electronics

MDPI

Article

Road Object Detection: A Comparative Study of Deep Learning-Based Algorithms

Malik Haris [1,*] and Adam Glowacz [2]

[1] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[2] Department of Automatic Control and Robotics, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Science and Technology, 30-059 Kraków, Poland; adglow@agh.edu.pl
* Correspondence: malikharis@hotmail.com

Abstract: Automated driving and vehicle safety systems need object detection. It is important that object detection be accurate overall and robust to weather and environmental conditions and run in real-time. As a consequence of this approach, they require image processing algorithms to inspect the contents of images. This article compares the accuracy of five major image processing algorithms: Region-based Fully Convolutional Network (R-FCN), Mask Region-based Convolutional Neural Networks (Mask R-CNN), Single Shot Multi-Box Detector (SSD), RetinaNet, and You Only Look Once v4 (YOLOv4). In this comparative analysis, we used a large-scale Berkeley Deep Drive (BDD100K) dataset. Their strengths and limitations are analyzed based on parameters such as accuracy (with/without occlusion and truncation), computation time, precision-recall curve. The comparison is given in this article helpful in understanding the pros and cons of standard deep learning-based algorithms while operating under real-time deployment restrictions. We conclude that the YOLOv4 outperforms accurately in detecting difficult road target objects under complex road scenarios and weather conditions in an identical testing environment.

Keywords: autonomous vehicles; intelligent transportation system (ITS); object detection; deep learning

check for updates

Citation: Haris, M.; Glowacz, A. Road Object Detection: A Comparative Study of Deep Learning-Based Algorithms. *Electronics* **2021**, *10*, 1932. https://doi.org/10.3390/electronics10161932

Academic Editor: Daniel Morris

Received: 21 June 2021
Accepted: 9 August 2021
Published: 11 August 2021
Retracted: 7 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The automobile industries have developed rapidly since the first demonstration in the 1980s [1], the vehicle navigation and intelligence system have improved. However, the increase in road vehicles raises traffic congestion, road safety, pollution, etc. Autonomous driving is a challenging task; a small error in the system can lead to fatal accidents. Visual data play an essential role in enabling advanced driver-assistance systems in autonomous vehicles [2]. The low cost and wide availability of vision-based sensors offer great potential to detect road incidents. Additionally, emerging autonomous vehicles use various sensors and deep learning methods to detect and classify four classes (such as a vehicle, pedestrian, traffic sign, and traffic light) to improve safety by monitoring the current road environment.

Object detection is a method of localizing and classifying an object in an image to understand the image entirely. It is currently one of the first fundamental tasks in vision-based autonomous driving. The object detection methods make bounding boxes around the detected objects and the predicted class label and confidence score associated with each bounding box. Figure 1 shows an example of an object detection method to identify and locate target objects in an image. Object detection and tracking are challenging tasks and play an essential role in many visual-based applications. At present, the deep learning field provides several methods for advancing the automation levels by improving the environment perception [3]. The autonomous vehicles have developed from Level-0 class with no automation to Level-1 with driver assistance automation. The Level-2 class with partial automation enables the vehicle to assist in steering and acceleration functionality, and the driver controls many safety-critical actions. For Level-3 class with conditional

RETRACTED

This paper has been retracted. A retraction notice was published on 7 April 2022 in *Electronics* **2022**, *11*, 1166. https://doi.org/10.3390/electronics11081166

automation, the intelligent vehicle must monitor the whole surroundings in real-time, and the driver can only take control over the vehicle when prompted by the system. However, to achieve autonomous driving in vehicles, there is a long way to reach the Level-4 class with high automation, and finally, the Level-5 class with full automation [3].

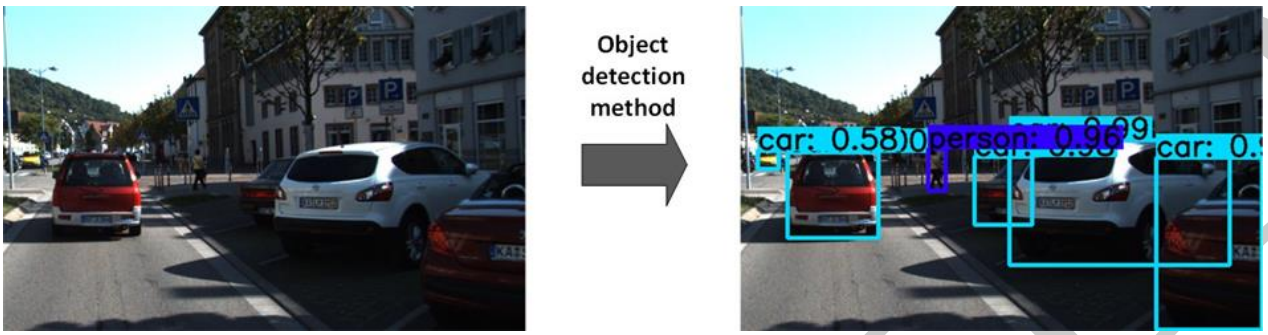


Figure 1. (**left**) Objects before detection; (**right**) Objects after detection show the detected object's bounding box coordinates with the predicted confidence score and class label.

The sensors used to detect and monitor vehicles may be identified as containing three components: the transducer, the unit for signal processing, and the device for data processing. In an autonomous vehicle, all types of sensors are essential to obtain correct information about its surrounding environment. At present, sensors used in the vehicles primarily include the Monocular Camera, Binocular camera, Light Detecting and Ranging (LiDAR), Global Navigation Satellite System (GNSS), Global Positioning System (GPS), Radio Detection and Ranging (Radar), Ultrasonic sensor, Odometer, and many more [4]. However, all these sensors have their benefits and drawbacks. A LiDAR operates with a similar principle of radar, but it emits infrared light waves instead of radio waves. It has much higher accuracy than radar under 200 meters. Weather conditions such as fog or snow have a negative impact on the performance of LiDAR. Another aspect is the sensor size: smaller sensors are preferred on the vehicle because of limited space and aerodynamic restraints, and LiDAR is generally larger than radar, stereo camera [5], flash camera [6], event camera [7], and thermal camera [8,9]. However, researchers work on reducing the cost, size, and weight of LiDAR recently [10], but it still needs more work. However, in this paper, we are working on the enhancement of visual data by using the camera.

Furthermore, the radar only detects objects in close range. Therefore, the radar may detect objects at less than their specified range if a vehicle moves faster. Furthermore, both binocular cameras and monocular cameras may produce worse detection results in low light. The GPS chip is a power-hungry device that drains the battery rapidly and is costly. The GPS may also have inaccuracy due to environmental interference. Sensors are mainly used to perceive the environment, including dynamic and static objects, e.g., drivable areas, buildings, pedestrian crossings, Cameras, LiDAR, Radar, and Ultrasonic sensors are the most commonly used modalities for this task. A detailed comparison of sensors is given in Table 1.

Early autonomous driving systems heavily relied on sensory data for accurate environment perception. Several instrumented vehicles are introduced by different research groups, such as Stanford's Junior [11], which employs various sensors with different modalities for perceiving external and internal variables. Boss won the DARPA Urban Challenge with an abundance of sensors [12]. RobotCar [13] is a cheaper research platform aimed at data collection. In addition, different levels of driving automation have been introduced by the industry; Tesla's Autopilot [14] and Google's self-driving car [15] are some examples.

RETRACTED

Table 1. Sensor Details.

Modality	Affected by Illumination	Affected by Weather	Color	Depth	Range	Accuracy	Size	Cost
LiDAR	-	√	-	√	medium (<200 m)	high	large	high
Radar	-	-	-	√	high	medium	small	medium
Ultrasonic	-	-	-	√	short	low	small	low
Camera	√	√	√	-	-	-	smallest	lowest
Stereo Camera	√	√	√	√	medium (<100 m)	low	medium	low
Flash Camera	√	√	√	√	medium (<100 m)	low	medium	low
Event Camera	limited	√	-	-	-	-	smallest	low
Thermal Camera	-	√	-	-	-	-	smallest	low

However, finding the coordinate of the object in the frame has become challenging due to various factors such as variations in viewpoints, poses, scales, lighting, occlusions, etc. [16]. After the development of the deep neural network, computer vision gained even more strides [17]. With the advances in sensing and computational technologies in the field of computer vision, the performance of traditional manual feature-based object detection algorithms has been compared to that of the deep learning-based object detection algorithms because of continuous growth in large volumes of data and fast development of hardware, particularly Multicore Processors and Graphical Processing Units (GPUs). Furthermore, the deep learning-based algorithms exceed the traditional algorithms in terms of detection speed and accuracy. The deep learning methods have gained much attention due to the promising results it has achieved in multiple fields, such as image classification [18], segmentation [19], and moving object detection and tracking [4]. Object counting, overtaking detection, object classification [5], lane change detection [20], emergency vehicle detection, traffic control, traffic sign, light identification, license plate recognition, and many other applications of deep learning-based detection can be found in every Intelligent Transportation System (ITS) field. Road object detection has been a hot topic for many researchers over the past decade. In the following literature: deep learning-based object detection [21], on-road vehicle detection [22], object detection, and safe navigation by Markov random field (MRF) [5] in which authors aim to analyze the deep learning-based algorithms without considering recent improvements in the deep learning field.

