70 FPS) and YOLOv7 (48 FPS). The proposed model runs at 111 FPS, which is slower than most YOLO variants but still significantly faster than Faster-RCNN (18 FPS) and only slightly behind SSD (26 FPS) and YOLOv4 (31 FPS).

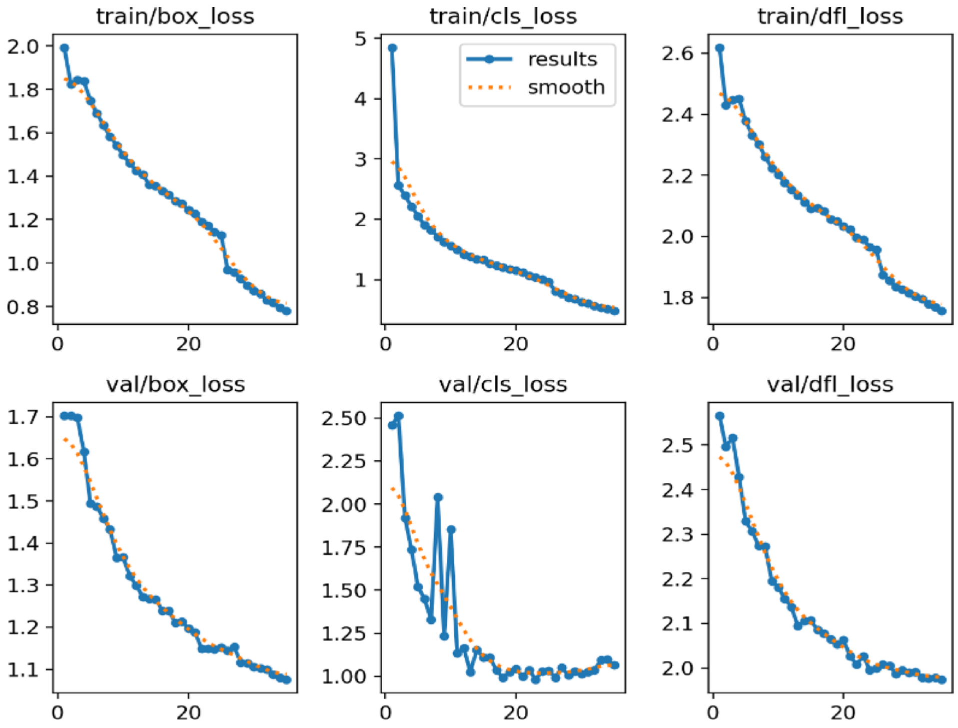


Fig. 5. Performance metrics and loss curves.

This speed is adequate for near real-time applications, though not as optimized for ultra-high-speed scenarios. In summary, while YOLOv8 dominates in raw speed and YOLOv7 excels in precision, the proposed model offers a remarkable trade-off delivering the highest recall and mAP, strong precision, and moderate speed making it ideal for applications where detection completeness and accuracy are more critical than maximum throughput.

In addition to object detection models, several instance segmentation and two-stage detection models, including Mask R-CNN, GCNet, PANet, CBNet, HTC, SipMask, SipMask++, and YOLACT are also shown in Table 1. Among the two-stage architectures, HTC with ResNet-101 demonstrates the strongest performance, achieving 61.3% Precision, 66.8% Recall, 63.9% F1-Score, and 54.3% mAP, outperforming Mask R-CNN, GCNet, PANet, and CBNet variants. SipMask + + and SipMask show noticeable improvements over their predecessors, with SipMask + + achieving the highest performance between them (63.6% Precision, 66.1% Recall, 64.8% F1-Score, and 51.1% mAP). Among the newly added methods, YOLACT reports the highest mAP of 61.3%, demonstrating its effectiveness within the instance segmentation-based category, although its Recall is slightly lower at 62.3%. Overall, HTC-based models deliver the most balanced results among two-stage frameworks, while SipMask + + and YOLACT show competitive performance within lightweight and real-time segmentation-based methods.

Table 2 presents the object detection performance of various vehicle components across four key metrics: Precision (P), Recall (R), mean Average Precision at IoU 0.5 (mAP@50), and mean Average Precision across IoU 0.5–0.95 (mAP@50–95). Overall, components such as the back_bumper (P: 90.6, R: 92.0, mAP@50: 96.4), front_bumper (P: 89.8, R: 97.1, mAP@50: 97.0), and front_glass (P: 89.2, R: 97.2, mAP@50: 96.1) show exceptionally high accuracy and detection quality. In contrast, smaller or less distinctive parts like the tailgate (P: 41.3, R: 80.0, mAP@50: 57.0), front_right_light (P: 42.0, R: 80.8, mAP@50: 54.7), and back_right_door (P: 43.0, R: 83.3, mAP@50: 59.8) display lower precision and mAP scores, indicating higher false positives and reduced localization accuracy. Notably, back_right_light achieves perfect recall (100.0) but only moderate precision (47.1), suggesting it detects all instances but with some incorrect identifications. Overall, large, and visually distinct components tend to have the strongest performance across all metrics, while smaller, less prominent parts show more variability. In the proposed model design, we used the concept of parameter scaling to improve accuracy. Table 3 shows the comparison of Baseline and proposed model parameter scaling factor.

