Hindawi
Complexity
Volume 2023, Article ID 6460639, 16 pages
https://doi.org/10.1155/2023/6460639

Research Article

Real-Time Instance Segmentation Models for Identification of Vehicle Parts

Abdulmalik Aldawsari, Syed Adnan Yusuf, Riad Souissi, and Muhammad AL-Qurishi

Research Department, R&I Division, Elm Company, Riyadh 12382-4182, Saudi Arabia

Correspondence should be addressed to Muhammad AL-Qurishi; qurishi@ksu.edu.sa

Received 18 July 2022; Revised 15 January 2023; Accepted 20 February 2023; Published 11 April 2023

Academic Editor: Lingzhong Guo

Automated assessment of car damage is a major challenge in the auto repair and damage assessment industries. The domain has several application areas, ranging from car assessment companies, such as car rentals and body shops, to accidental damage assessment for car insurance companies. In vehicle assessment, the damage can take many forms, from scratches, minor dents, and major dents to missing parts. Often, the assessment area has a significant level of noise, such as dirt, grease, oil, or rush, which makes accurate identification challenging. Moreover, in the repair industry, identifying a particular part is the first step in obtaining an accurate labor and part assessment, where the presence of different car models, shapes, and sizes makes the task even more challenging for a machine-learning model to perform well. To address these challenges, this study explores and applies various instance segmentation methodologies to determine the best-performing models. This study focuses on two genres of real-time instance segmentation models, namely, SipMask and YOLACT, owing to their industrial significance. These methodologies were evaluated against a previously reported car parts dataset (DSMLR) as well as an internally curated dataset extracted from local car repair workshops. The YOLACT-based part localization and segmentation method outperformed other real-time instance mechanisms with an mAP of 66.5. For the workshop repair dataset, SipMask++ reported better accuracy for object detection with a mAP of 57.0, with outcomes for $AP^{IoU=.50}$ and $AP^{IoU=.75}$ reporting 72.0 and 67.0, respectively, whereas YOLACT was observed to be a better performer for AP^{s} with 44.0 and 2.6 for object detection and segmentation categories, respectively.

1. Introduction

Automotive parts assessment is a crucial process that primarily originates from the requirements of the insurance industry. The process has several other industrial applications that are being rapidly developed, including assembly line analysis (engine parts defect detection) [1, 2], surgical instrument localization [3], healthcare (body pose, skeletal, organ, and cancer/tumor segmentation) [4–7], botanical segmentation (plant and weed detection/smart farming) [8–10], geological map analysis [11, 12], 3D object segmentation with primary applications in autonomous/robotics point-cloud analysis and shape understanding domains [13–15], and construction health monitoring and assessment [16–18]. Most of these applications are increasingly focusing on real-timepixel-level assessment of the area of interest with minimal impact on overall accuracy. Precision is considered paramount in certain applications, such as surgery or weed removal.

The work reported in this study focuses on automotive parts assessment and identification because of its importance in fields, such as accidental damage assessment, insurance claims' processing, car condition assessment for rentals and other automotive portals, and automation of car repair and body shop garages. Conventional vehicle assessment processes for accidental damage repair are initiated by vehicle owners or enforcement authorities, who then follow a complex fault and claim assessment procedure to compensate the party who is not at fault [19]. The same applies for car-rental returns, where drivers and rental firms disagree on ownership of damage to the vehicle at the time of return. The compensation procedures involved in these processes are often time consuming and require weeks to complete. Moreover, a substantial proportion of the claim amount is wasted because of incorrect or unfair assessments, which eventually affect the premiums of the parties involved. First responders and authorities also rely on visual evaluations and record keeping, which is time consuming. A real-time automated visual accidental damage assessment system is a promising area of study.

Recent advancements in edge and mobile device-basedmachine-learning algorithms have made it easier to perform real-time inference of complex vision-based tasks. MobileNet is a lightweight deep neural network, specifically focused on hyperparameter tuning to maintain a trade-off between the latency and accuracy of the classifier [20]. Similarly, DeepDecision is another architecture that focuses on balancing trade-offs, such as the impact of video resolution, parameterizing accuracy, end-to-end latency, and video compression, to develop an edge video analytics deep learning framework [21]. The object bounding box estimation mechanisms of R-CNN [22], Fast R-CNN [23], Faster R-CNN [24], SSD [25], and YOLO [26], are well-known algorithms. In industrial applications, faster detection and segmentation mechanisms are increasingly used in domains, such as defect/weld image surface inspections [27–30], manufacturing/textile line quality analysis [31], smart crops and farming [32], semiconductor fabrication and design processes [33], and engineering condition monitoring fault diagnosis owing to their faster and more accurate performance [34–36].

These frameworks require a large amount of training data. Moreover, for the current scope of work, identification of automotive parts cannot be limited to determining bounding boxes because of the complexity of automotive parts and damage outlines; hence, it is difficult to achieve end-to-end detection. Moreover, automotive datasets are extremely large owing to the variability of the model variants. Hence, at the feature extraction stage, an increase in the number of convolutional layers leads to gradient disappearance or explosion. To address these challenges, an array of deep network architectures, each with their own strengths and weaknesses, was introduced. He et al. proposed a resi network (ResNet) that assisted in model convergence by utilizing a residual module; this module accelerated the neural network training process [37] by combining it with the target detection model using Mask R-CNN [38]. The mask R-CNN algorithm is among the first deep learning mechanisms to combine object detection and segmentation techniques to improve the overall identification accuracy. Other similar deep learning variants in this domain include AlexNet [39], VGGNet [40], and GoogLeNet [41].

A major drawback of these architectures has been their primary focus on achieving higher accuracy instead of improving latency and performance, which makes them unsuitable for real-time image processing. Within the scope of car parts segmentation, most object detection and segmentation techniques offer two major shortcomings: (1) the ability to process images in real time and (2) the model's ability to segment adjacent parts uniquely. One such case is that of the front and rear doors, which, in many cases, have very little separation to be uniquely identified. These limitations lead to a well-known image processing principle known as "instance segmentation," which is one of the most performance-intensive methodologies in deep learning because several unique classes are often adjacent and have high similarity. To date, previous models have focused on accuracy rather than speed.

For mobile/edge applications, smaller network architectures are more efficient in distributed training, require less bandwidth during remote model updates, and most importantly, can be deployed on smaller, mobile, and edge devices [42]. Moreover, limited space on mobile and edge devices' remains a major challenge. Regardless of large storage sizes, installing a mobile application that requires 500 MB of memory in the form of a weights file is a substantial limiting factor for users owing to the fact that most smartphone applications are not above 100 MB in size. Model pruning is a well-known method that is used to create smaller and more efficient neural networks. This technique involves eliminating unnecessary values in the weight tensors of a neural network, resulting in a compressed network that runs faster and has a reduced computational cost during network training [43].

This study explores the context of utilizing and improving existing mobile/edge-device-level deep neural network algorithms for car part segmentation. The proposed methodology employs various object detection architectures to identify vehicle parts. The phase is crucial because the identification of missing vehicle parts using a standard object detection algorithm presents a high number of true negative (missing part identification) or false positive (missing part identification in a nonvehicle background) cases owing to the high variability of backgrounds that are visible in missing car parts. In the present case, the part detection methodology trained 29 car part types with each part and then further labeled them to have three unique damage categories: scratches, minor, and major dents. The severity of each damage type was then mapped to generate the overall damage level of the category as a regression score. Generalized scoring is performed owing to the unavailability of a single part; the labor cost prediction paradigm is used owing to the variability of these in different world economies. In addition, we have developed comprehensive car parts datasets. The first dataset was extracted from national government databases of accident information and data management activities for law enforcement, insurance, and other purposes. The second dataset involved samples from the initial dataset extracted from local resources. The third dataset involved data augmentation and class merging phases in a bid to improve overall segmentation and detection accuracy by minimizing crossclass similarity. Next, we implemented a single stage for real-time instance segmentation algorithm (Yolact) [44]. Finally, a comparison of the two-stage detection mechanism with a single-stage instance segmentation mechanism (SipMask) and its variant (SipMask++) [45] is conducted. We compared four algorithms against an initially extracted third-party dataset (DSMLR) as well as one of our internally extracted dataset. The YOLACT-based part localization and segmentation method as shown in 4 outperformed other real-time instance mechanisms with a mAP of 66.5. For the workshop repair dataset, SipMask++ reported better accuracies for object detection with a mAP of 57.0 with outcomes for $AP^{IoU=.50}$ and $AP^{IoU=.75}$ reporting 72.0 and 67.0, respectively, whereas YOLACT was observed to be a better performer for AP^{s} with 44.0 and 2.6 for object detection and segmentation categories, respectively. Generally, the main goal of this research is to evaluate the performance of various CNN-based instance segmentation models on multiple datasets of vehicle parts. We also provide insights on how to improve the performance of these models through data augmentation and model hyperparameter optimization, based on the challenges we encountered during this work. The remaining of our paper is organized as follows. Section 2 presents a review of existing deep-learning methodologies, existing vehicle parts, damage assessment processes, and the scope of the research. Section 3 presents the architecture, including the instance segmentation methodologies employed, including the YOLACT, SipMask, and SipMask++ architectures. This section presents an in-depth comparison of the performance of these architectures. Section 4 concludes the paper with a discussion of the key outcomes and future directions of this study. A preprint of this work has previously been published [46].

2. Background and Overview

Industrial visual inspections are gaining rapid advancement and interest. Advanced computer vision and deep learning methodologies have been explored to facilitate automation while addressing problems, such as weakly annotated/sparse datasets [47], depth-wise separable convolutions (MYOLOV3-Tiny) [48], mixed-supervision annotation for quicker and better identification model training, and overlapping or complex region localization [49]. Most of these techniques focus on improving the deep-learning modeling pipeline either by diversifying the data extraction process, tuning the pipeline parameters, enhancing the training mechanism, or evolving the existing architecture, to gain on the underlying segmentation or detection techniques. Table 1 shows a list of several abbreviations that are used in this work for different tasks.