In contrast, various literature use datasets with low fine-grained recognition and are limited in significant aspects, discussed in the dataset description section. Furthermore, Wang et al. [23] focus on detecting a single object, namely vehicles. Furthermore, different types of deep learning detection methods have been analyzed, but only a few comparable studies test both the detection speed and accuracy on different road scenarios of different objects [24,25], so there is a lack of detailed analysis and comparison between the different trending state-of-the-art detection models. This comparative study aims to fill the gap in the literature with the primary key contributions as follows:

1. A comparative review of different aspects of the five popular and trending deep learning algorithms (R-FCN [26], Mask R-CNN [27], SSD [28], RetinaNet [29], and YOLOv4 [30]) from many object detection algorithms with their key contributions on popular benchmarks are presented.
2. The primarily-used deep learning-based algorithms for road object detection are compared on a new diverse and large-scale Berkeley Deep Drive (BDD100K) dataset [31].
3. The results are analyzed and summarized to show their performance in terms of detection speed and accuracy. The generalization ability of each model is shown under different road environmental conditions at different times of day and night. The parameters are chosen to ensure the credibility of experimental results.

RETRACTED

4. Significant guidance for future research and the development of new trends in the field of deep learning-based object detection are included in this work.

2. Related Work

In general, over the past two decades, the field of object detection has been through two periods: Traditional object detection algorithms (before 2012) and Deep Learning-based object detection algorithms (since 2012).

2.1. Traditional Object Detection Algorithms

Object detection using computer vision has been a challenging task in the literature for over two decades. Many objects under different road conditions in dynamic environments become difficult to identify, so detecting these images captured by the camera is always challenging. In early studies, several extracted features using Artificial Intelligence (AI) were used to detect objects. The traditional feature-based object detection algorithms include three primary modules; first is informative region selection, second is feature extraction, and last is classification. The most common traditional based feature extraction models in the literature for road object detection are Haar [32] and the histogram of oriented gradients (HOG) [33].

HOG features are extracted in two steps: evaluating edge operators over the image, discretizing, and discarding edge intensities' orientations into the histogram. The limitation of this method is high time consumption through it achieved good performance for pedestrian detection. The HOG feature vector combined with the Support Vector Machine (SVM) classifier has been commonly used to detect the object orientation. The HOG-SVM [34] has achieved great success in object detection. Haar features are calculated by sums and differences of rectangles over an image patch. It was best suited for real-time detection as it was highly efficient in computing the symmetric structure of detecting objects. A combination of Haar [32] and HOG [33] was used for object detection and tracking. The Haar feature vector and the AdaBoost [35] were popular in the computer vision field to detect various feature applications, including face, person, vehicle, and many more. However, the limitation of Haar cascades is notoriously prone to false-positive—the Viloa–Jones algorithm can easily report a face in an image when no face is present.

2.2. Deep Learning-Based Object Detection Algorithms

Over the past decade, the field of deep learning has attracted much attention from many researchers, and several deep learning-based algorithms have emerged in recent years. The traditional detection methods require manual extraction of features by experts over time. However, the deep learning methods require huge volumes of data to learn feature characteristics over time. In computer vision, the visual recognition done by the Convolutional Neural Network (CNN) feature extraction is close to that of human visual mechanism, representing a complete process from the edge to the part of an object. Because of the continuous rise in data and fast hardware advancement, particularly the GPUs (mainly used for parallel processing), the deep learning-based algorithms for object detection have better performance than traditional detection algorithms. This gave recognition to deep learning approaches to the industry worldwide. Focusing on the real-time performance in terms of accuracy and speed in the academic field, the deep learning-based algorithms for object detection are developed in two directions: two-stage object detection algorithms and one-stage object detection algorithms [36], emphasizing the detection accuracy and detection speed, respectively.

Several deep learning-based algorithms have emerged in recent years that may consider the object specificity on the road, such as dynamic road environments, background lighting conditions, position, dimension, color, shape, size, etc. Autonomous vehicles have high requirements for the real-time detection of road targets. This article focuses on five independent mainstream deep learning-based object detection algorithms from many deep learning-based algorithms based on significant improvements proposed by the model to

RETRACTED

provide comprehensive and comparative research on the analysis of speed and accuracy performance results.

2.2.1. Two-Stage Object Detections Models

The two-stage detectors are also referred to as region-based methods. In this type of object detection model, the processing of an image is carried in two steps. A series of sparse candidate frames are generated in the first step, i.e., region proposals are extracted from a scene. The candidate frames are verified in the second step, followed by classification and regression to improve the scores and locations. The main advantage is the high accuracy and localization of object recognition. The two-stage detector models are slower than one-stage detector models and require more complex training. This article uses R-FCN [26] and Mask R-CNN [27] as the representative algorithms of these two-stage object detection models.

R-FCN

The Region-based Fully Convolutional Networks (R-FCN) [26] algorithm was proposed by J. Dai et al. in 2016. It is based on the modified structure of the Faster R-CNN [37] to maximize the sharing of convolution parameters to address the contradiction between position sensitivity and translation invariance and increase the speed [26]. Translation variance in detecting objects and translation invariance in classifying images may cause high inconsistencies. The R-FCN addresses this problem partially under the same detection precision, but the detection speed is still much inferior to the one-stage algorithms. The architecture of R-FCN is shown in Figure 2. The R-FCN uses the convolution layers to produce the position-sensitive score maps, such as size $k \times k \times (C + 1)$, where $(C + 1)$ represents the total number of object classes with one additional background class, and each object class having $k \times k$ score maps. Furthermore, the R-FCN generates Region of Interest (RoI) proposals using the fully convolutional Region Pyramid Network (RPN) (position-sensitive RoI pooling layer). The different RoI regions can correspond to different positions on the score map. Each RoI region is divided into $k \times k$ subregions, each corresponding to one area of the score map [26]. This process is repeated for evaluating the score of each class. If the response value of the class match values for $k \times k$ subregions of each class, then an average of sub-regions values (average voting) is used to obtain the single object match score for each class [26]. Finally, the RoI is classified with a softmax function for the remaining $C + 1$ object class to get the probability of belonging to each class. A class-agnostic bounding box regression, named the regression score map [26], is obtained by appending another $4 \times k \times k$ convolution layer.

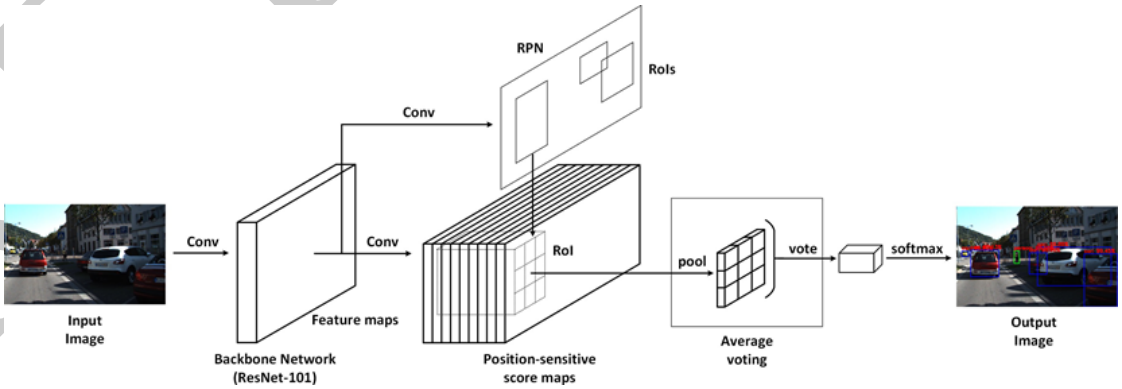


Figure 2. The architecture of the Region-based Fully Convolutional Networks (R-FCN) [26].

Experiments show that using ResNet-101 as a backbone in R-FCN achieved the mean Average Precision (*mAP*) of 31.5 on the Microsoft Common Objects in Context (MS COCO) dataset [26]. The R-FCN has the same detection precision but is much faster than Faster R-

CNN [26]. In this work, the R-FCN is compared with the other object detection algorithms, and it is found that R-FCN outperforms SSD [28] in terms of detection accuracy.