Modified scaling parameters:

- The proposed model adopts increased depth and width multipliers compared to the YOLO baseline, allowing deeper feature extraction and stronger representational power without compromising efficiency.
- The modified C2fCIB, PSA, and SPPF modules are systematically integrated within the backbone and neck, improving feature diversity, gradient stability, and local–global context fusion, which directly contribute to the performance gains.

In Fig. 7., the first diagram represents the ground truth annotations for car parts detection, where each car image is labeled with colored bounding boxes indicating the exact positions and classes of parts such as back_light,

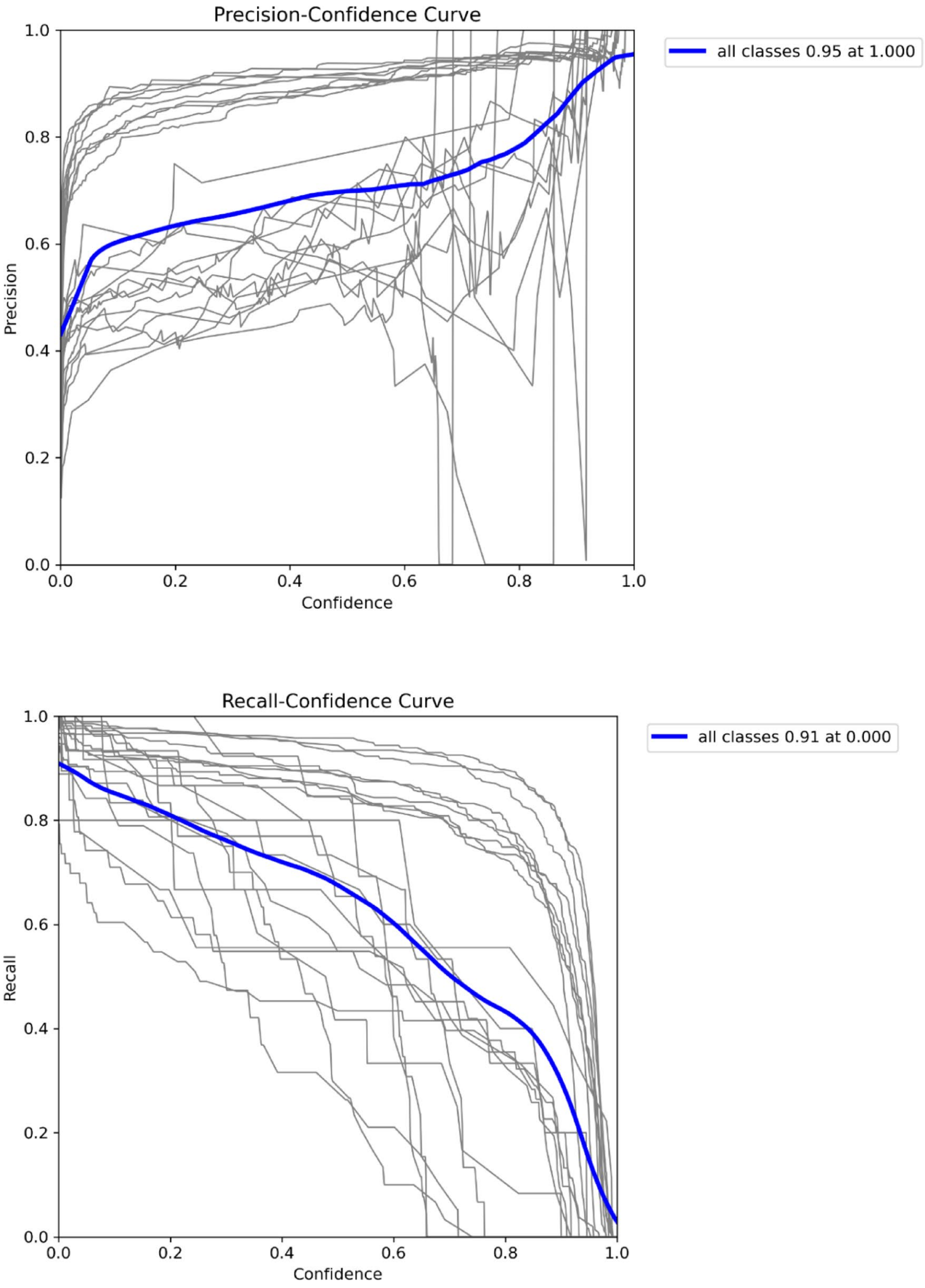
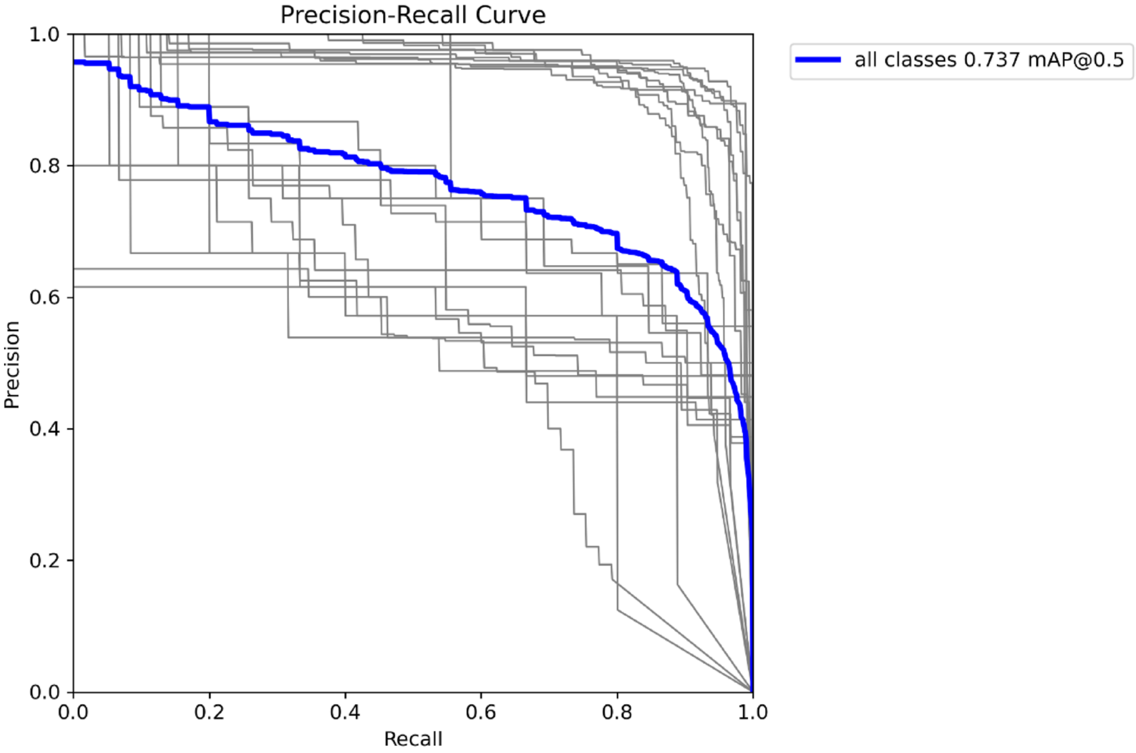