The research presented in this paper focuses on the evaluation and application of real-time segmentation methods for the identification of complex part groups in vehicles. The layout of automotive parts follows several spatial constraints that can be very strict in some cases, such as side mirrors connected to doors, or diverse, such as a backdoor window enclosed within the rear door or in the body of the vehicle itself (for two-door vehicles). These constraints can only be addressed using statistical multimodel distribution models. Active appearance models [50] and deformable part models [51] are two such models that are used to ascertain, in advance, various combinations of parts that may form in a vehicle. Such combinations may not be sufficient to cover all shape groups because of the variations induced in automotive images owing to angle, rotation, and object deformation.

Deep convolutional neural networks (CNNs) have recently demonstrated outstanding results in a range of computer vision problems. During the past decade, much interest and research has focused on aspects, such as hyperparameter tuning and optimization, network pruning, and connectivity learning to improve the model size and performance, specifically on mobile and edge devices [52, 53]. Moreover, model pruning and connectivity learning are being emphasized to increase model size and network performance [54, 55]. Semantic and instance segmentation mechanisms are two object shape estimation techniques commonly used to identify object boundaries.

At this stage, detailed information of the vehicle is extracted. In conventional cases, this may include marking a vehicle template to record damage areas that are specific to the accident as well as the severity of the damage. The party involved in the accident or the officer in charge of record-keeping may also capture images and videos for insurance and other investigative purposes. The information is then consolidated by relevant authorities, such as insurance agencies, to prepare damages grant claims to be given to the affected parties.

The objective of smart damage assessment is to automate and streamline the entire process. In most vehicular damage cases, the assessment comprises four core phases as explained in Figure 1:

(1) **Evidence data extraction**: Extraction of visual evidence, such as pictures and videos, taken to clearly record the damage. It is ensured that the damage context is suitably preserved such that the images are taken from a suitable distance to allow the algorithm to differentiate between the various car parts involved.

(2) **Automated parts identification**: The visual information is then submitted to a vehicle part identification algorithm that utilizes pretrained artificial intelligence (AI) models to identify the boundary part. This phase included the vehicular brand, type, and age of the vehicle to select the identification model relevant to that model type. For example, a model trained to predict a 3-year-old sedan cannot be used to identify the bumper of less-than-a-year-old SUV.

(3) **Automated damage assessment**: this phase includes the damage area extraction of each car part along with the AI logic that estimates the part cost as well as the labor estimate.

(4) **Damage recommendation output**: the part and damage information are then modeled against other relevant information, such as car model and type, to generate an accurate damage report. This report may contain assessments ranging from car part price to the overall labor cost that the repair will need. The report may also generate a recommendation if complete replacement of the part is required.

Table 1: Abbreviation definitions.

Abbreviation	Definitions
SipMask	Spatial information preservation for fast image and video instance segmentation
YOLACT	You only look at the coefficients
mAP	Mean average precision
AP	Average precision
CNN	Convolutional neural network
BBox	Bounding box
FCN	Fully convolutional network
AI	Artificial intelligence
SUV	Sport utility vehicles

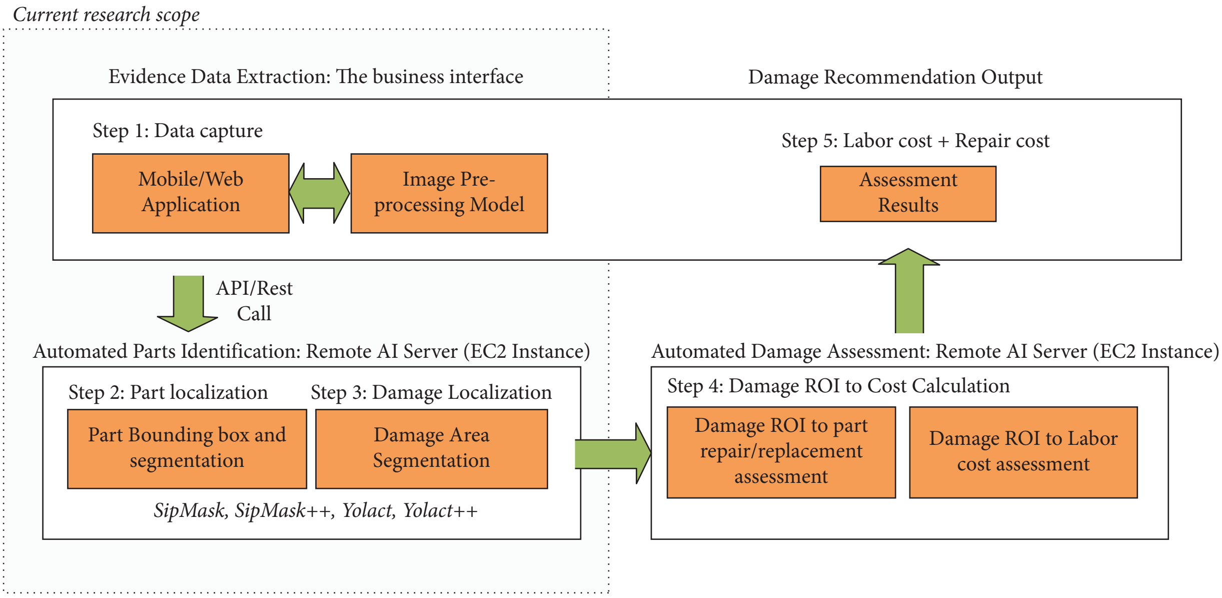
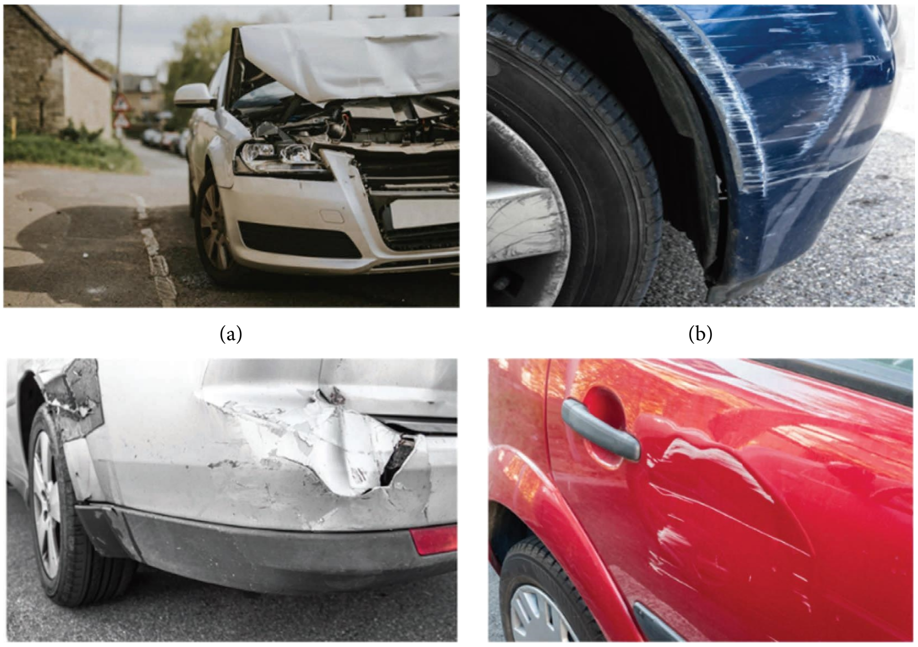

Current research scope
Evidence Data Extraction: The business interface
Damage Recommendation Output
Step 1: Data capture
Mobile/Web Application
Image Pre-processing Model
Step 5: Labor cost + Repair cost
Assessment Results
API/Rest Call
Automated Parts Identification: Remote AI Server (EC2 Instance)
Automated Damage Assessment: Remote AI Server (EC2 Instance)
Step 2: Part localization
Step 3: Damage Localization
Part Bounding box and segmentation
Damage Area Segmentation
SipMask, SipMask++, Yolact, Yolact++
Step 4: Damage ROI to Cost Calculation
Damage ROI to part repair/replacement assessment
Damage ROI to Labor cost assessment

Figure 1: Description of the scope of research presented as well as the overall car damage assessment use case.

2.1. Anatomy of Car Part Damage. A vehicle comprises a set of parts that are joined in a complex assembly, where the distribution of damage cannot be directly ascertained. This indicates that the damage done to a front bumper will have a different cost assessment paradigm than that of a door. Moreover, different severity levels of the parts also have variable impacts. For instance, back bumper and chassis damage to a rear-engine vehicle would have a different assessment regime for major damage than that of a front-engine vehicle. In a conventional setting, a damaged car is first assessed by a designated workshop, where the outer damage is first categorized as belonging to one of many damage types. Based on the severity of the damage, a decision is made to repair the part, replace it, or completely write-off the vehicle. The part repair cost estimate is often diverse for various cases, such as scratches and minor dents, depending on the spread of damage to each part as well as the part type. For instance, a minor, scratchless dent would often require a pull-correction without repainting, as shown in Figure 2(a), whereas a scratch-and-dent combination on a vehicle door would not only require a larger surface area to be repainted but also the structural deformation to be corrected, as shown in Figure 2(b). However, damage leading to substantial structural deformation may generate a total replacement payout (vehicle written-off). Minor dents are often corrected using special part-molds and heat guns, whereas scratches are repainted to the part color code.

2.2. Identification of Missing Parts. Another major challenge in car part identification is the missing part itself, which is quite common in detachable vehicle sections, such as bumpers, side mirrors, and wheel caps, as shown in Figure 2(d). Owing to the high variability of backgrounds in missing car parts, it is often challenging to classify an absence based merely on the background of that part. This is very common in major accidents where a car part is either completely absent or distorted to the extent that it cannot be visibly identified. One such example is that of the left fender, as shown in Figure 2(c).

2.3. Generalizing a Model for Accurate Segmentation. Another major challenge reported later in this study is that of training a single generic car part identification model, as presented by Pasupa et al. [56], where well-established deep learning models, such as Mask R-CNN and GCNet, generated low mAP outcomes ranging from approximately 48.5 for GCNet to 54.3 for HTC over the ResNet-101 encoder for



(a)

(b)

(c)

(d)

Figure 2: Various damage categories with variable repair outcomes including (a) deformed/missing parts, (b) scratches, (c) major dent, and (d) scratch-and-dent combination.

part detection and 43.0 to 65.2 for GCNet and CBNet for ResNet-101 and ResNet-50, respectively, for part segmentation. The lower rates for both part localization and segmentation form a substantial basis for this study, as this approach utilized a single model to identify car parts for a wide range of designs, despite the high variability of automotive designs.