Mask R-CNN

The Mask Region-based Convolutional Neural Network's (Mask R-CNN) [27] algorithm was proposed by K. He et al. in 2017. It is an extending work of Faster R-CNN [37], primarily, for instance, the segmentation task. The Faster R-CNN [37] proposed in 2015 replaces the traditional selective search with Feature Pyramid Network (FPN) [38] to perform detection but fails to address the problem of translation invariance and position sensitivity. Instance segmentation can be classified into two parts: first is object detection, and second is semantic (pixel-level) segmentation. Mask R-CNN extracts feature proposals from scrutinized images using Faster R-CNN [37] method and then performs segmentation by object detection with simultaneous generation of bounding boxes and high-quality masks of highest Intersection over Union (IoU) prediction with an increase in the inference speed.

Mask R-CNN has two significant contributions. Firstly, it uses a more accurate RoI align module, called RoI Align [27], to replace the traditional RoI pooling module because pixel-level segmentation needs much more fine-grained alignment. The traditional RoI pooling method has poor location misalignment due to quantization. Secondly, an extra branch from the RoI Align module is inserted. This additional branch extracts a small feature map or bin from each RoI, which accepts the output of the RoI Align layer to be fed into the two convolution layers. At the last step, the final output obtained from these layers is the mask (bounding box) itself, as shown in Figure 3. The RoI Align module initially calculates the floating number of the coordinates of each RoI bin. Then a bilinear interpolation operation is performed to find the exact feature values at regularly sampled locations in each feature map or bin to solve quantization [27]. The results are then aggregated using the max or average pooling technique to find values of each feature map or bin to produce a more precise pixel-to-pixel alignment. Experiments show that using FPN with ResNeXt-101 as a backbone in Mask R-CNN achieved the *mAP* of 39.8 on the MS COCO dataset [27]. The Mask R-CNN is the recent release in the family of two-stage object detection models and overcomes many limitations of previous R-CNN algorithms. In this work, the Mask R-CNN is compared with the other object detection algorithms, and it is found that Mask R-CNN outperforms RetinaNet [29], R-FCN [26], and SSD [28] in terms of detection accuracy.

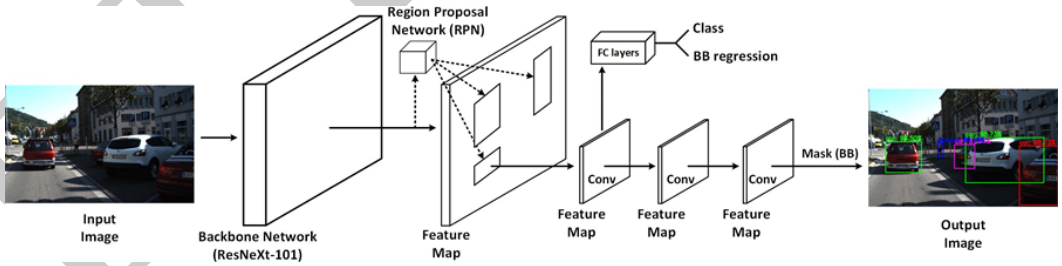


Figure 3. The architecture of the Mask Region-based Convolutional Neural Networks (Mask R-CNN) [27].

2.2.2. One Stage Object Detection Algorithms

There is no region proposal step or intermediate region process for detection in one-stage detection algorithms, and the prediction results are acquired directly from the image. In this model type, the input image is sampled at different positions uniformly using variable scales and aspect ratios, followed by sampling a CNN layer to extract features to perform classification and regression accurately. The main advantage is the high detection speed. The one-stage detection models are time efficient, easier to optimize, and more suitable for real-time devices. In this work, SSD [28], RetinaNet [29], and YOLOv4 [30] are used as the representative algorithms of this one-stage object detection model.

RETRACTED

SSD

The Single Shot Multi-Box Detector (SSD) [28] model was proposed by W. Liu et al. in 2015. In earlier approaches by R-CNN [39] and R-FCN [26], Region Pyramid Network (RPN) [29] was used to generate regions of interest, and then classification was performed on these regions using fully connected or convolution layers. The disadvantage of the R-CNN [39] algorithm is that the small-scale object detection is difficult, and their positioning is incorrect. Therefore, SSD can overcome these limitations to some extent. It performs both operations in only a single shot. It has two significant contributions. Firstly, feature maps of different sizes (10×10, 5×5, 3×3, etc.) are extracted from the scene where small-scale feature maps are generally used to detect large objects and large-scale feature maps are generally used to detect small objects (multi-reference detection). Secondly, anchor boxes of different scales and aspect ratios are generated (multi-resolution detection). Another biggest advantage of SSD is improving the performance in low-resolution images, which helps us not use costly sensors. VGG-16 [40] is used as a feature extraction network in SSD. On top of the VGG-16 [40] network six more convolution layers are added for detection, as shown in Figure 4. If an image and a set of truth or object labels are given as an input to the SSD, it produces a sequence of feature maps of different scales, followed by a 3×3 convolution filter on each feature map to produce default bounding boxes. As the image processes, the bounding box offset and the class probabilities are predicted simultaneously for each box. SSD runs detection only at the top layer to produce the best-predicted bounding box and label. Experiments show that using VGG-16 as a basic feature extraction network, the SSD512 (input image size: 512×512) algorithm achieved a *mAP* of 28.8 on the MS COCO dataset [28]. In this work, the detection precision of SSD is comparable to that of the two-stage detector R-FCN [26]. The SSD shows fast detection speed when compared to the other one-stage and two-stage object detection algorithms.

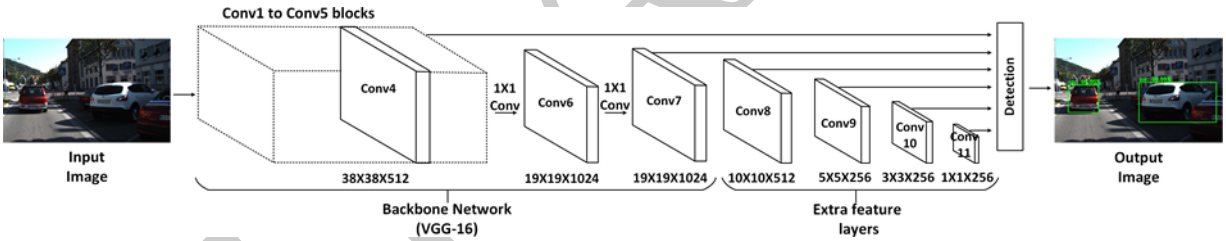


Figure 4. The architecture of the Single Shot Multi-Box Detector (SSD) [28].

RetinaNet

The one-stage detectors are known for their simplicity and high speed, but the accuracy of one-stage detection models is always lower than two-stage detection models. After identifying possible reasons behind it, T.Y. Lin et al. proposed RetinaNet [29] in 2017. The one-stage models have a higher magnitude than two-stage models because of the lack of region proposals. During training, the initial stage filters out many negative locations and proposes a small number of candidate locations. This is mainly because of the extreme foreground-background class imbalance encountered during the training of dense detectors [29]. To solve this problem, the authors propose a new loss function called Focal Loss [29] to replace the binary cross-entropy from previous works [26,28,39]. The loss of simple samples covers difficult cases by adding an estimated value of weight to each sample. After adding the value, the predicted weight is the probability of the network for classifying the sample as simple or difficult. For an effective classification, the weights of simple samples must be reduced relatively by increasing the weights of hard samples. The focal loss function focuses more on misclassified hard training examples and avoids many easy examples by balancing these weights, thereby increasing the accuracy of the model so that the one-stage detector models can perform better as compared to two-stage models while maintaining high detection speed. RetinaNet uses ResNet [40] as a backbone and FPN [38] as a feature extraction network. The RetinaNet model has two phases, as shown

RETRACTED

in Figure 5. In the first phase, a sparse set of region proposals are generated using FPN, and in the second phase, each candidate location is classified.