Fig. 6. Precision, recall, map@0.5, and F1 curves of the proposed model w.r.t confidence scores.

back_door, front_door, front_light, back_bumper, back_glass, front_bumper, hood, and front_glass. These serve as the reference for training and evaluation. The second diagram shows the predicted output from the trained model on the same images, where each bounding box includes the predicted class label along with a confidence score ranging from 0 to 1. For instance, back_light is predicted with high confidence scores of 0.8, 0.9, and 1.0, front_light with 0.8 and 0.9, and back_bumper with 0.7 and 0.9. Some predictions like back_glass also have strong scores around 0.9–1.0, whereas others like hood have lower confidence (0.4), indicating less certainty.



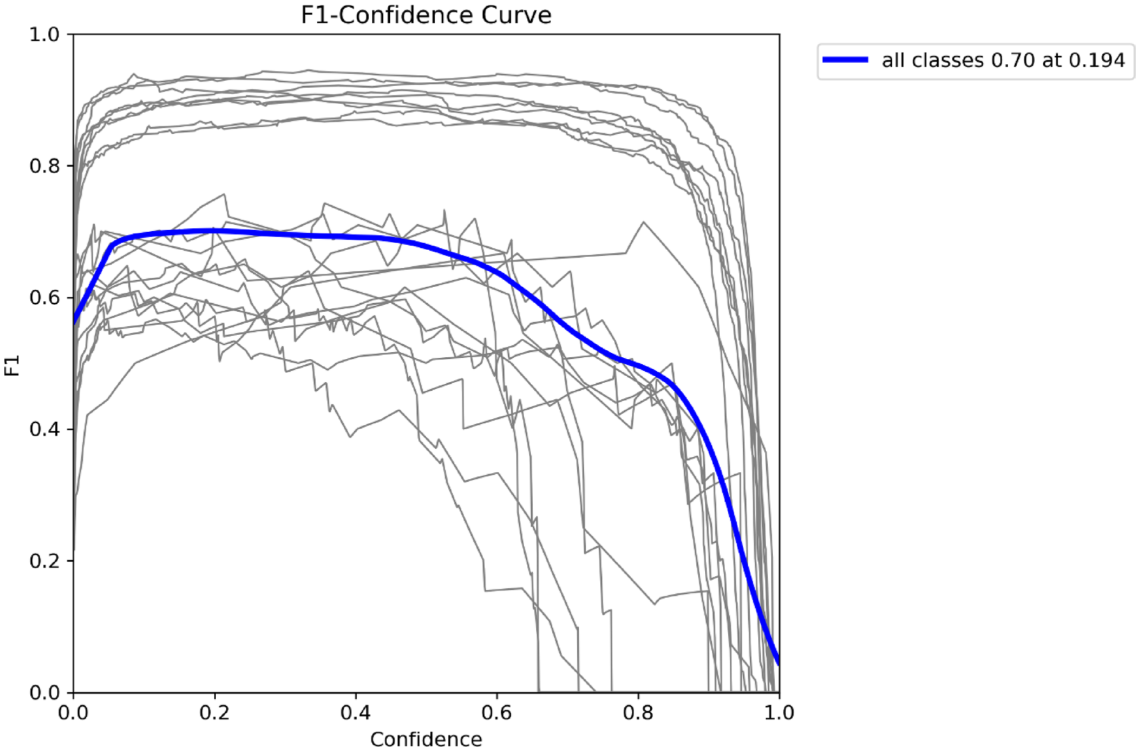


Fig. 6. (continued)

Overall, most predicted parts closely match the ground truth in both location and classification, with high confidence values showing strong detection performance, though lower scores in some categories suggest areas where the model could improve in certainty and robustness.

Method	Precision (%)	Recall (%)	F1 Score (%)	mAP (%)	FPS (ms)
"Mask RCNN_Resnet50"[11]	58.4	63.1	60.6	50.4	--
"Mask RCNN_ Resnet101"[11]	59.2	64.3	61.6	50.8	--
"GCNet_ Resnet50"[11]	57.3	62.7	59.9	50.9	--
"GCNet_ Resnet101"[11]	58.8	61.3	60.0	48.5	--
"PANet_ Resnet50"[11]	56.7	61.1	58.8	48.8	--
"PANet_ Resnet101"[11]	57.9	62.4	60.0	49.6	--
"CBNet_ Resnet50"[11]	59.6	63.8	61.6	51.9	--
"CBNet_ Resnet101"[11]	60.4	62.1	61.2	49.5	--
"HTC_Resnet50"[11]	60.7	66.3	63.4	54.1	--
"HTC_Resnet101"[11]	61.3	66.8	63.9	54.3	--
"SipMask++"[18]	63.6	66.1	64.8	51.1	--
"SipMask"[18]	62.5	64.3	63.4	49.7	--
"YOLACT"[18]	64.1	62.3	63.2	61.3	--
"Faster-RCNN"[30]	63.1	65.3	64.2	67.3	18
"SSD"[30]	62.5	60.3	61.4	62.6	26
"YOLOv4"[30]	61.5	61.6	61.6	62.4	31
"YOLOv5"[30]	60.7	64.5	62.5	62.7	70
"YOLOv7"[30]	68.3	65.8	67.0	68.0	48
"YOLOv8"[30]	66.1	68.3	67.2	71.7	126
"Efficient YOLOv9"[13]	67.5	66.9	67.2	67.7	--
"Proposed model"	63.3	81.6	71.3	73.7	111

Table 1. Performance metrics comparison with state-of-the-art methods.