3. Proposed System Architecture and Methods

So far, most reviews in vehicle damage assessment have focused on two core areas: car parts segmentation and damage assessment. Both stages are generally connected sequentially to obtain a more accurate assessment of part damage based on the model, year, and actual price of the car at the time of evaluation. Subsequently, the damage assessment phase is divided into parts cost and labor cost prediction models, in addition to advisory aspects that vary from case to case depending on the fact that the underlying policy requires replacement based on the level of damage incurred to that part. A model trained for a specific group of cars (e.g., sedans or SUVs) may be reasonably accurate for certain parts, but a single universal damage assessment model cannot be efficiently trained. This study makes several contributions to the field of vehicle part segmentation. First, we compared the performance of four deep learning models proposed by Pasupa et al.: Mask R-CNN, GCNet, PANet, CBNet, and HTC over the same dataset on the YOLACT, YOLACT++, SipMask, and SipMask++ algorithms. We then present a more detailed part detection architecture on a dataset collected in-house and compare the performance of the two single-stage instance segmentation mechanisms (YOLACT) as shown in Figure 3 against two single-stage mechanisms (SipMask/SipMask++).

3.1. Deep-Learning Methodologies for Car Parts Segmentation. Car-part identification can be addressed in two unique contexts in deep learning. One common approa4ch originates from part detection, where each part is labeled as a bounding box. However, because a car is a symmetrically complex combination of smaller parts/components, annotating each part with a matching background creates lower cross-class variability, which may lead to a higher level of cross-class mismatches. The other method is to have a pixel-level segmentation of each car part, leading to a more accurate polygonal representation of each car part. This technique is considered more reliable, although the inference time for a semantic or instance segmentation mechanism has so far been a computationally expensive task that has focused more on accuracy and performance. Recently, real-time object segmentation has gained traction owing to improvements in hardware and methodology in soft computing techniques [57]. In deep learning, various architectures manage frame rates of 40+ fps for instance segmentation. Mask R-CNN is the most common instance segmentation technique that comprises a two-stage modeling mechanism that involves an object proposal stage extending to a segmentation calculation, mask, class confidence, and bounding box offset estimation stages. Hence, the second stage is essentially the calculation of per-instance mask coefficients. Because the two tasks run in parallel, the segmentation process is substantially faster than that of the other methods. The part detection system was evaluated on

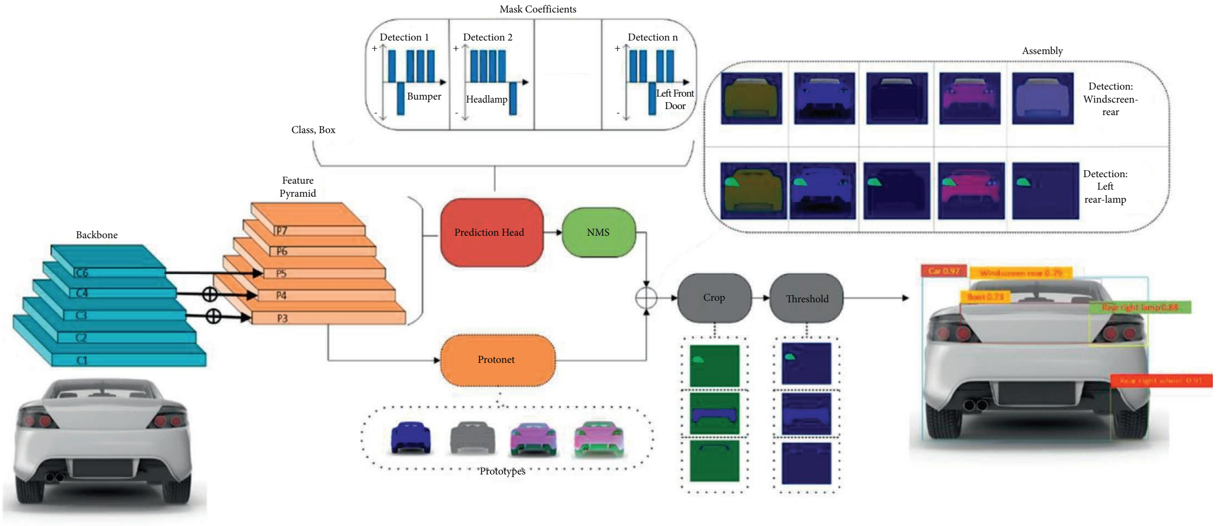


Figure 3: YOLACT architecture explained for the car parts segmentation use case.

three major variants of instance segmentation techniques: YOLACT, SipMask, and SipMask++.

YOLACT is a recently proposed instance segmentation mechanism reported to exhibit a superior trade-off between performance and accuracy by predicting a dictionary of basis masks (category-independent maps) for an image and a set of coefficients that are instance-specific. However, the method has inferior results compared to the two-stage methods. YOLACT is an instance segmentation mechanism presented by Bolya et al. as the first method to attempt real-time instance segmentation [44]. Within the scope of this study, instances play a significant role in car parts segmentation; this study primarily focuses on the evaluation of YOLACT and extends and compares its performance against other well-known and reported segmentation mechanisms, including the SipMask, SipMask++, and YOLACT++ paradigms [58], as shown in Figure 4 So far, SipMask++ has been reported to have the best mask AP (35.4), followed by YOLACT-550++ (34.6) on a ResNet-101 backbone. However, YOLACT precedes SipMask in framerate, 45.3 fps over SipMask's 41.7 fps; SipMask++ has a significantly lower rate of 27 fps. The outstanding performances of the YOLACT and SipMask regimes inspired the evaluation of these techniques for the segmentation of car parts.

The SipMask module is a lightweight spatial preservation technique that preserves the spatial information of each car part within a bounding box. The approach is a single-stage instance segmentation method aimed at faster inference speeds by avoiding the proposal generation and feature pooling stages. However, the accuracy of the single-stage approaches is poor. Figure 5 shows the overall architecture of the SipMask instance segmentation mechanism used for car part segmentation, which comprises fully convolutional mask-specialized classification, and regression branches. The SipMask design focuses on the spatial preservation (SP) module in the mask-specialized regression branch and performs tasks to align features and generate spatial coefficients. In this approach, each predicted bounding box contains a separate set of spatial coefficients. These spatial coefficients preserve the spatial information contained within each object instance and, hence, allow a better definition of adjacent spatial objects, particularly where the spatial similarity of such objects is high. One such example is that of the front and rear car doors, where the only separation between the two is often a vague line. The mask-specialized regression branch, conversely, predicts the bounding box offsets as well as a set of basis masks that are category independent. Figure 5 shows an example car image where a set of spatial coefficients for a basis mask is generated, including the bumper, rear lights, and boots. The SipMask case and YOLACT comparison is shown in Figure 6, where only the case of "bumper" instance is considered for the sake of simplicity. A comparison of the corresponding mask generations for both YOLACT and SipMask is presented in (a) and (b). Map M_j shows the linear combination of a set of single sets of coefficients and basis masks. In YOLACT, the final mask $\widehat{M}_j$ is obtained by pruning and thresholding mask M_j. Figure 6(b) shows the second quadrant ($k = 2$) spatial coefficient generation for bounding box j. This results in a separate set of spatial maps M_{ij}, where i is the quadrant number for the bounding box j. The spatial maps are then pruned and integrated using simple addition and thresholding to obtain the final map $\widehat{M}_j$. This spatial relationship reduces the influence of the adjacent "boot" instance and generates a better mask prediction. The process is further repeated for a better spatial comparison of "boot" to "LT Tail lamp" and "RT Tail lamp" instances.

3.2. Dataset Description. In the research community, only a limited number of datasets are publicly available. Pasupa et al. [56] used the most recent car part dataset comprising 500 car images captured from various angles for a wide range of cars with 18 car part masks and bounding boxes. This