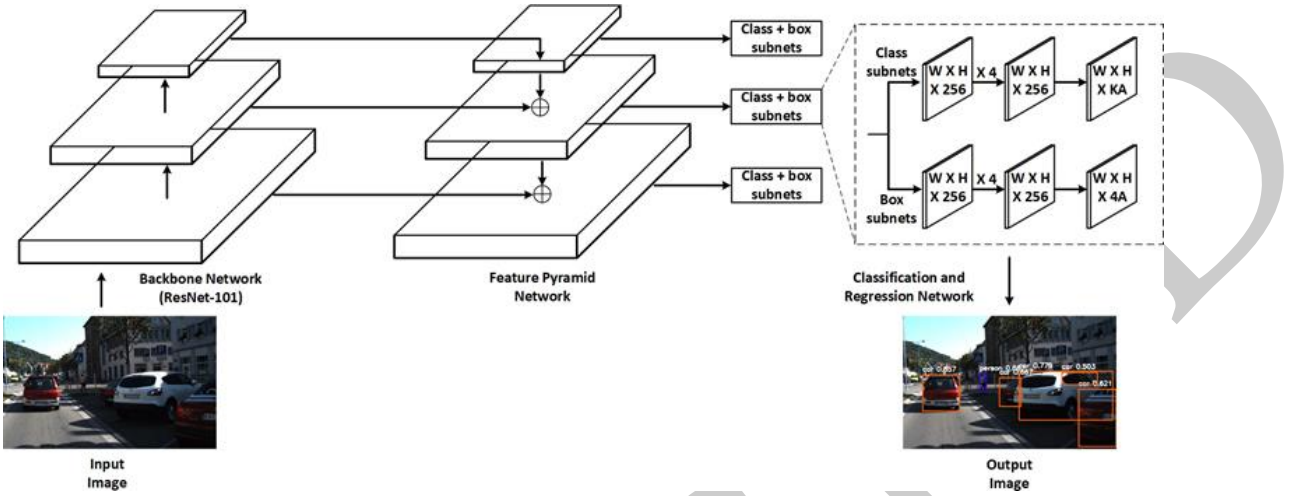


Figure 5. The architecture of the RetinaNet [29].

Experiments show using the Focal Loss as a classification function and ResNet-101-FPN as a backbone; the RetinaNet model achieved a *mAP* of 39.1 on the MS COCO dataset [29]. In this work, the RetinaNet is compared with the other object detection algorithms, and it is found that RetinaNet outperforms R-FCN [26] and SSD [28] in terms of detection accuracy.

YOLOv4

The You Only Look Once (YOLO) algorithms [41–43] have been the most balanced object detectors in speed and accuracy. Various advancements have been suggested to overcome its previous three generations YOLO [41], YOLOv2 [42], and YOLOv3 [43] (unsatisfactory detection of small objects, low detection accuracy, etc.). Wang CY et al. proposed YOLOv4 [30] in 2020 with significant changes from the previous versions. It is the most recent and advanced release of the YO-LO family. YOLOv3 [43] uses DarkNet-53 [43] and CSPNet [43] as the main network. The Darknet-53 [43] has the same accuracy as Resnet [40] but is faster. The dense block comprises multiple convolutional layers with batch normalization, Rectified Linear Unit (ReLU). A dense net is formed consisting of multiple dense blocks with transition layers followed by convolution. Cross-Stage Partial (CSP) connections divide the input feature maps of dense blocks into two parts. One part goes as an input to the next transition layer, and the other goes through the dense block. Based on this, YOLOv4 proposes a new feature extractor, CSPDarknet-53 [30]. This extractor enhances the CNN network learning ability and improves accuracy while reducing calculation and memory cost. YOLOv4 us-es Spatial Pyramid Pooling (SPP) module along with Path Aggregation Network (PANet), is used as the neck over CSPDarknet-53 because it increases the receptive field, differentiates the most significant features, and does not causes any speed reduction. The original YOLOv3 [43] network is used as the head. A simple architecture of YOLOv4 is shown in Figure 6. Few other features in YOLOv4, such as the selection of optimal hyper-parameters while applying Genetic Algorithm (G.A.) and Cross mini-Batch Normalization (CmBN) apart from the architecture, enable the detector to be more suitable for training on a single GPU. YOLOv4 proposes two new components classified as Bag of Freebies (BoF) [30] and Bag of Specials (BoS) [30]. The BoF consists of multiple training strategies to get better accuracy without an increase in the hardware cost. BoF uses techniques, such as Complete-IoU loss, Drop Block regularization, Class label smoothing, Mosaic and Cut Mix data augmentation, Self-Adversarial Training (SAT), Cosine annealing scheduler [30], and many others to increase the flexibility of input

RETRACTED

images. This increases the robustness of the object detection model for the images obtained from 12 dynamic environments. BoS consists of methods, such as Mish and Leaky-ReLU activations, Distance-IoU post-processing, PANet path-aggregation block, SPP-block, SAM-block, and many others that may slightly increase the inference cost but provide significant improvements in speed and accuracy for real-time object detection. YOLOv4 includes many innovative ideas compared to the existing models [27,29,43], thereby showing a prominent advantage in real-time performance.

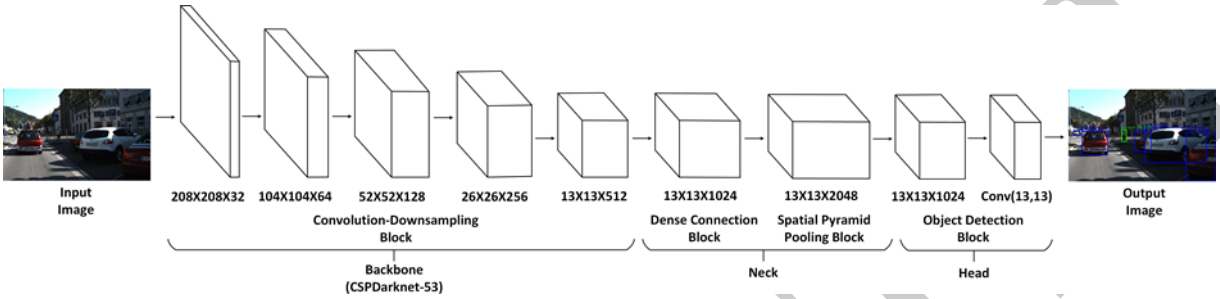


Figure 6. The architecture of the You Only Looks Once v4 (YOLOv4) [30].

The YOLOv4 algorithm achieved a *mAP* of 43.5 on the MS COCO dataset [30]. This article found that YOLOv4 is currently the best choice for road object detection with exceptionally good detection accuracy and speed compared to the previous one-stage and two-stage detectors.

2.2.3. Summary of Two-Stage/One-Stage Object Detections Models

In summary, the deep learning methods can be classified into two types of object detectors: two-stage and one-stage. The two-stage object detection can be considered a coarse-to-fine process, while the one-stage object detection can be considered a complete-in-one-step procedure. Earlier studies [23,44–46] show analysis of deep learning-based object detection models ignoring several other essential aspects in comparison. Table 2 lists the observed strengths and limitations of two-stage and one-stage detection models along with the significant innovations the model proposes in the field of deep learning-based object detection.

Table 2. Innovations with strengths and limitations of various two-stage and one-stage deep learning models.

Model		Innovations	Strengths	Limitations
Two-Stage	R-FCN [26]	Proposes position-sensitive score maps for accurate RoI pooling Performs class-agnostic bounding box regression for effective localization	Fast inference speed Low computational RoI operations	Complicated training
	Mask R-CNN [27]	Performs instance segmentation for object detection Uses RoIAlign layer for pixel-to-pixel alignments	Provides semantic segmentation Fast inference speed	The high number of false detections Low accuracy in dynamic environments
One-Stage	SSD [28]	Uses multiscale feature map for small object detection Adopts anchor boxes of different scales and aspect ratios for accurate position	Simple model Low computing requirements	Low accuracy in dynamic environments
	RetinaNet [29]	Uses multiscale ResNet with FPN as a feature extraction tool Proposes a focal loss function to balance the adaptively weighs of different samples	Improved detection precision on small objects Stable training for class imbalance	Requirements for real-time detection
	YOLOv4 [30]	Proposes a new feature extractor CSPDarknet-53 Introduces BoF and BoS methods for detectors and backbone networks.	Fast detection accuracy Training optimized for a single GPU	Small target detection accuracy could be better

RETRACTED

3. Experimental Evaluations

3.1. Dataset Description

The selection of a dataset is considered very necessary for exploiting various tasks in real-time autonomous driving. Table 3 shows a comparison of a few different datasets available for autonomous driving applications, primarily for road object detection. The annotations include only four classes (per availability): vehicle, pedestrian, traffic sign, and traffic light. In comparison to other datasets, the BDD100K dataset [31] is selected for experimental evaluation containing 100 K images (70 K training images, 20 K testing images, and 10 K validation images) from different geographic, environmental, and weather scenarios, such as streets, tunnel, gas station, parking lot, residential, and high-ways under diverse conditions at different times of day and night in real traffic urban streets; with up to 90 objects per image. Each object has Boolean values for occlusion and truncation.