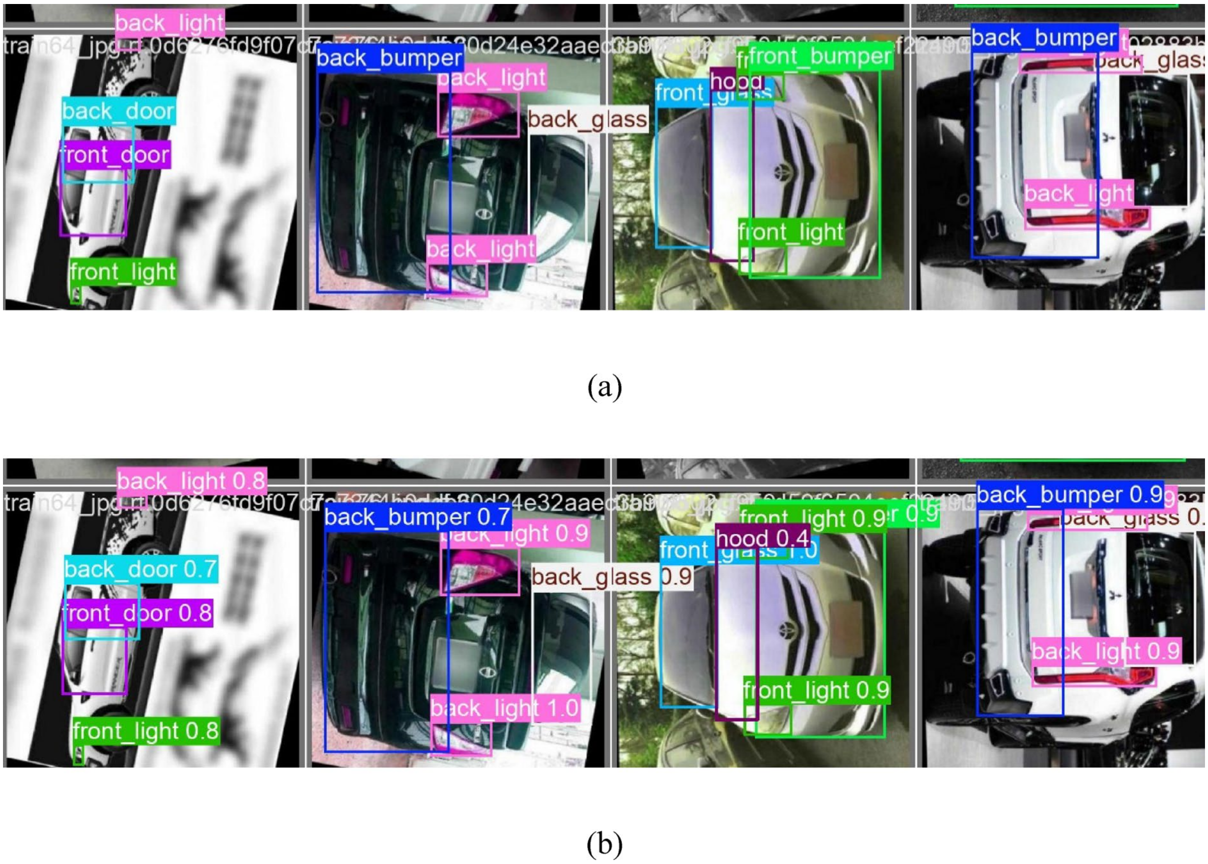
Class	Precision (%)	Recall (%)	*mAP@50* (%)	*mAP@50–95* (%)
back_bumper	90.6	92.0	96.4	85.8
back_door	81.9	91.8	94.1	86.9
back_glass	86.9	92.2	95.6	85.7
back_left_door	55.2	86.7	72.6	67.7
back_left_light	52.7	76.2	59.2	47.3
back_light	83.2	88.5	89.5	71.7
back_right_door	43.0	83.3	59.8	54.8
back_right_light	47.1	100	74.2	62.4
front_bumper	89.8	97.1	97.0	91.4
front_door	85.5	95.0	94.6	89.4
front_glass	89.2	97.2	96.1	92.4
front_left_door	62.0	93.3	77.2	76.7
front_left_light	50.1	90.0	56.0	45.7
front_light	88.5	91.4	91.7	75.1
front_right_door	46.7	80.3	61.9	58.9
front_right_light	42.0	80.8	54.7	48.9
hood	85.5	96.7	94.2	89.7
left_mirror	55.0	63.0	69.9	50.0
right_mirror	61.6	80.6	69.3	48.7
tailgate	41.3	80.0	57.0	43.1
trunk	65.8	66.7	78.0	64.3
wheel	53.4	54.1	55.2	39.7

Table 2. Individual class performance metrics of proposed model.

To strengthen the scientific rigor of the study, we have performed additional experiments evaluating the model's performance on unseen data variations. Specifically, we tested the model on different car models, background environments, and lighting conditions that were not part of the training distribution. These results will help to clarify how well the model maintains detection performance beyond the training dataset and reflect its suitability for real-world deployment.

Model variant	Depth multiple	Width multiple	Params (M)	GFLOPs
YOLO (Baseline)	0.33	0.25	3.2	4.3
Proposed model	1.33	1.25	74.6	154.8

Table 3. Comparison of model width and depth between baseline YOLO and the proposed Framework.



(a)

(b)

Fig. 7. (**a**) Ground truth images, (**b**) Predicted images.