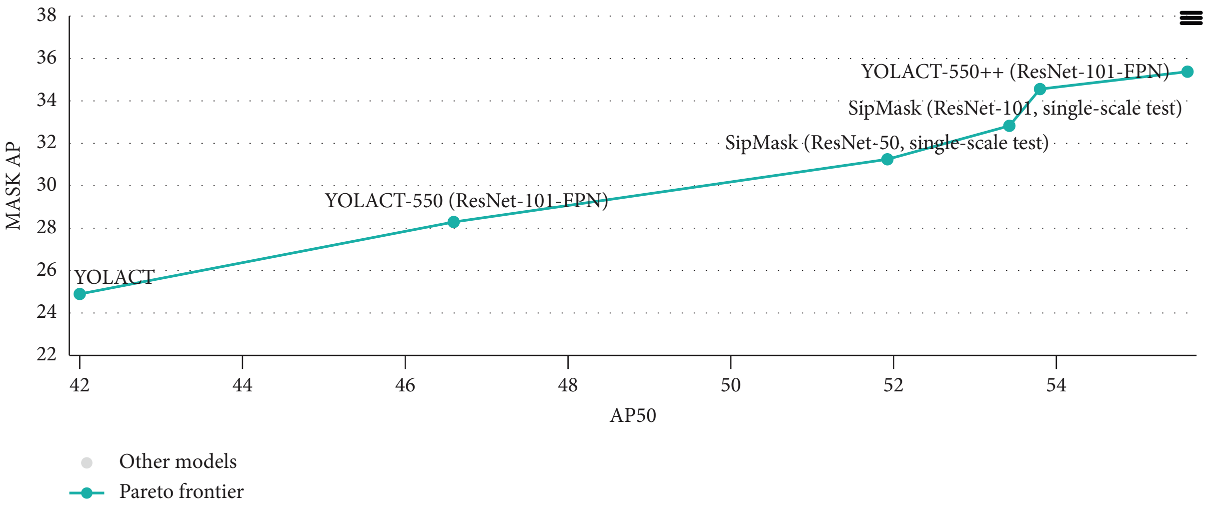
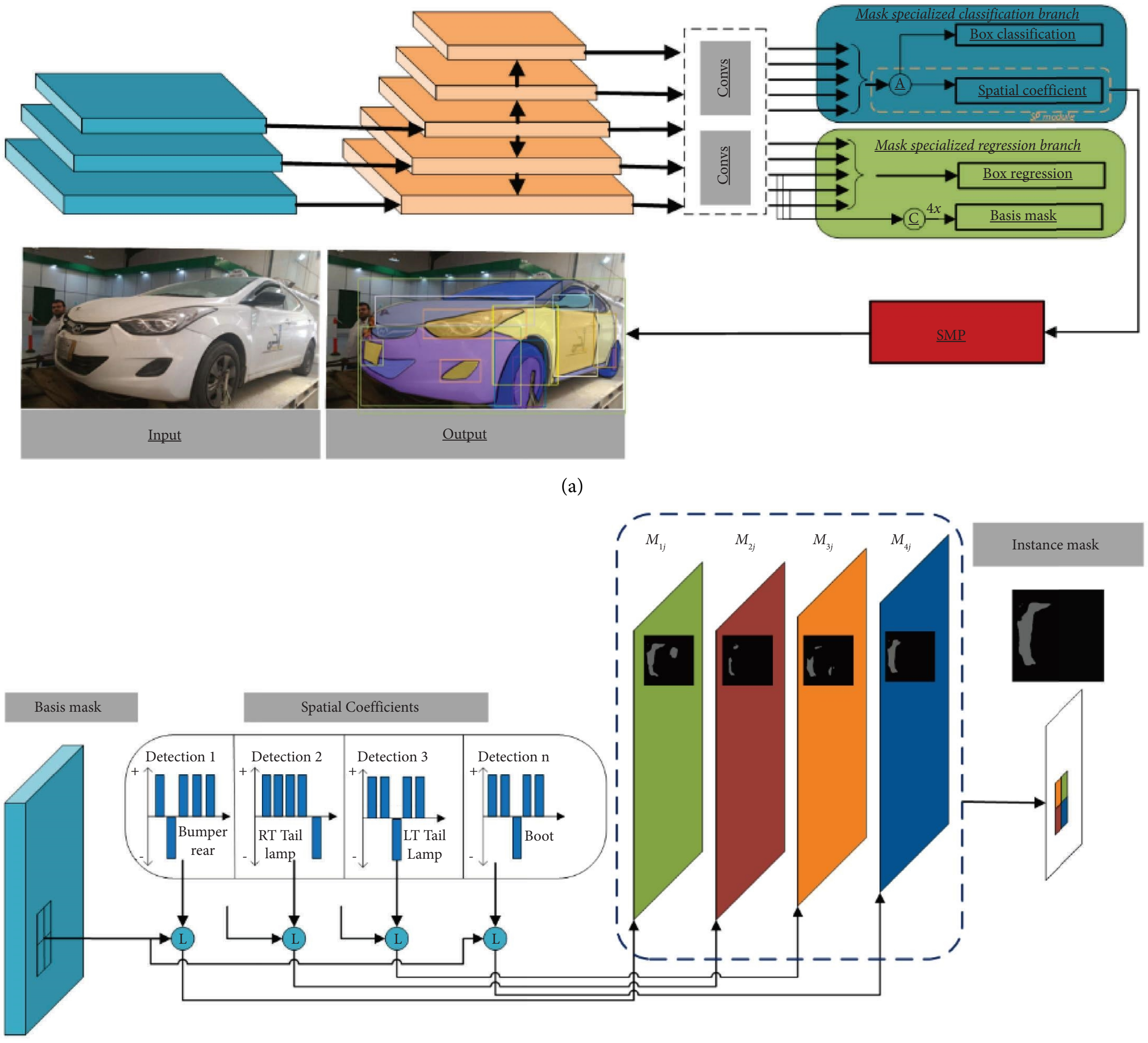


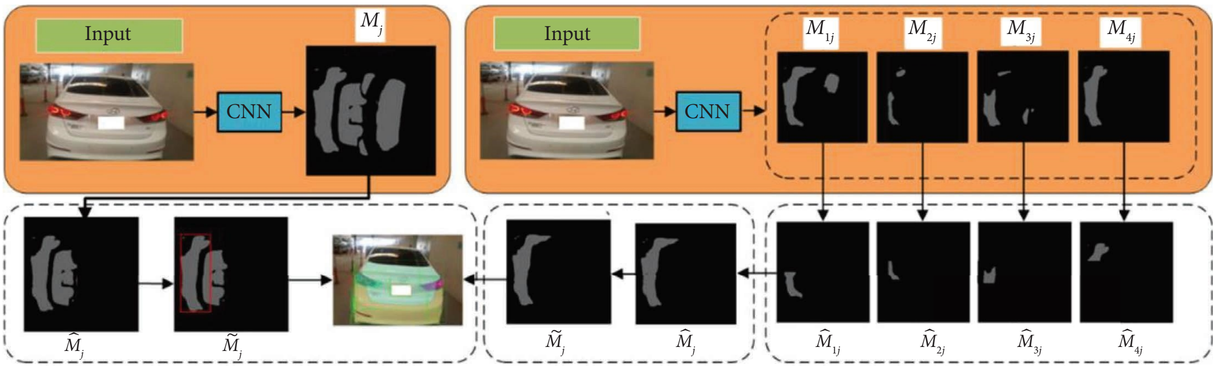
Figure 4: Performance comparison of real-time instance segmentation techniques including YOLACT/YOLACT-550 (ResNet-101), SipMask (ResNet-50)/SipMask (ResNet-101), and YOLACT-550++ (ResNet-101-FPN) [58] on the COCO dataset.



(a)

(b)

Figure 5: SipMask architecture within the car parts segmentation context covering (a) feature alignment and spatial coefficient generation process and (b) basis mask to spatial coefficient generation process leading the final instance segmentation of the “Bumper-rear” instance.



(a) Yolact-based mask generation based on a single set of coefficients

(b) SipMask-based sub-region-specific spatial coefficients

FIGURE 6: A YOLACT to SipMask comparison with (a) a single coefficient set (nonspatial) example compared against (b) subregion-specific spatial coefficient generation using SipMask.

dataset was used to train a YOLACT model to assess the suitability of the dataset for segmenting car parts. Owing to the high variability of the dataset covering sparsely distributed class groups and models, the data generated a very low BBox mAP of 0.33, improving the MAP50 accuracy to 0.65. Thus, it is necessary to have a more organized and balanced dataset aimed at specific car types. Because the aim of this study was to perform a damage cost assessment, the car model also played a substantial role. Hence, training a part detection model from a variable range of car models would also generate part and labor cost inaccuracies at later stages. Hence, within the scope of this study, the car model considered for data extraction was the Hyundai Elantra model from the past seven years.

The dataset for this study included 1032 images containing 11707 annotated car parts originating from 29 unique classes (Figure 7). Most images were extracted from pictures captured from car workshops where vehicles were brought for repair. The initial extraction process was arbitrary, and pictures were taken for the shop's own record-keeping purposes. The existing results are reported on these arbitrary images, although the data collection method has since been organized substantially for model retraining and further tuning stages and includes the capturing of some or all instances of the car. The data did have an inherent bias towards the "Wheel" and "Wheel_cap" categories (Figure 8) though owing to the unique spatial characteristic of these classes; the bias was assumed to have lesser impact on any other parts.

3.3. Training of the Instance Segmentation Models. The training objective of this study was to learn the features of parts to identify and segment them in unseen images. Because the goal of this study is to identify and assess damage, the presumption here was that damage done to any model type could more broadly be identified on that specific model and not on any car models in general owing to a variable vehicle cost range and hence the rationale behind focusing only on a single car model. The model selection specifically focused on the following objectives:

(i) The model's ability to separately identify overlapping parts

(ii) Determination of the model's capability to process both part localization and segmentation in real time

(iii) The precision of both the localization and segmentation routines

Based on the evaluation in Section 3.2, two algorithm genres were evaluated, along with their extensions. Among these, SipMask/SipMask++ differed from the YOLACT regime because of its single-stage segmentation network that elevated the segmentation performance without any tradeoff in speed using a novel architecture called spatial preservation (SP), in which the network generated a set of spatial coefficients per box prediction. During the originally reported tests on the COCO dataset, the information of the adjacent objects was preserved. Because the underlying base architectures for YOLACT and SipMask are the same, the differences come to play at their heads. The two heads in YOLACT are (1) anchor-based regressions to produce bbox (bounding box), class, and coefficients and (2) Protonet to produce image-sized masks. Both branches were combined into a single network at the end to produce a rectified mask along with the bounding box. By contrast, in SipMask, a single regressor yields a base mask and bounding box that are fed thereafter to the ConvNet for the generated spatial coefficient and box classification. The novelty of the SP module is that the coefficients and basis masks are divided into the KxK region to preserve and delineate the adjacent object masks from each other. SipMask++ claims an exceptional performance on the level of small pictures (approximately 550 × 550), while the original YOLACT outperformed others for big pictures (approximately 1330 × 800). We also observed better classification performance in two-stage networks, such as YOLACT. Our reasoning is that separate networks dedicated to such tasks should yield improved segmentation results. Moreover, there are plans for the future to integrate DarkNet as a backbone because it generates good performance in terms of bbox and classification, which is likely to play a substantial role in the optimization of the network as well as in the

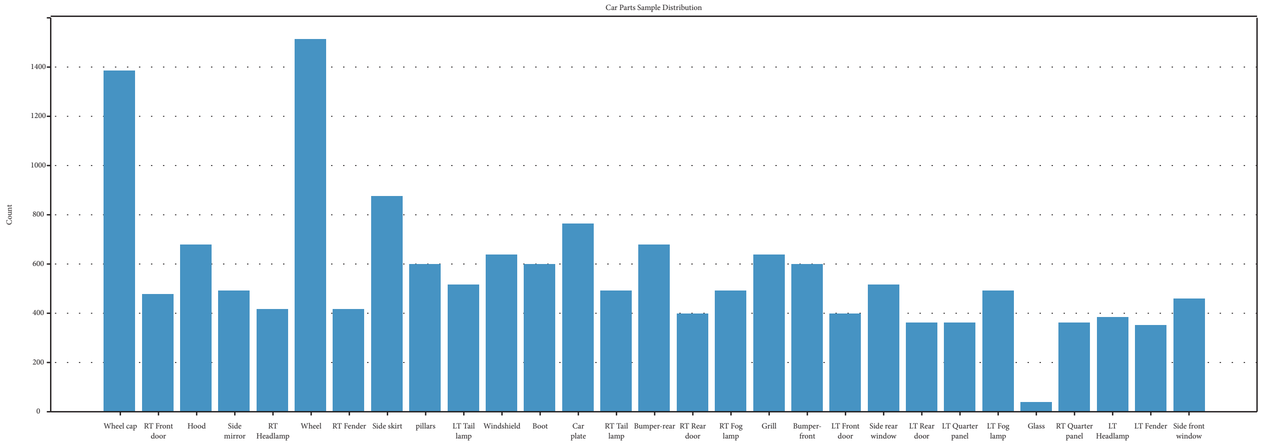


Figure 7: Distribution of various car part types in the dataset, including 29 different classes.