Table 3. Comparison of autonomous driving datasets for road object detection.

Dataset	Year	Images	Classes	Annotations	Weather Diversity
KITTI [47]	2012	15 K	8	0.2 M	Low
BDD100K [31]	2018	100 K	10	1.8 M	High
Apollo Scape [48]	2018	144 K	27	2.1 M	Low
Euro City [49]	2019	47 K	30	0.8 M	Medium
nuScenes [50]	2019	1.4 M	23	224 M	Medium
Argo [51]	2019	993 K	15	0.22 M	High
Waymo [52]	2020	12 M	3	0.230 M	High

This article selects the BDD100K dataset [31] due to its many images under complex road scenarios and diverse weather conditions. Additionally, the dataset provides real-world driving images to test the deep learning-based algorithms under constraints performing tasks of various complexities. A snapshot of multiple images from the BDD100K dataset [31] is shown in Figure 7.

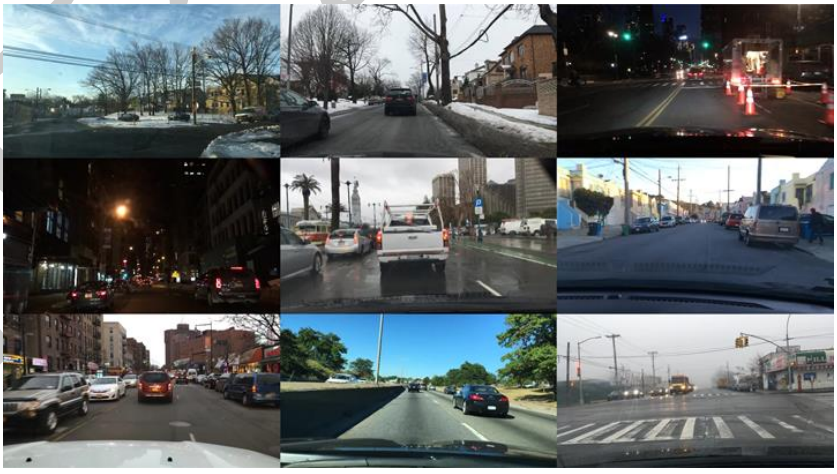


Figure 7. A snapshot of multiple images from the Berkeley Deep Drive (BDD100K) dataset.

The BDD100K dataset [31] originally contains objects, namely person, rider, bike, car, bus, truck, motor, train, traffic light, and a traffic sign. In this evaluation, the irrelevant samples are eliminated because of class imbalance and specifically four object classes that are considered most critical for an autonomous vehicle, namely vehicle (bike, car, bus, truck, and motor), a pedestrian (person and rider), traffic sign, and traffic light are retained. It is worth mentioning that few images in the dataset are invalid (incorrect label or coordinates). After resolving the number of annotated objects that are considered in this article are listed in Table 4.

RETRACTED

Table 4. The number of annotated objects in the Berkeley Deep Drive (BDD100K) dataset.

Set	Vehicles	Pedestrian	Traffic Signs	Traffic Light
Training Set	765,066	95,866	239,686	186,117
Testing Set	218,587	27,524	68,984	53,291
Validation Set	109,807	13,911	34,908	26,885

3.2. Dataset Augmentation

Data augmentation is a set of different techniques used to enhance the size and quality of training models. The objective is to increase the generalization ability without changing the category of images. This article performs basic augmentation operations on the BDD100K dataset, including image displacement, horizontal flipping, random scaling, random cropping, motion blurring, and random noise adding, as shown in Figure 8. In addition to basic augmentation methods, this article uses Mosaic data augmentation [53]. The Mosaic is a new data augmentation technique proposed by Jocher et al. in 2020 that uses four training images instead of a single image [53]. This has two main advantages. Firstly, the model learns to recognize objects outside of the normal contexts found in the dataset. Secondly, it improves the batch normalization statistics using small mini-batches, making the training possible on a single GPU. The implementation of Mosaic data augmentation is shown in Figure 9.

Figure 8. Data augmentation methods (**a**) original image (**b**) displacement (**c**) horizontal flipping (**d**) random cropping (**e**) motion blurring (**f**) random noise adding.

PROOF

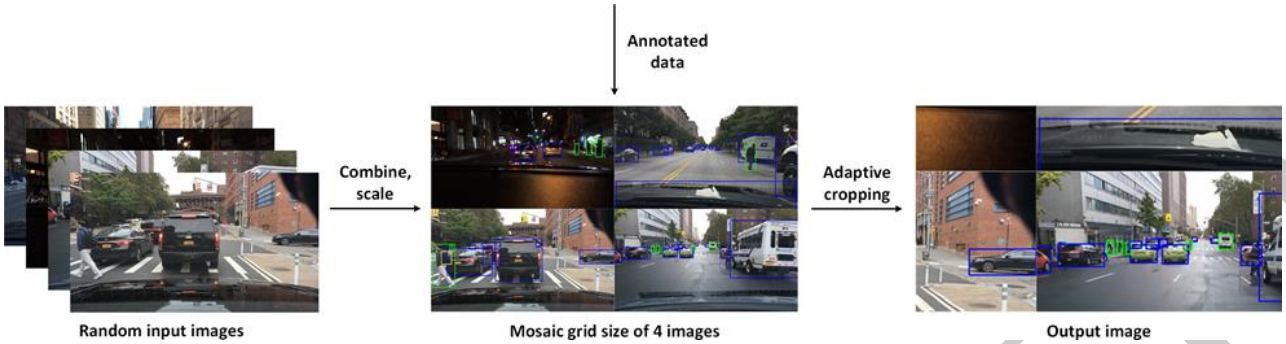


Figure 9. Implementation of the Mosaic data Augmentation.

3.3. Experimental Setup

The popular deep learning frameworks, TensorFlow [54] and Caffe [55], are used to train the models as per official documentation. The experimental environment consists of CUDA v10.1, cuDNN v7.6.4, OpenCV v4.5.0, Ubuntu 18.04.5 OS, 16 GB RAM, Intel i7-8750H CPU, and Nvidia GeForce RTX 2080 Ti GPU. The augmentation is applied to all the images to increase the learning ability. Therefore, the optimal batch sizes of 2 (for Mask-RCNN), 8 (for R-FCN and RetinaNet), and 16 (for SSD and YOLOv4) are used in this evaluation. The initial value of the learning rate is set to 0.0001, the momentum rate is set to 0.949, and the decay rate is set to 0.001 to reduce training loss and prevent gradient explosion. No other parameters were modified and taken as default by the frameworks. The training lasted for 500 epochs. After training, the network weights are obtained to test the experimental results on images from the BDD100K test dataset.

3.4. Object Detection Metrics

For evaluation of detection performance, the mean Average Precision (*mAP*) is computed. A *mAP* is currently the most popular object detection measure for evaluating detection methods, as described by PASCAL VOC Challenge [56]. The basic parameters used are True Positive (*TP*), False Positive (FP), and False Negative (*FN*). *TP* is the number of detections in both ground truth and results. FP is the number of detections in results only, but not in the ground truth. *FN* is the number of detections in ground truth only, but not in results. Precision (*P*) is the percentage of correct positive predictions, defined in Equation (1). Recall (*R*) is the percentage of correct positive predictions among all ground truths, defined in Equation (2). The *P* and *R* cannot comprehensively determine the performance of a model, so Average Precision (*AP*) is obtained individually for each class. The *AP* is a more accurate metric than F1-score because it displays the relationship between *P* and R glob-ally; therefore, F1-score is not calculated in this article. A Precision-Recall (PR) curve is initially computed based on prediction results against all the ground truth values. A prediction is considered as *TP* if it satisfies the following two criteria: (1) the bounding box it represents has an IOU > 50% (default threshold) relative to its corresponding ground truth bounding box, and (2) the predicted class label must be same as the ground truth. The curve is then updated by monotonically decreasing the precision. This is obtained by considering the precision for recall R to the maximum precision obtained for whose recall value is greater than or equal to R. Finally, *AP* is the area under the PR curve calculated under the default threshold, defined in Equation (3). The *mAP* is simply the average of *AP* over all classes, defined in Equation (4).

$$Precision(P) = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \tag{1}$$

$$Recall(R) = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truths} \tag{2}$$

RETRACTED

$$Average\ Precision\ (AP) = \sum_{n}(R_{n+1} - R_n)^{max}_{R:\widetilde{R}\geq R_{n+1}} P\left(\widetilde{R}\right) \quad (3)$$

where $P\left(\widetilde{R}\right)$ is the measured precision at recall $\widetilde{R}$