The qualitative results in Fig. 8 clearly demonstrate the enhanced discriminative capability of the proposed model in part-level car component detection across diverse visual conditions. The integration of enhanced C2fCIB and PSA significantly strengthens the feature encoding capacity, enabling precise localization and consistent confidence scores even under scale variation, illumination changes, and background clutter. The proposed model eliminated the redundant background information effectively and preserved high-frequency structural cues, resulting in tightly aligned bounding boxes with minimal spatial deviation from the ground truth. Complex scenarios involving reflections, partial occlusions, and multi-instance overlapping parts were handled robustly, indicating strong contextual and inter-part data. Only a few challenging samples showed minor missed detections under extreme viewpoints or heavily occluded regions. These qualitative findings validate the model's improved representational richness and inference reliability for real-world deployment in automated vehicle inspection applications.

Despite the improved performance, the proposed model exhibits certain limitations. The detection accuracy declines in scenarios involving extreme occlusion, severe motion blur, and highly reflective car surfaces, where feature ambiguity reduces the model's confidence. Although PSA and C2fCIB enhance feature discrimination, the model still struggles with very small parts that occupy < 1% of the image area, indicating a need for finer multi-scale aggregation. The proposed architecture requires moderately higher computational resources, which may constrain deployment on ultra-low-power edge devices. Furthermore, the model's performance shows slight degradation when evaluated on non-trained car brands and uncommon part shapes.

A systematic ablation study was performed by incrementally integrating the C2fCIB, SPPF, PSA, SCDown, and Dual Assignment Head into the baseline YOLO model. The baseline configuration achieved an F1-Score of 65.4% and mAP@50 of 65.2%. Adding the C2fCIB block increased the F1-Score to 67.9% due to enhanced cross-channel feature interaction, while the SPPF module further improved mAP@50 to 69.7% by efficiently capturing multi-scale spatial context. The PSA module gave the most notable gain, achieving 69.6% F1-Score and 72.1%

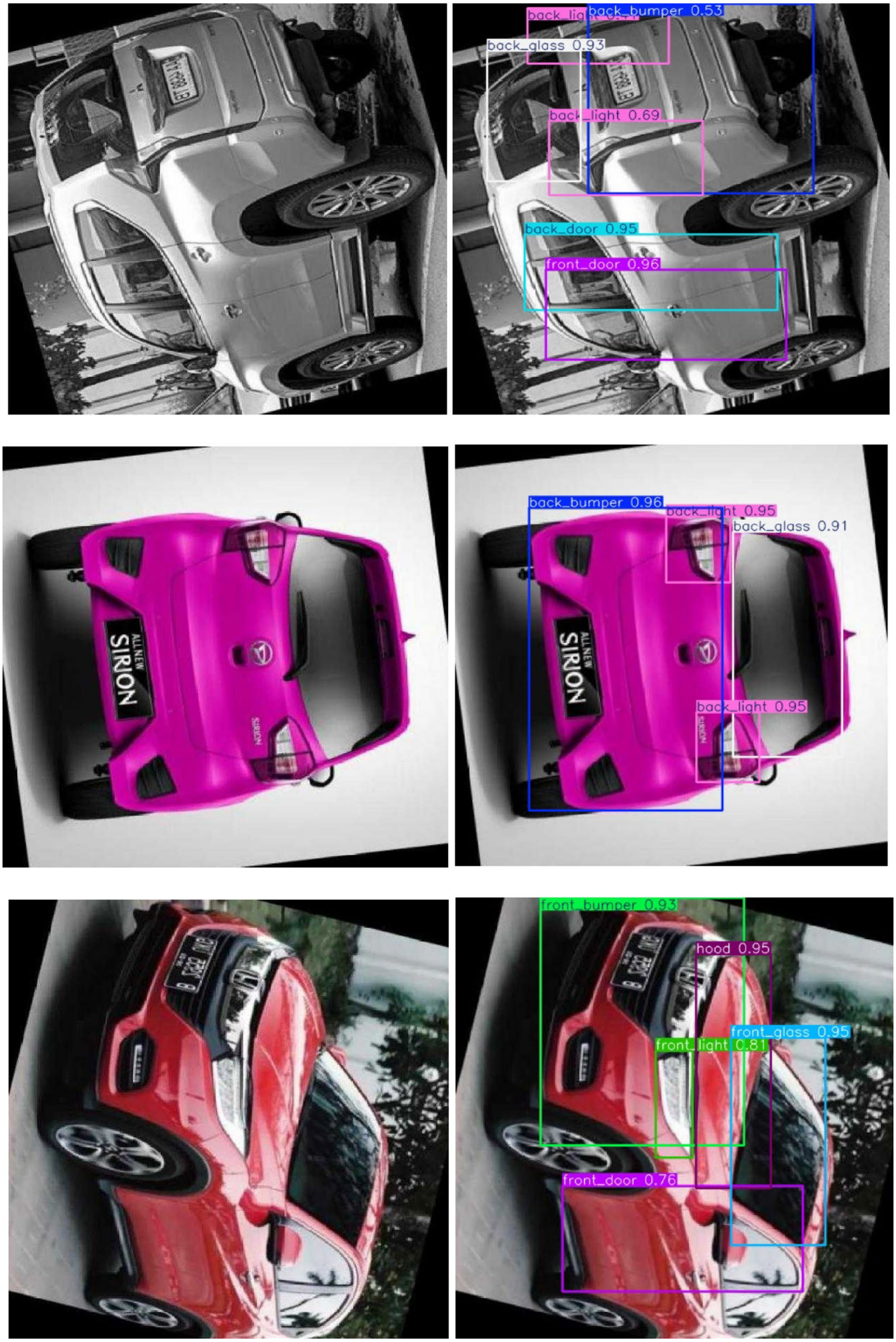


Fig. 8. Test time Predictions of proposed model on unseen data.