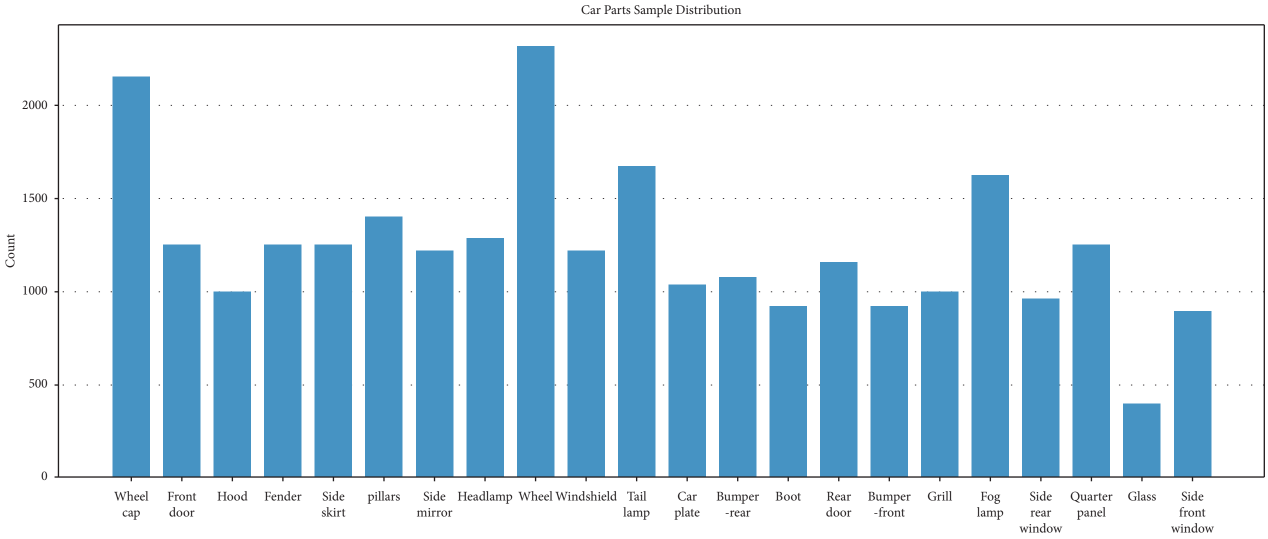


Figure 8: Distribution of various car part types merged in the dataset including 22 different classes.

incorporation of the Bayesian layer or graph similarity nodes to obtain adjacency information benefits.

3.4. Experimental Setup and Results. To train the ML model for car parts segmentation, the system was trained on three car parts datasets, as follows.

3.4.1. Dataset 1 (DSMLR). Initial DSMLR dataset with 18 classes with instance masks and bounding boxes are:

(i) Dataset size: 500 images of sedans, pickups, and SUVs were predominantly scraped from the online images.

(ii) Format: the dataset is available in the COCO Challenge format from [56].

(iii) Preparation: the images were normalized to 1024 × 1024 pixels, with zero padding used to maintain the aspect ratio. The dataset was randomly partitioned into a training set (70%) and test set (30%). The model was trained for 300 epochs while saving the best model for the epoch that generated the lowest validation loss.

(iv) The losses were calculated for

 (1) Classification loss
 (2) Localization loss
 (3) Parts segmentation task loss

(v) A crossentropy loss was used to calculate validation losses with the stochastic gradient descent (SGD) method for parameter optimization with a learning rate of 0.1 and weight decay of 0.0015. The experiments were run for predetermined epochs, and the best intermediate model was stored despite the training running up until the entire number of epochs is reached.

3.4.2. Dataset 2 (Workshop Repair Dataset). The workshop repair dataset was gathered, as described in Section 3.2, with 29 classes with instance masks and bounding boxes:

(i) Dataset size: 1032 images of sedans, pickups, and SUVs.

(ii) Preparation: the images were normalized to 550×550 pixels, with the dataset randomly partitioned into a training set (70%) and test set (30%). This normalization is different from the previous one because it involves using low resolution images to improve model generalization. Additionally, using low-resolution images with a larger batch size is more effective than using high-quality images with a small batch size. Padding images to match the sizes in dataset 1 is also unnecessary because many of the added pixels will be zeros, which do not contribute to model training. The model was trained for 300 epochs while saving the best model for the epoch that generated the lowest validation loss.

(iii) The losses were calculated for

(1) Classification loss
(2) Localization loss
(3) Parts segmentation task loss

(iv) A crossentropy loss was used to calculate validation losses with the stochastic gradient descent (SGD) method for parameter optimization with a learning rate of 0.1 and weight decay of 0.0015. The experiments were run for predetermined epochs, and the best intermediate model was stored despite the training running up until the entire number of epochs is reached.

3.4.3. Dataset 3. For this dataset, the number of samples was increased to 2200 images, while also incorporating a low number of images of different brands belonging to the same vehicle type, that is, sedans. These images represent less than 20% of the dataset. In addition, to overcome the shortage of dataset samples, removing the positional labelling of some car parts helped improve the model's performance by a significant margin. The class merger reduced the total number of classes to 22, from 29 unique classes. Positional labels were meant for the classes that had left or right classes for the same class type (e.g., the side doors), and a single class label was allocated. We added out-of-distribution images (different car brands but of the same type, i.e., sedan) to improve the model diversity and robustness.

3.4.4. Data Augmentation. The initial technique explained in Figure 6 exposes the underlying models to a wide variety of test images with substantial variations in backgrounds, unique car colors, unclear, angled, or body parts, or flipped images. Moreover, there was substantial similarity between the left and right car parts, such as doors or mirrors, which resulted in a significant cross-class bias. To overcome these shortcomings, the following augmentation steps were performed:

(i) Augmentation to multiply less represented classes, such as the window glass

(ii) Addressing light intensity variations by adjusting gamma values

(iii) Merging side-classes (e.g., left and right tail lamps)

(iv) Flipping to adjust for left- and right-sided images both horizontally and vertically

(v) Incorporating rotations because the dataset was identified to be skewed at certain angles by injecting standardized images as a simple FCN on top of a VGG16 backbone with four rotation categories (0, 90, 270, and 360)

(vi) To compensate for the color variation, the model was color agnostic by introducing RGB shift, contrasting, and normalization

(vii) To address the vague part boundaries, blurring and hue saturation was used

Moreover, because very small objects show a low mAP, a copy-paste augmentation methodology was introduced [59]. This substantially improves the baseline results of SipMask. The technique generates new training images with different backgrounds, part-visible car parts, and scattered car parts (e.g., detached bumpers). In contrast to other conventional datasets, where objects can be isolated from the background, the bounding box regressor benefitted from the multipart complexity of the car parts dataset.

3.4.5. Hyperparameters. The following parameters produced the best results for the reported models. Some parameters were common to all models, while others were based on recommendations from the authors. We used a learning rate of 0.1 and weight decay of 0.0015. To address the imbalance in the number of classes in the images, we modified the focal loss parameters to handle imbalanced datasets and encourage the model to pay more attention to hard cases, i.e., very few cars of some brands, by increasing the gamma value to 2.2. The alpha parameter was kept at 0.25, as recommended for balanced class datasets. Although the datasets only had one imbalanced class (wheels), we found that it was unnecessary to change the parameter.

3.5. Performance Evaluation Metrics. During training, each of the segmentation algorithms compared the predicted polygons with the ground-truth based on the intersection over union by updating its parameters at each iteration as follows:

$$\mathrm{IoU} = \frac{(\text{Area of overlap})}{(\text{Area of union})}. \tag{1}$$

The underlying principle, as per the COCO Challenge, was to obtain an IoU > = 0.5, hence, indicating any overlap of more than or equal to 50% as a true prediction. For the car parts' prediction use case, this threshold was maintained as it stood. Each of the four algorithms presented in Table 2 was evaluated for six out of the 12 detection evaluation metrics defined by the COCO Challenge, including the average precision primary challenge metric AP, PASCAL VOC metric $\mathrm{AP}^{\mathrm{IoU}=.50}$, and strict metric $\mathrm{AP}^{\mathrm{IoU}=.75}$ and across scales

AP of $\mathrm{AP}^{\mathrm{small}}$, $\mathrm{AP}^{\mathrm{medi\,um}}$, and $\mathrm{AP}^{\mathrm{large}}$. mAP was calculated with an average of 20 intersection over union (IoU) values of each object between the thresholds 0.5 and 0.95 as follows:

$$\mathrm{mAP} = \frac{\sum_{i=0}^{14} AP_{50+(2.5*i)}}{20}. \qquad (2)$$

$\mathrm{AP}^{\mathrm{IoU}=.50}$ was calculated as a single IoU of 0.50 and 0.75 corresponding to the metrics, $\mathrm{AP}^{\mathrm{IoU}=.50}$ and $\mathrm{AP}^{\mathrm{IoU}=.75}$, respectively.The $\mathrm{AP}^{\mathrm{IoU}=.50}$ and $\mathrm{AP}^{\mathrm{IoU}=.75}$ metrics indicate that the IoUs were greater than or equal to 0.5 and 0.75 intersections of the original and detected bounding boxes, respectively.