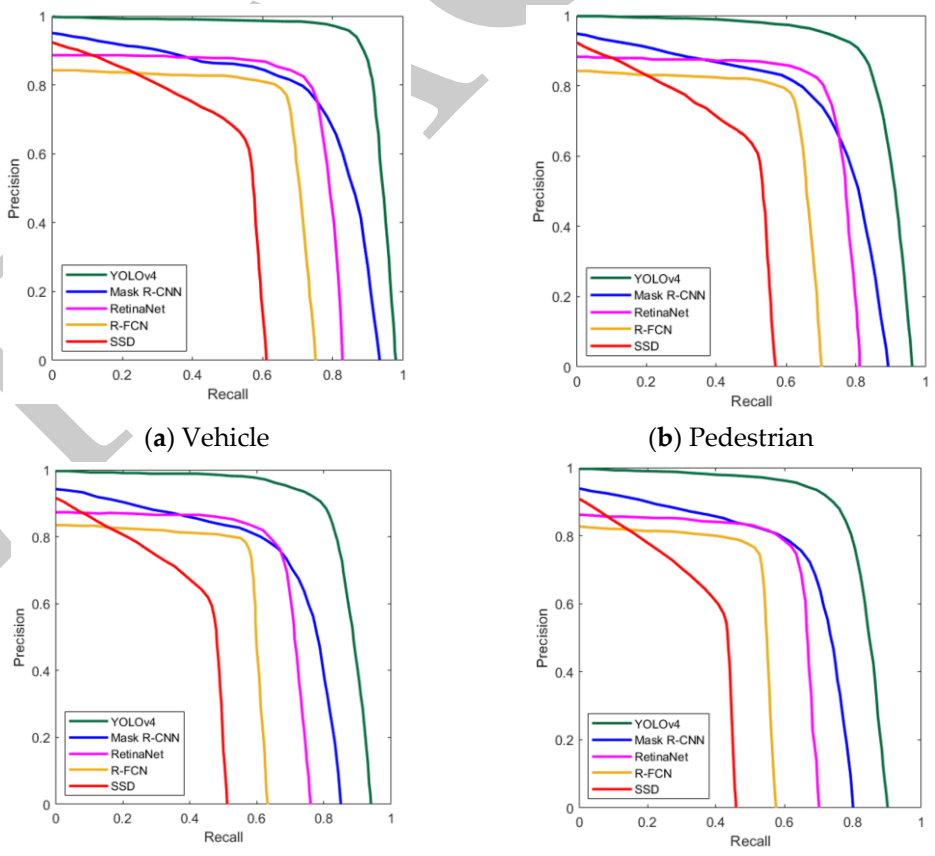
$$mean\ Average\ Precision(mAP) = \frac{1}{4}\sum_{i=1}^{4} AP_i \quad (4)$$

where AP_i is the AP at class i

4. Results and Discussion

4.1. Results

From earlier studies [16,57], it is found that occluded and truncated objects can greatly impact the performance of an object detection model. Therefore, in this comparative study, the test set is divided into two category levels, i.e., objects without occlusion and truncation and objects with occlusion and truncation. Figure 10 shows the PR curves obtained during model training of each object detection algorithm for each class without occlusion and truncation. Figure 11 shows the PR curves obtained during model training of each object detection algorithm for each class with occlusion and truncation. Table 5 lists the detection performance in terms of AP of each class and mAP of object detection models. Table 6 lists the comparison of object detection algorithms with average execution time on CPU and GPU. All results are obtained on the images from the BDD100K test dataset. The best values in Tables 5 and 6 are represented in bold.

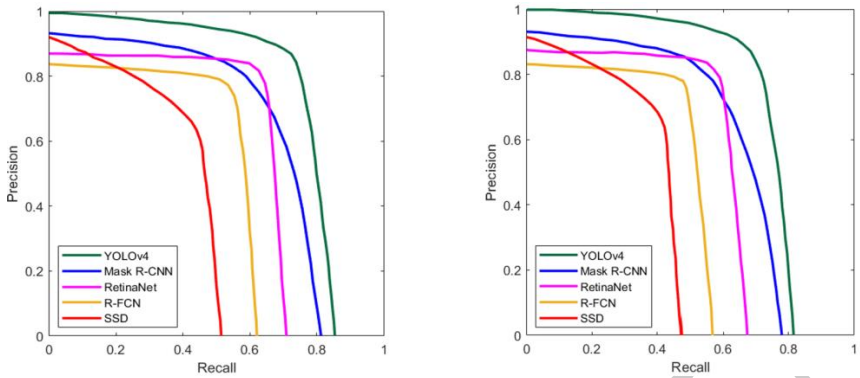


(**a**) Vehicle (**b**) Pedestrian

(**c**) Traffic Sign (**d**) Traffic Light

Figure 10. Precision-Recall (PR) curves of object detection models without occlusion and truncation on BDD100K dataset.

RETRACTED



(**a**) Vehicle (**b**) Pedestrian

(**c**) Traffic Sign (**d**) Traffic Light

Figure 11. Precision-Recall (PR) curve of object detection models with occlusion and truncation on BDD100K dataset.

Table 5. Detection Accuracy of Object Detect Models on BDD100K dataset.

Model	Without Occlusion and Truncation (WOT)					With Occlusion and Truncation (WIOT)				
	AP_{vehicle} (%)	$AP_{\text{pedestrain}}$ (%)	$AP_{\text{traffic sign}}$ (%)	$AP_{\text{traffic light}}$ (%)	mAP_{WOT} (%)	AP_{vehicle} (%)	$AP_{\text{pedestrain}}$ (%)	$AP_{\text{traffic sign}}$ (%)	$AP_{\text{traffic light}}$ (%)	mAP_{WIOT} (%)
R-FCN	58.02	55.67	51.96	45.05	52.86	48.17	44.54	39.65	34.45	41.86
Mask R-CNN	73.08	68.88	66.27	62.08	67.61	66.11	64.64	56.79	55.21	60.69
SDD	46.46	44.33	39.67	35.46	41.52	37.15	35.26	33.47	24.32	32.62
RetinaNet	68.05	62.67	60.96	58.05	62.57	59.17	58.54	50.65	47.45	54.02
YOLOv4	**93.26**	**90.04**	**87.52**	**83.26**	**88.67**	**78.25**	**73.63**	**70.44**	**68.86**	**73.09**

Table 6. The computation time of object detection models on the BDD100K dataset.

Model	CPU Time (s)	GPU Time (s)
R-FCN [26]	2.96	0.17
Mask R-CNN [27]	3.32	0.35
SDD [28]	**0.04**	**0.01**
RetinaNet [29]	3.68	0.20
YOLOv4 [30]	0.10	0.02

Power consumption using different feature extraction can be found at [58].

From Figures 10 and 11, the following inferences can be drawn:

- The one-stage detection model YOLOv4 achieves the highest detection precision and recall for target detection at all levels.
- Mask R-CNN's two-stage detection model shows better precision and recall than RetinaNet, R-FCN, and SSD for target detection at all levels.
- The R-FCN shows better precision as compared to SSD for target detection at all levels.

RETRACTED

- The recall of Mask R-CNN is close to YOLOv4 for target detection with occlusion and truncation.
- SSD recall is the lowest among all the models, which shows a high miss detection rate for targets at all levels.

From Table 5 following inferences can be drawn:

- The one-stage detection model YOLOv4 achieves the highest detection accuracy for target detection at all levels.
- The two-stage detection model Mask R-CNN shows better detection accuracy over RetinaNet, R-FCN, and SSD.
- The SSD shows the lowest *AP* among all the models at all levels.

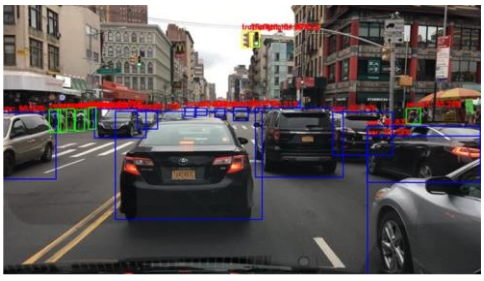
From Table 6 following inferences can be drawn:

- The one-stage detector SSD is faster than other two-stage and one-stage detection models.
- The YOLOv4 shows almost the same detection speed as compared to SSD on GPU.
- The R-FCN is faster than Mask R-CNN and RetinaNet on CPU and GPU.
- The Mask R-CNN is slower than other two-stage and one-stage detection models on GPU.
- The RetinaNet is slower than other two-stage and one-stage detection models on CPU.