mAP@50 by using spatial-channel attention and suppressing background noise. The SCDown module improved localization during down sampling, yielding incremental gains in both F1-Score and mAP@50. The Dual Assignment Head further enhanced detection accuracy in complex scenarios, achieving 69.7% F1-Score and 72.4% mAP@50. When all five modules were combined, the proposed model reached its highest performance

Module name	Precision (%)	Recall (%)	F1-Score (%)	mAP@50 (%)
Baseline YOLO	58.4	74.7	65.4	65.2
YOLO + C2fCIB	60.9	77.1	67.9	68.4
YOLO + SPPF	59.3	75.5	66.3	69.7
YOLO + PSA	62.8	78.3	69.6	72.1
YOLO + SCDown	60.3	76.4	67.3	70.9
YOLO + Dual Assignment Head	62.5	79.1	69.7	72.4
Proposed Model	63.3	81.6	71.3	73.7

Table 4. Ablation results of the proposed model.

of 71.3% F1-Score and 73.7% mAP@50, demonstrating the strongest performance. These results confirm that the architectural enhancements collectively lead to superior detection efficiency, proving the necessity and effectiveness of the proposed framework. The ablation results were given in the Table 4.

Conclusion

This work presents an enhanced car parts detection framework integrating the C2fCIB, SPPF, and PSA modules to improve multi-scale feature fusion, spatial context aggregation, and attention-driven feature refinement. Experimental evaluation demonstrates that the proposed model achieves superior recall (81.6%) and mAP (73.7%) compared to existing state-of-the-art detectors, highlighting its robustness in identifying diverse automotive components. While the inference speed (111 FPS) is lower than YOLOv8, the significant improvement in detection accuracy, particularly for small and complex parts, validates the effectiveness of the proposed architectural enhancements. These findings suggest that the model can be a valuable tool for real-world automotive inspection, safety analysis, and maintenance automation.

Data availability

The datasets generated and/or analyzed during the current study are available in the Ultralytics repository at the following link: https://docs.ultralytics.com/datasets/segment/carparts-seg/. All relevant data supporting the findings of this study can be accessed and utilized in accordance with the repository's usage guidelines.

Received: 22 September 2025; Accepted: 19 November 2025
Published online: 26 November 2025