3.6. Experimental Results and Discussion. This section presents a performance comparison of the three algorithms (SipMask++, SipMask, and YOLACT) against the DSMLR and Repair Workshop datasets (both nonaugmented and augmented). The comparisons made are based on

(1) The instance/semantic segmentation results of car parts against their ground truth

(2) The robustness at various arbitrary angles and zoom levels

(3) The computation efficiency (frames per second)

The use of those algorithms resulted in a significant decrease in both speed and model size. For example, in [45], the frame per second of Mask R-CNN using a ResNet101 backbone was 116 ms, while SipMask's fps was 89 ms on the coco-test dataset, with both models showing comparable AP of around 38 on large images (1333 × 800). Our dataset has an average mask size above 96^2 pixels, and we found that SipMask++ performed the best in this category, with an AP(L) of 56.8, outperforming both two-stage models such as MASK R-CNN and one-stage models such as PolarMask. This held true on our dataset, as shown in Table 3. We had fewer medium and small mask sizes, and in these cases, the two-stage model performed better and had higher fps, making it unsuitable for real-time use. We also compared our models to those used by popular websites, such as ProovStation, which focuses on providing fast picture-taking in light settings. Our solution sacrifices some accuracy for speed while still meeting minimum standards to ensure accurate results.

3.6.1. Dataset 1. This section presents a performance comparison of the DSMLR dataset against the five algorithms (Mask R-CNN, GCNet, PANet, CBNet, and HTC) reported by Pasupa et al. against the three algorithms (YOLACT, SipMask, and SipMask++) compared in this study. As shown in Table 1, YOLACT presented a marked improvement in the overall object detection accuracy with mAP of 61.3 for object detection against HTC accuracy of 54.1. In terms of segmentation performance, YOLACT outperformed HTC with mAP of 66.5, whereas GCNet performed better on $\mathrm{AP}^{\cdot}50$ with 78.2. In general, the AP^{s} values are reported to have the worst performing accuracies, where YOLACT performed slightly better (42.6) than PANet (38.5).

3.6.2. Dataset 2. The performance output of Dataset 2 is shown in Table 2, which includes mAP and AP at various thresholds with the ResNet-101 backbone. As shown in the table, SipMask++ showed the highest mAP of 0.57 along with the best outcomes for $\mathrm{AP}^{\mathrm{IoU}=.50}$ and $\mathrm{AP}^{\mathrm{IoU}=.75}$ of 0.72 and 0.67, respectively, for object detection and 0.65 and 0.44, for instance, segmentation categories. YOLACT was identified to be a better performer for AP^{s} with 0.44 and 0.026 for object detection and segmentation categories, respectively

Figure 9 presents a visual comparison of the detection and segmentation results with each of the three algorithms (YOLACT, SipMask, and SipMask++) shown row-wise from top to bottom. For Car A, all three algorithms failed to segment the left and rear doors. At straighter angles (Car B), the door classes were detected correctly; however, YOLACT and SipMask missed the back fender detection and segmentation. In general, SipMask++ showed more instance segmentation resilience for both smaller and larger parts, with some masking overlaps/inaccuracies in (e) along with low part localization accuracy. Part localization (detection) yielded the best results for all three cars using SipMask (g-i). In terms of overall computational efficiency, SipMask++ again took the lead with an average of 17.5 fps, followed by the second-best performer, SipMask with 20.8 fps. The consensus on these results was that SipMask++ showed better localization/detection and mAP of 0.57. This was followed by YOLACT with mAP of 0.49 and 0.41 for detection and segmentation, respectively. YOLACT was observed to handle AP^{s} better for both detection and segmentation, as explained in Table 3.

3.6.3. Dataset 3. The performance on the merged classes' dataset results is shown in Table 4. A significant improvement can be noticed by all aspects, including the bbox and mAP scores. The outcomes showed that the SipMask genre outperforms the YOLACT algorithm except in the case of smaller bounding boxes. YOLACT and SipMask adjusted far better at the introduction of augmentation and class adjustment measures.

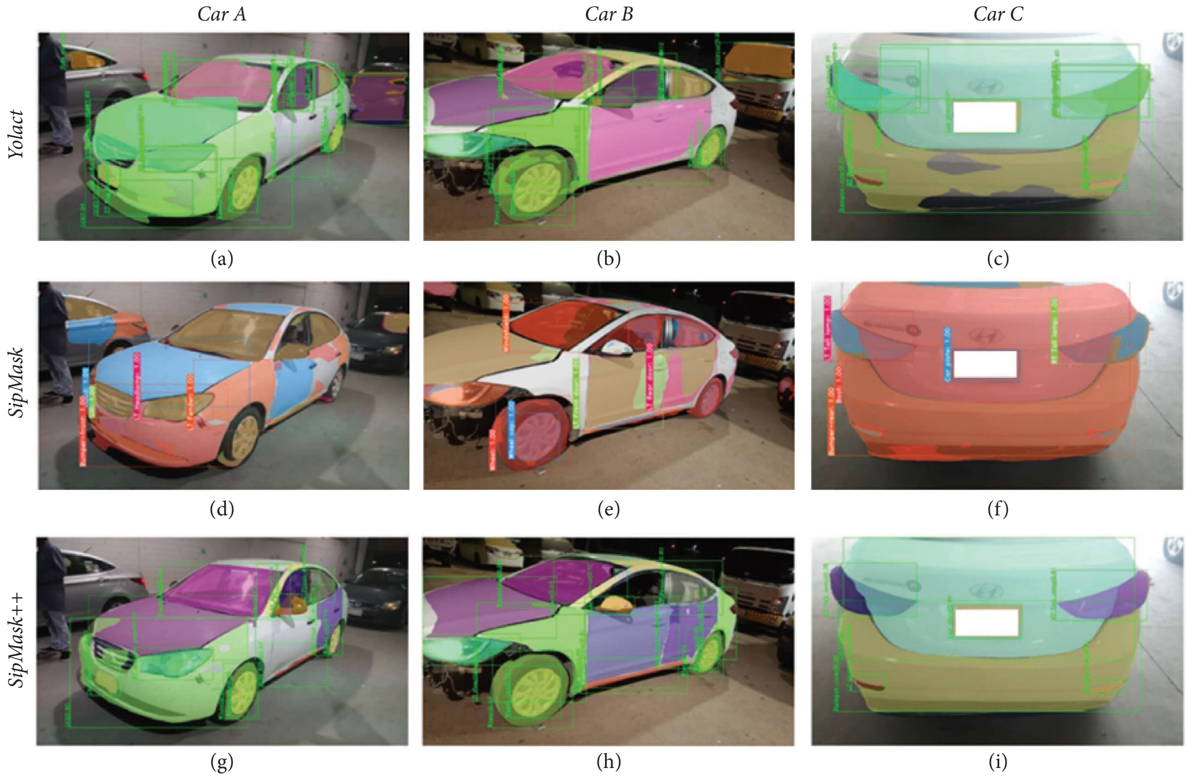
3.7. Class-Level Performance for YOLACT. Further examination of class-level mAP for the YOLACT algorithm showed a marked performance loss in classes that were either very diverse (e.g., the glass class, owing to its varied shapes and sizes as well as its reflectivity) or had smaller cross sections, such as the pillar or side skirt class (Figure 10. These three classes had mAPs of 20.3, 21.9, and 27.6, respectively, compared with classes that had a higher likelihood of being similar for different models, such as the Hood, Bumper-rear, and Boot classes with mAPs (0.775, 0.739, and 0.734). The Glass class was under-represented in the dataset; however, for the remaining two classes, regardless of their abundance in the dataset, their masks were outliers to the remaining parts compared to their bounding boxes, which led to lower small-object mAPs. In general, larger and spatially similar objects showed better mAP than smaller or thinner parts. The same observation holds true for both detection and segmentation outcomes; hence, the analysis

Table 2: A performance comparison ranking with the DSMLR dataset and the performance reported in Pasupa et al. [60] against the algorithms reported in this study SipMask++, YOLACT-550, SipMask, and YOLACT.

Models	Object detection						Instance segmentation					
	mAP	AP$^{.50}$	AP$^{.75}$	AP^{s}	AP^{m}	AP^{l}	mAP	AP$^{.50}$	AP$^{.75}$	AP^{s}	AP^{m}	AP^{l}
SipMask++	51.1	69.2	58	33.9	41	58.4	55	77	59	32.1	**54**	59.2
SipMask	49.7	71.6	60.5	30.3	48.5	59.8	53.7	71.3	62.1	30.7	48.1	61.2
YOLACT	**61.3**	61.3	60.8	**39.5**	**60.5**	**63.1**	**66.5**	67.1	58.8	**42.6**	44.5	**65.7**
MaskR-CNN	51.1	69.2	58.0	33.9	41.0	58.4	55.0	77.0	59.0	32.1	**54.0**	59.2
GCNet	50.9	**76.8**	57.7	32.3	45.6	56.7	54.6	**78.2**	63.9	34.9	48.3	61.7
PANet	48.8	76.5	56.4	32.9	48.6	51.8	54.0	77.3	63.5	38.5	51.4	58.7
CBNet	51.9	71.6	60.8	28.6	48.3	57.9	53.0	72.2	63.0	28.6	48.3	61.7
HTC	54.1	75.7	**63.6**	34.4	50.4	61.1	55.2	76.1	**65.2**	36.1	50.2	63.6

Table 3: A performance comparison ranking of SipMask++, SipMask, and YOLACT instance segmentation performance against the workshop repair dataset with merged, 22-class Dataset 2.

		Object detection						Instance segmentation						
Rank	Models	mAP	AP$^{.50}$	AP$^{.75}$	mAP^{s}	mAP^{m}	mAP^{l}	mAP	AP$^{.50}$	AP$^{.75}$	mAP^{s}	mAP^{m}	mAP^{l}	fps
1	SipMask++	**57**	**72**	**67**	39	**64**	**71**	**43**	**65**	**44**	2	**12**	**47.2**	**17.5**
3	SipMask	44	58	51	29	50	59	39.7	62.1	42	0.7	11.1	43.8	20.8
2	YOLACT	49	63	53	**44**	55	66	41.2	**65**	43	**2.6**	10.5	45.7	21.1



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 9: Comparing YOLACT to SipMask/SipMask++ instance segmentation regimes. (a–c) YOLACT, (d–f) SipMask, and (g–i) SipMask++.

was further extended to three randomly selected examples (Car A, Car B, and Car C), as shown in Figure 11 with the detailed mAPs shown in Table 5. The table shows valid mAPs as green cells, with any misclassified cells in a pair (e.g., wheel cap) shown as amber. Any undetected or incorrectly classified classes are indicated by red cells. A further precision/recall analysis of the three cars against the three algorithms shows a marked difference between the YOLACT and SipMask genres, with the latter category clearly outperforming the bounding box scales (Table 6).