In this evaluation, the vehicles are considered large objects, pedestrians as medium objects, and traffic signs and traffic lights as small objects. Figure 12 shows the performance of object detection models on few test images. This article selects few distinct images based on the occlusion and truncation levels to visually show each algorithm's effect. From top to bottom, object detection algorithms are R-FCN, Mask R-CNN, SSD, RetinaNet, and YOLOv4. It can be seen clearly that all models have good performance in large target detection, but there is a huge difference in detecting small targets. The selection of YOLOv4 can be considered best for target detection at all levels. The SSD has started to miss targets; as occlusion and truncation increase, the vehicles and pedestrians occluded in the lower left and right are not detected accurately. As the detection size reduces with occlusion and truncation, the YOLOv4 and, in few cases, Mask R-CNN detect the medium and small targets accurately.

In addition to further test the accuracy on the BDD100K dataset, to investigate the generalization ability of each model under complex weather scenarios, various models are used to test under different road environmental conditions during the clear, partly cloudy, overcast (as shown in Figure 13), foggy, snowy, and rainy (as shown in Figure 14) at different times of day and night. Table 7 lists the generalization ability of five object detection models using low, medium, and high indicators for all road weather conditions at daytime, dawn/dusk, and night. The one-stage detector YOLOv4 has shown to be very strong for the generalization of real scenarios with an excellent detection capability in other road environments. Moreover, the two-stage detector Mask R-CNN performs equivalently to YOLOv4 in all scenarios at night. However, YOLOv4 performs accurately in precipitating dusk/dawn while Mask R-CNN performs accurately on rainy days.

(**a**) R-FCN$_{WOT}$

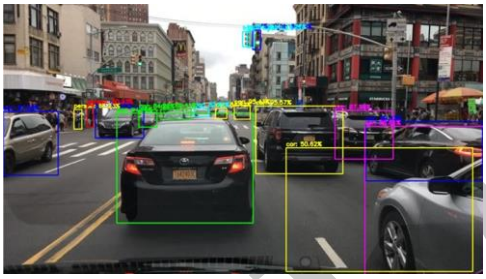


(**b**) R-FCN$_{WIOT}$

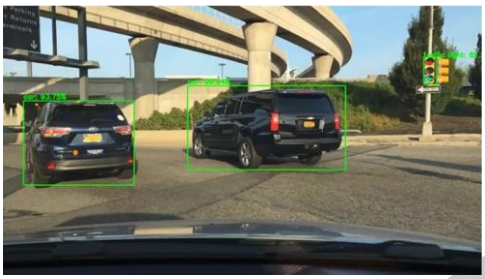
Figure 12. *Cont.*

RETRACTED

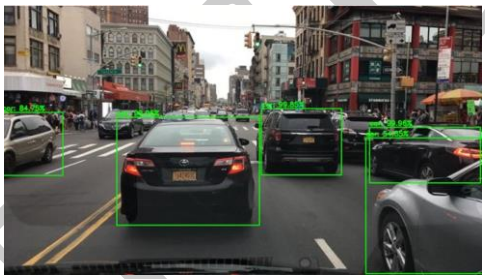
(**c**) Mask R-CNN$_{WOT}$



(**d**) Mask R-CNN$_{WIOT}$

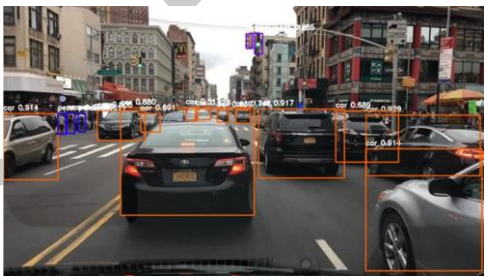


(**e**) SDD$_{WOT}$



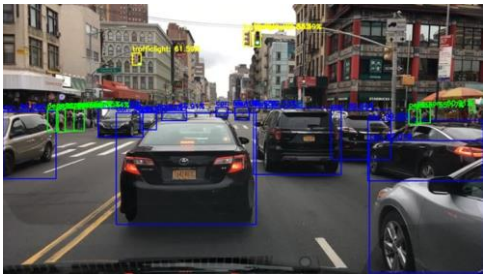
(**f**) SDD$_{WIOT}$

(**g**) RetinaNet$_{WOT}$



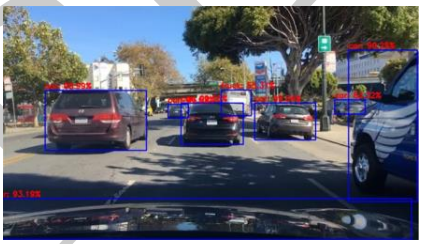
(**h**) RetinaNet$_{WIOT}$

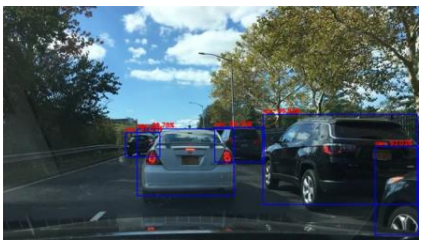
(**i**) YOLOv4$_{WOT}$

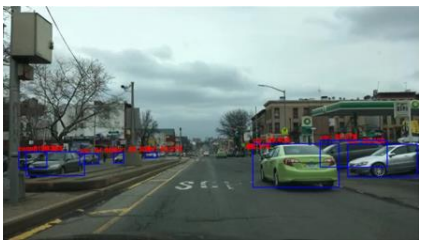


(**j**) YOLOv4$_{WIOT}$

Figure 12. Performance of five deep learning models in two category levels.

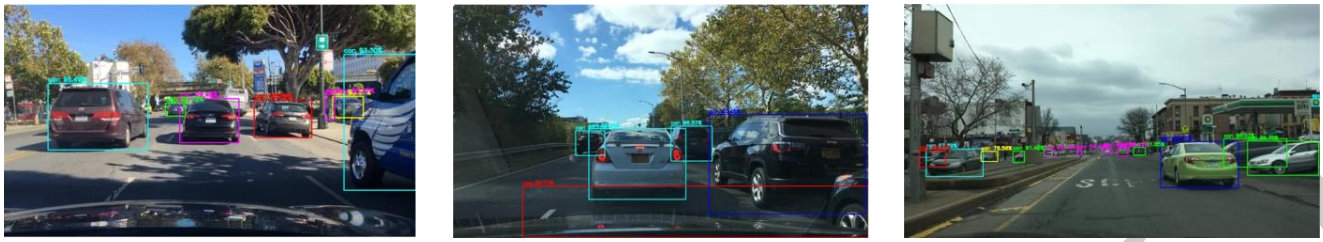




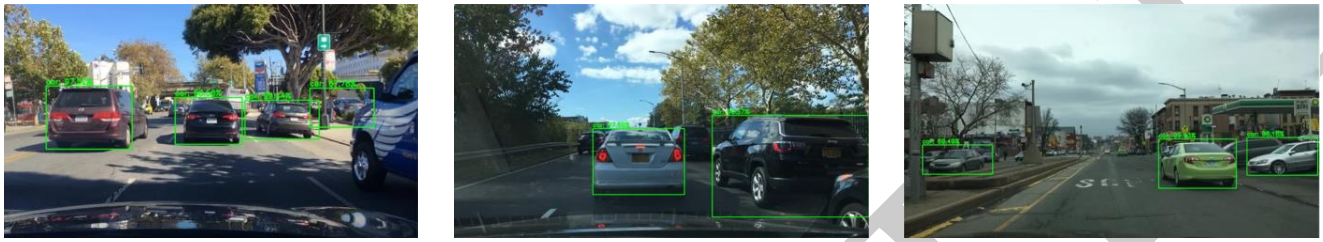


(**a**) R-FCN

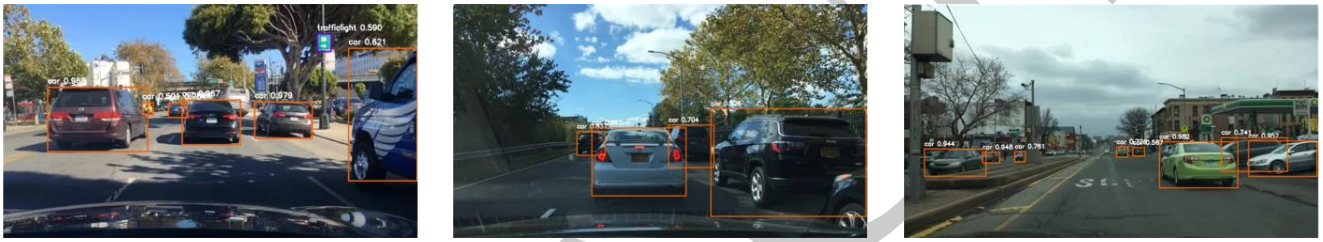
Figure 13. *Cont*.