References

1. Shi, P., Dong, X., Ge, R., Liu, Z. & Yang, A. Dp-M3D: Monocular 3D object detection algorithm with depth perception capability. *Knowl.-Based Syst.* **318**, 113539 (2025).
2. Dong, X., Shi, P., Qi, H., Yang, A. & Liang, T. TS-BEV: BEV object detection algorithm based on temporal-spatial feature fusion. *Displays* **84**, 102814 (2024).
3. Baird, M. L. *August. Image Segmentation Technique for Locating Automotive Parts on Belt Conveyors* 694–695 (IJCAI, 1977).
4. Huang, C., Jia, F., Fang, C., Fan, Y. & Hu, Q. CAR/CAD joint session on image segmentation. *Int. J. Comput. Assist. Radiol. Surg.* **8** (1), S237–S239 (2013).
5. Lu, C., Lian, W. & Yuille, A. Parsing semantic parts of cars using graphical models and segment appearance consistency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 3618–3625. https://openaccess.thecvf.com/content_cvpr_2014/papers/Lu_Parsing_Semantic_Parts_2014_CVPR_paper.pdf (IEEE, 2024).
6. Patil, K., Kulkarni, M., Sriraman, A. & Karande, S. Deep learning-based car damage classification. In *IEEE International Conference on Machine Learning and Applications (ICMLA)* 50–54 (2017).
7. Zhang, T., Xu, H., Zhou, C. & Zhang, L. Part-level car parsing and reconstruction from a single street view. Preprint at https://arxiv.org/abs/1812.06162 (2018).
8. Singh, R., Ayyar, M. P., Sri Pavan, T. V., Gosain, S. & Shah, R. R. Automating car insurance claims using deep learning techniques. In *IEEE International Conference on Multimedia Big Data (BigMM)* 199–207 (2019).
9. Dhieb, N., Ghazzai, H., Besbes, H. & Massoud, Y. A very deep transfer learning model for vehicle damage detection and localization. In *International Conference on Microelectronics (ICM)* 158–161 (2019).
10. Khanal, S. R., Amorim, E. V. & Filipe, V. Classification of car parts using deep neural network. In *APCA International Conference on Automatic Control and Soft Computing* 582–591 (Springer, 2020).
11. Pasupa, K., Kittiworapanya, P., Hongngern, N. & Woraratpanya, K. Evaluation of deep learning algorithms for semantic segmentation of car parts. *Complex. Intell. Syst.* **8** (3), 3613–3625. https://doi.org/10.1007/s40747-021-00397-8 (2021).
12. Lin, Y. Y., Yu, C. C. & Lin, C. H. Automatically segmentation the car parts and generate a large car texture images. In *DeLTA* 185–190 (2021).
13. Shaik, B. An efficient YOLOV9 model for car parts detection and segmentation. *Telecommun. Radio Eng.*
14. Jurado-Rodríguez, D. et al. Semantic segmentation of 3D car parts using UAV-based images. *Comput. Graph.* **107**, 93–103 (2022).
15. Lin, C. H., Yu, C. C. & Chen, H. Y. Augmentation dataset of a two-dimensional neural network model for use in the car parts segmentation and car classification of three dimensions. *J. Supercomputing*. **78** (17), 18915–18958 (2022).
16. Yusuf, S. A., Aldawsari, A. A. & Souissi, R. Automotive parts assessment: applying real-time instance-segmentation models to identify vehicle parts. Preprint at http://arXiv.org/220200884 (2022).
17. ACM. Vehicle appearance parts identification based on instance segmentation. In *Proceedings of the 2023 ACM Conference on Multimedia* (ACM, 2023).
18. Aldawsari, A., Yusuf, S. A., Souissi, R. & AL-Qurishi, M. Real-time instance segmentation models for identification of vehicle parts. *Complexity* **2023**, 1–16 (2023).
19. Kothala, L. P., Jonnala, P. & Guntur, S. R. Localization of mixed intracranial hemorrhages by using a ghost convolution-based YOLO network. *Biomed. Signal Process. Control* **80**, 104378 (2023).

20. Vasanthi, P. & Mohan, L. A reliable anchor regenerative-based transformer model for x-small and dense objects recognition. *Neural Netw.* **165**, 809–829. https://doi.org/10.1155/2023/6460639 (2023).
21. Anupama, H. S., Ranjitha, R. & Srinivas, K. Comparative analysis of deep learning models for car part image segmentation. In *Recent Trends in Computer Vision* 147–162 (Springer, 2024).
22. Kothala, L. P. & Guntur, S. R. *GEL-TTA Net: A Global Ensemble Learning Network for the Localization of Small-Scale and Mixed Intracranial Hemorrhages Through Test Time Augmentations* 1–32 (Multimedia Tools and Applications, 2024).
23. Vasanthi, P. & Mohan, L. Multi-Head-Self-Attention based YOLOv5X-transformer for multi-scale object detection. *Multimedia Tools Appl.* **83** (12), 36491–36517 (2024).
24. Panboonyuen, T. ALBERT: advanced localization and bidirectional encoder representations from transformers for automotive damage evaluation. Preprint at http://arXiv.org/2506.10524 (2025).
25. VigneshArjunRaj. *MAnet: Multi-Scale Attention Network for Car Parts & Damages*. https://github.com/VigneshArjunRaj/MA-Net-CarParts (n.d.).
26. Liu, Y. et al. Segmentation by weighted aggregation and perceptual hash for pedestrian detection. *J. Vis. Commun. Image Represent.* **36**, 80–89 (2016).
27. Dwivedi, M. et al. Deep learning-based car damage classification and detection. In *Advances in Artificial Intelligence and Data Engineering* 207–221 (Springer, 2020).
28. Jocher, G. et al. ultralytics/yolov5: v3. 0. *Zenodo* (2020).
29. Sapkota, R. & Karkee, M. Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 object detectors for computer vision and pattern recognition. Preprint at http://arXiv.org/2510.09653 (2025).
30. Huang, H. & Zhu, K. Automotive parts defect detection based on YOLOv7. *Electronics* **13** (10), 1817 (2024).
31. Dong, X., Shi, P., Liang, T. & Yang, A. CTAFFNet: CNN–transformer adaptive feature fusion object detection algorithm for complex traffic scenarios. *Transp. Res. Rec.* **2679** (1), 1947–1965 (2025).

Author contributions

Raghuveer Chandaluri and Ponduri Vasanthi wrote the main manuscript text. Lakshmi Prasanna Kothala and Y. Chakrapani performed the experiments, Akula Rajesh prepared figures and tables. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to L.P.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.