Table 4: A performance comparison ranking of SipMask++, YOLACT-550, SipMask, and YOLACT instance segmentation performance against the workshop repair dataset.

		Object detection						Instance segmentation						
Rank	Models	mAP	$AP^{.50}$	$AP^{.75}$	mAP^{s}	mAP^{m}	mAP^{l}	mAP	$AP^{.50}$	$AP^{.75}$	mAP^{s}	mAP^{m}	mAP^{l}	fps
2	SipMask++	55.9	84.4	63	2.5	26,6	60.8	50.6	77.5	54.9	2	17.7	56	
1	SipMask	**58.8**	**85.9**	**66**	2.3	**28.3**	**63.8**	**53.3**	**80.2**	**57.3**	1.1	19.2	**59.3**	
3	YOLACT	44.6	80	45.7	**8.4**	24.1	47.1	49.2	74.6	53.5	**2.8**	**24.5**	53	

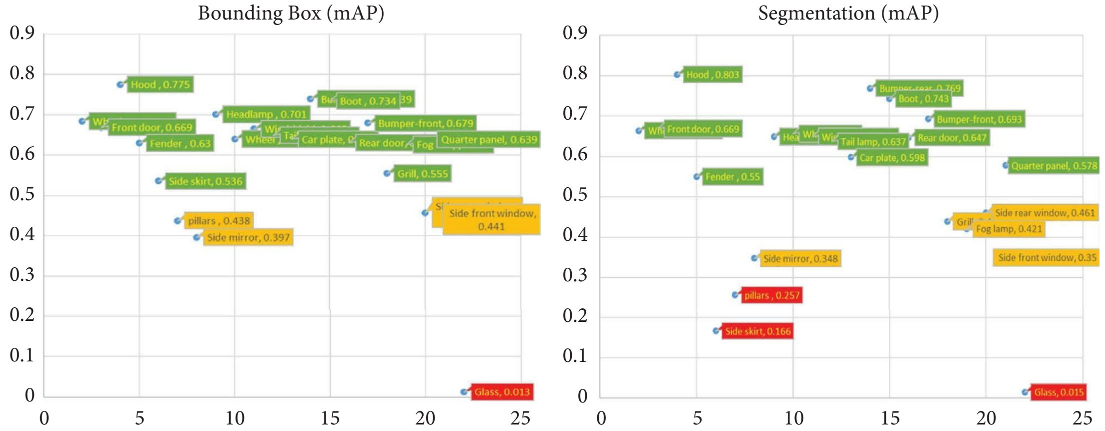
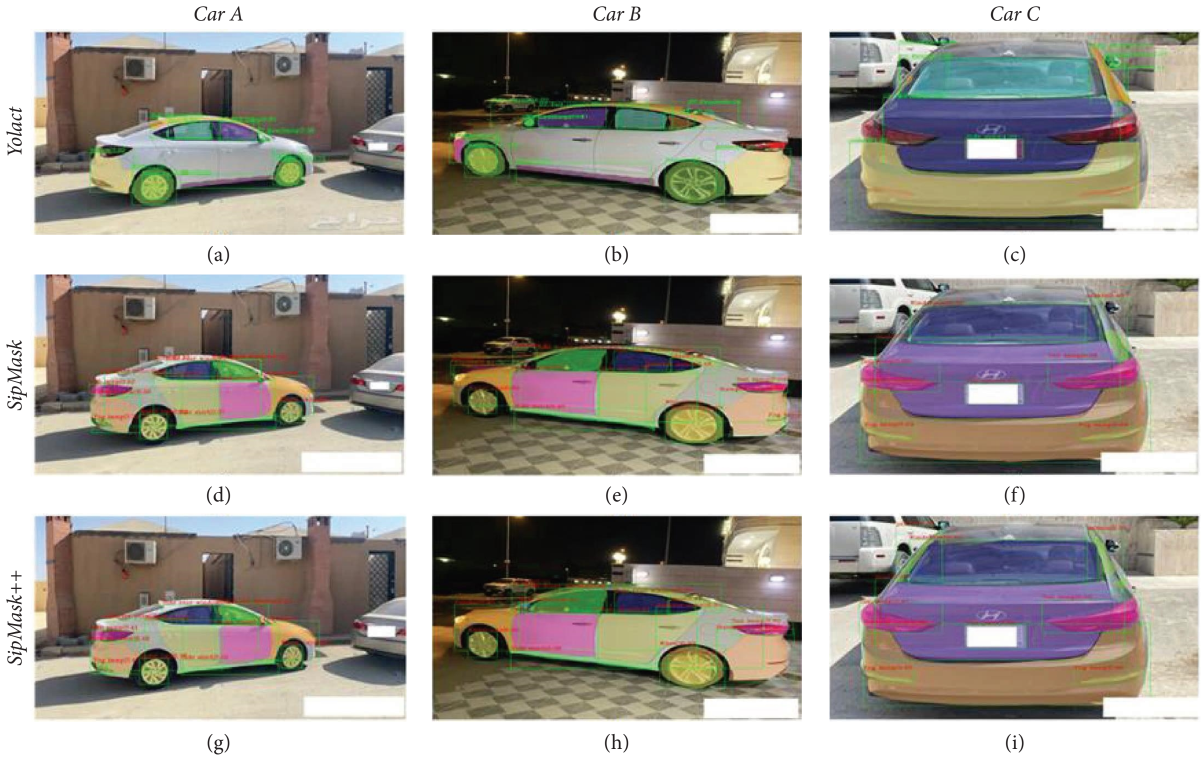


Figure 10: Class-level mAP values for the YOLACT algorithm.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 11: Comparing YOLACT to SipMask/SipMask++ instance segmentation regimes on a merged-classes (22) dataset with an increased sample size and augmentation modules: (a–c) YOLACT, (d–f) SipMask, and (g–i) SipMask++.

Table 5: A part-level bounding box analysis of Dataset-3 performance. The numbers represent the score of one or more parts identified correctly. **NV** means not visible. **ND** means not detected (false negative) and the remaining is the parts names.

	Bounding box			Bounding box			Bounding box		
	Car A			Car B			Car C		
	YOLACT	SipMask	SipMask++	YOLACT	SipMask	SipMask++	YOLACT	SipMask	SipMask++
Wheel cap (P)	0.87	0.6	0.69	0.5	1	0.58	NV	NV	NV
Front door	FN	0.61	0.52	FN	0.62	0.58	NV	NV	NV
Hood	NV	NV	NV	FN	0.31	FN	NV	NV	NV
Fender	FN	0.5	0.39	FN	0.41	0.35	NV	NV	NV
Side skirt	Hood	0.37	0.52	Hood	0.4	0.36	NV	NV	NV
Pillars	FN	0.32	FN	FN	FN	FN	NV	NV	NV
Side mirror	Headlamp	FN	FN	Headlamp	FN	0.31	NV	NV	NV
Headlamp	NV	NV	NV	NV	NV	NV	NV	NV	NV
Wheel (P)	1	FN	FN	1	0.35	0.35	NV	NV	NV
Windshield	NV	NV	NV	NV	NV	NV	Fender	0.55	0.62
Tail lamp	FN	0.52	0.41	FN	0.6	0.6	ND	0.54	0.5
Car plate	NV	NV	NV	NV	NV	NV	Side-skirt	0.6	0.56
Bumper-rear	Pillars	0.56	0.48	Pillars	0.37	0.57	Pillars	0.63	0.69
Boot	NV	NV	NV	NV	NV	NV	FN	0.66	0.65
Rear door	FN	0.54	0.61	FN	0.72	0.63	NV	NV	NV
Bumper front	NV	NV	NV	NV	NV	NV	NV	NV	NV
Grill	NV	NV	NV	NV	NV	NV	NV	NV	NV
Fog lamp	FN	0.39	0.42	NV	NV	NV			
Side rear window	Boot	0.49	0.47	Boot	0.53	0.56	NV	NV	NV
Quarter panel	FN	0.43	0.52	Car plate	0.48	0.57	NV	NV	NV
Glass	FN	FN	FN	FN	0.47	0.41	NV	NV	NV
Side front window	Tail-lamp	0.42	0.45	Tail-lamp	0.45	0.36	NV	NV	NV
Precision	3/(3 + 5)	12/13	13/13	3/3 + 5	13/15	14/15	1/3	5/5	5/5
Recall	3/12	12/17	13/17	3/11	13/17	14/17	1/3	5/5	5/5

Table 6: Precision/recall analysis of part-level accuracies for Cars A, B, and C for the three algorithms including YOLACT, SipMask, and SipMask++.

	Car A			Car B			Car C		
	YOLACT	SipMask	SipMask++	YOLACT	SipMask	SipMask++	YOLACT	SipMask	SipMask++
Precision	0.38	0.92	1.00	0.38	0.87	0.93	0.33	1.00	1.00
Recall	0.25	0.71	0.76	0.27	0.76	0.82	0.33	1.00	1.00

4. Conclusion and Future Work

This research applied several real-time instance segmentation methodologies to a car part segmentation system. There are many practical applications for this technology, such as on-road assessments of vehicle damage, car rental agencies, and remote vehicle repair quotations. To evaluate the performance of these methods, we used three car parts datasets, one publicly available and the others collected internally. We compared one-stage and two-stage instance segmentation models and found that the one-stage models were robust and performed comparably to the more complex two-stage models. The underlying models were generalized to mimic the random and noise-prone nature of real-world vehicular damages. The research also took into account various factors, such as image size, light intensity, merging side-classes, distortion, and image rotation/flipping.

In the future, we plan to address issues, such as underrepresented classes, localization of small objects such as side mirrors, and the specific needs of different car brands and types such as SUVs. This will involve the use of synthetic datasets to replicate cars with different backgrounds and settings, as well as the exploration of transformer-based models. Our goal is to create a more reliable and efficient instance segmentation model that can capture relevant information about surrounding objects and improve overall performance. We also plan to continue exploring tiny object detection literature and consider transfer learning with other relevant datasets. These efforts will support the use of this research in car appraisals.

Data Availability

The data used to support the findings of this study belong to a private company called Elm and can be licensed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the research department at Elm Company.

References

[1] M. Mazzetto, M. Teixeira, É. O. Rodrigues, and D. Casanova, "Deep Learning Models for Visual Inspection on Automotive Assembling Line," 2020, https://arxiv.org/abs/2007.01857.

[2] Z. Qu, J. Shen, R. Li, J. Liu, and Q. Guan, "Partsnet: a unified deep network for automotive engine precision parts defect detection," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, pp. 594–599, Shenzhen, China, December 2018.

[3] A. A. Shvets, R. Alexander, A. A. Kalinin, and I. Vladimir, "Automatic instrument segmentation in robot-assisted surgery using deep learning," in *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 624–628, IEEE, Orlando, FL, USA, December 2018.

[4] S. Lindgren Belal, M. Sadik, R. Kaboteh et al., "Deep learning for segmentation of 49 selected bones in ct scans: first step in automated pet/ct-based 3d quantification of skeletal metastases," *European Journal of Radiology*, vol. 113, pp. 89–95, 2019.

[5] O. Mohamed, C. Lassner, G. Pons-Moll, P. Gehler, and S. Bernt, "Neural body fitting: unifying deep learning and model based human pose and shape estimation," in *Proceedings of the 2018 International Conference on 3D Vision (3DV)*, pp. 484–494, IEEE, Verona, Italy, September 2018.

[6] M. Mittal, L. M. Goyal, S. Kaur, I. Kaur, A. Verma, and D. Jude Hemanth, "Deep learning based enhanced tumor segmentation approach for mr brain images," *Applied Soft Computing*, vol. 78, pp. 346–354, 2019.

[7] P. M. Shakeel, M. Burhanuddin, and M. I. Desa, "Lung cancer detection from ct image using improved profuse clustering and deep learning instantaneously trained neural networks," *Measurement*, vol. 145, pp. 702–712, 2019.

[8] J. Champ, A. Mora-Fallas, H. Goëau, E. Mata-Montero, P. Bonnet, and A. Joly, "Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots," *Applications in plant sciences*, vol. 8, no. 7, Article ID e11373, 2020.

[9] A. Khan, T. Ilyas, M. Umraiz, Z. I. Mannan, and H. Kim, "Ced-net: crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture," *Electronics*, vol. 9, no. 10, 2020.

[10] P. Bosilj, T. Duckett, and G. Cielniak, "Connected attribute morphology for unified vegetation segmentation and classification in precision agriculture," *Computers in Industry*, vol. 98, pp. 226–240, 2018.

[11] T. Hoeser and C. Kuenzer, "Object detection and image segmentation with deep learning on earth observation data: a review-part i: evolution and recent trends," *Remote Sensing*, vol. 1667, no. 10, p. 1667, 2020.

[12] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 94–114, 2020.

[13] T. Le and Y. Duan, "Pointgrid: a deep network for 3d shape understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9204–9214, Salt Lake City, UT, USA, June 2018.

[14] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: Similarity group proposal network for 3D point cloud instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2569–2578, Salt Lake City, UT, USA, June 2018.

[15] R. Prophet, A. Deligiannis, J. C. Fuentes-Michel, I. Weber, and M. Vossiek, "Semantic segmentation on 3d occupancy grids for automotive radar," *IEEE Access*, vol. 8, pp. 197917–197930, 2020.

[16] J. J. Rubio, T. Kashiwa, T. Laiteerapong et al., "Multi-class structural damage segmentation using fully convolutional networks," *Computers in Industry*, vol. 112, Article ID 103121, 2019.

[17] H. Wang, Y. Li, L. M. Dang, S. Lee, and H. Moon, "Pixel-level tunnel crack segmentation using a weakly supervised annotation approach," *Computers in Industry*, vol. 133, Article ID 103545, 2021.

[18] D. Mazzini, P. Napoletano, F. Piccoli, and R. Schettini, "A novel approach to data augmentation for pavement distress segmentation," *Computers in Industry*, vol. 121, Article ID 103225, 2020.

[19] K. S. Braunwarth, M. Kaiser, and A. L. Müller, "Economic evaluation and optimization of the degree of automation in insurance processes," *Business and Information Systems Engineering*, vol. 2, no. 1, pp. 29–39, 2010.

[20] A. G. Howard, M. Zhu, and B. Chen, "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[21] X. Ran, H. Chen, X. Zhu, Z. Liu, and J. Chen, "Deepdecision: a mobile deep learning framework for edge video analytics," in *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1421–1429, IEEE, Honolulu, HI, USA, April 2018.

[22] R. Girshick, J. Donahue, D. Trevor, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[23] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Columbus, OH, USA, June 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[25] W. Liu, D. Anguelov, and D. Erhan, "Ssd: single shot multibox detector," in *Proceedings of the European conference on computer vision*, pp. 21–37, Springer, Amsterdam, The Netherlands, October 2016.

[26] R. Joseph, S. Divvala, R. Girshick, and F. Ali, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[27] Z. Ma, Y. Li, M. Huang, Q. Huang, J. Cheng, and S. Tang, "A lightweight detector based on attention mechanism for aluminum strip surface defect detection," *Computers in Industry*, vol. 136, Article ID 103585, 2022.

[28] M. Liu, J. Xie, J. Hao, Y. Zhang, X. Chen, and Y. Chen, "A lightweight and accurate recognition framework for signs of x-ray weld images," *Computers in Industry*, vol. 135, Article ID 103559, 2022.

[29] A. S. Mangat, J. Mangler, and S. Rinderle-Ma, "Interactive process automation based on lightweight object detection in manufacturing processes," *Computers in Industry*, vol. 130, Article ID 103482, 2021.

[30] M. Xiao, B. Yang, S. Wang, Z. Zhang, X. Tang, and L. Kang, "A feature fusion enhanced multiscale cnn with attention mechanism for spot-welding surface appearance recognition," *Computers in Industry*, vol. 135, Article ID 103583, 2022.

[31] M. Chen, L. Yu, C. Zhi et al., "Improved faster r-cnn for fabric defect detection based on gabor filter with genetic algorithm optimization," *Computers in Industry*, vol. 134, Article ID 103551, 2022.

[32] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Computers in Industry*, vol. 108, pp. 115–120, 2019.

[33] H. Kang and S. Kang, "A stacking ensemble classifier with handcrafted and convolutional features for wafer map pattern classification," *Computers in Industry*, vol. 129, Article ID 103450, 2021.

[34] R. Chen, X. Huang, L. Yang, X. Xu, X. Zhang, and Y. Zhang, "Intelligent fault diagnosis method of planetary gearboxes based on convolution neural network and discrete wavelet transform," *Computers in Industry*, vol. 106, pp. 48–59, 2019.

[35] J. U. Ko, J. H. Jung, M. Kim, H. B. Kong, J. Lee, and B. D. Youn, "Multi-task learning of classification and denoising (MLCD) for noise-robust rotor system diagnosis," *Computers in Industry*, vol. 125, Article ID 103385, 2021.

[36] M. L. Hoffmann Souza, C. A. da Costa, G. de Oliveira Ramos, and R. da Rosa Righi, "A feature identification method to explain anomalies in condition monitoring," *Computers in Industry*, vol. 133, Article ID 103528, 2021.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, October 2017.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[41] C. Szegedy, W. Liu, and Y. Jia, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.

[42] F. N. Iandola, S. Han, and M. W. Moskewicz, "Squeezenet: alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," 2016, https://arxiv.org/abs/1602.07360.

[43] B. Davis, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[44] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: real-time instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9157–9166, Seoul, South Korea, November 2019.

[45] J. Cao, R. M. Anwer, and H. Cholakkal, "Sipmask: spatial information preservation for fast image and video instance segmentation," in *European Conference on Computer Vision*, pp. 1–18, Springer, Berlin, Germany, 2020.

[46] Y. Syed Adnan, A. A. Abdulmalik, and R. Souissi, "Automotive parts assessment: applying real-timeinstance-segmentation models to identify vehicle parts," 2022, https://arxiv.org/abs/2202.00884.

[47] C. Ge, J. Wang, J. Wang, Q. Qi, H. Sun, and J. Liao, "Towards automatic visual inspection: a weakly supervised learning method for industrial applicable object detection," *Computers in Industry*, vol. 121, Article ID 103232, 2020.

[48] H. Qi, T. Xu, G. Wang, Y. Cheng, and C. Chen, "Myolov3-tiny: a new convolutional neural network architecture for real-time detection of track fasteners," *Computers in Industry*, vol. 123, Article ID 103303, 2020.

[49] J. Božič, D. Tabernik, and D. Skočaj, "Mixed supervision for surface-defect detection: from weakly to fully supervised learning," *Computers in Industry*, vol. 129, Article ID 103459, 2021.

[50] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[51] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, Anchorage, AK, USA, June 2008.

[52] S. Jasper, O. Rippel, and S. Kevin, "Scalable bayesian optimization using deep neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 2171–2180, PMLR, Lille, France, July 2015.

[53] A. Gholami, S. Kim, and Z. Dong, "A survey of quantization methods for efficient neural network inference," 2021, https://arxiv.org/abs/2103.13630.

[54] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[55] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks," *Journal of Machine Learning Research*, vol. 22, no. 241, pp. 1–124, 2021.

[56] K. Pasupa, P. Kittiworapanya, N. Hongngern, and K. Woraratpanya, "Evaluation of deep learning algorithms for semantic segmentation of car parts," *Complex & Intelligent Systems*, vol. 8, no. 5, pp. 3613–3625, 2021.

[57] A. Tabb and H. Medeiros, "Automatic segmentation of trees in dynamic outdoor environments," *Computers in Industry*, vol. 98, pp. 90–99, 2018.

[58] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact++: better real-time instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 1108–1121, 2022.

[59] G. Ghiasi, Y. Cui, and A. Srinivas, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, Montreal, Canada, June 2021.

[60] K. Pasupa, P. Kittiworapanya, N. Hongngern, and K. Woraratpanya, "Evaluation of deep learning algorithms for semantic segmentation of car parts," *Complex & Intelligent Systems,*, pp. 1–13, 2021.