(**b**) Mask R-CNN



(**c**) SDD



(**d**) RetinaNet

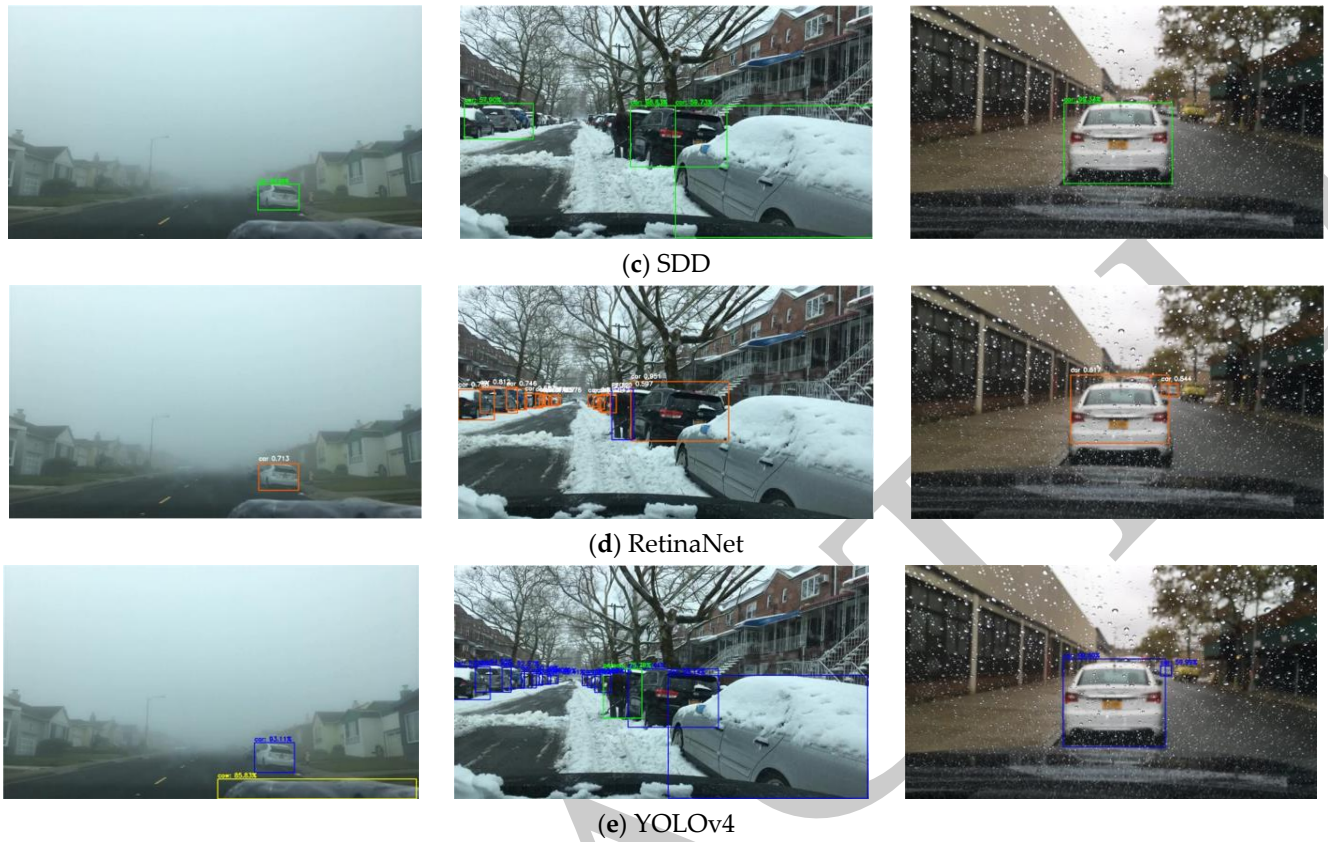
(**e**) YOLOv4

Figure 13. Performance of five deep learning models in different road environmental conditions (from left to right columns: clear, partly cloudy, and overcast).

(**a**) R-FCN

(**b**) Mask R-CNN

Figure 14. *Cont.*



(**c**) SDD

(**d**) RetinaNet

(**e**) YOLOv4

Figure 14. Performance of five deep learning models in different road environmental conditions (from left to right columns: foggy, snowy, and rainy).

Table 7. The generalization ability of object detection models on the BDD100K dataset.

Model		Weather Scenarios					
		Clear	Partly Cloudy	Overcast	Foggy	Snowy	Rainy
R-FCN [26]	daytime	82.10	80.23	77.87	74.20	70.77	59.50
	dawn/dusk	79.61	77.33	75.37	69.20	64.77	57.01
	night	75.37	74.22	64.77	39.52	45.69	40.01
Mask R-CNN [27]	daytime	87.98	85.70	77.01	75.20	71.77	61.32
	dawn/dusk	85.58	81.23	73.26	67.70	61.54	55.38
	night	86.73	82.85	72.72	66.76	60.98	52.01
SDD [28]	daytime	80.17	78.27	75.20	65.89	57.90	58.36
	dawn/dusk	75.37	74.22	73.26	65.12	57.58	55.08
	night	74.22	73.37	72.72	64.95	56.50	55.15
RetinaNet [29]	daytime	83.98	81.70	76.10	71.30	72.56	54.40
	dawn/dusk	79.67	77.74	73.82	67.23	61.99	54.96
	night	75.41	74.53	73.75	67.42	61.95	54.98
YOLOv4 [30]	daytime	91.16	88.49	79.93	77.01	75.79	65.86
	dawn/dusk	88.59	86.54	78.93	76.26	75.14	58.37
	night	87.72	84.70	76.36	72.72	63.90	55.76

AP Threshold Values: High (Dark Blue), Medium (Gold), and Low (Red).

RETRACTED

4.2. Analysis of Results

The two-stage detectors generally showed better detection accuracy compared to one-stage detectors. As per the above experimental results, it is observed that the one-stage detection algorithm YOLOv4 has outstanding performance in terms of both speed and accuracy than other one-stage and two-stage detection algorithms for road object detection. It uses a new backbone, CSPDarkNet-53, which increases the accuracy of the classifier and detector. Moreover, adding an SPP block with PANet increases the receptive field that separates the most significant context features without reducing the network speed. This makes the YOLOv4 detection accuracy exceed the two-stage detectors, with the advantage of fast detection speed on the BDD100K dataset. The two-stage detection algorithm Mask R-CNN shows good detection accuracy because it uses Roi Align to produce a more accurate pixel-to-pixel bounding box alignment. As shown in Figures 10 and 11, the SSD shows a low precision and recall because the model runs detection for medium and small targets from very few layers; therefore, it has insufficient semantic and edge information to detect the majority of road objects. Whereas the R-FCN shows a low precision and recall because it does not use global average pooling, final predictions are based on the position-sensitive feature maps decreasing the overall precision and recall. The Mask R-CNN shows a low precision because it ignores the spatial information between receptive fields, which means that it does not consider the instance details between the edge pixels. Furthermore, the Mask R-CNN uses multiscale FPN, and the Roi Align module reduces misalignments between classification confidence and predicted boxes, improving the detection of medium and small targets. The recall rate of Mask R-CNN is close to YOLOv4 for object detection with occlusion and truncation because the number of false detections is significantly higher than correct detections; therefore, it cannot be considered a balanced algorithm. In this article, it is found that the accuracy of two-stage models is lower than one-stage models. As the detection level changes, the performance decreases with the occlusion and truncation transformation, particularly for the SSD because it has started to miss targets. The SSD shows high precision and low recall, increasing the miss detection rate, indicating that it was more unchecked while detecting medium and 12 small targets; therefore, it cannot be considered a balanced algorithm.

On the contrary, the RetinaNet has strong adaptability to multiscale images because of FPN and Focal Loss; therefore, it achieves good results for medium and large target detection. As shown in Table 5, the RetinaNet can be considered as a balanced algorithm. Still, the YOLOv4 can be seen as a more balanced algorithm than other models because it maintains high precision on high recall in all category levels for all road objects. This article uses specific test samples like the actual road scenario from the BDD100K test dataset to analyze the detection performance further. As shown in Table 7, the performance results of YOLOv4 show excellent generalization ability even in complex and low-lighting conditions because of using a robust feature extractor, CSPDarkNet-53, but limited in rainy weather conditions. At the same time, the generalization ability of the two-stage detector Mask R-CNN is very high, especially in low-lighting scenarios because of using the RoiAlign layer for accurate localization but limited in snowy and rainy conditions. The R-FCN shows high generalization ability, especially in low-lighting conditions, but is limited in foggy, snowy, and rainy weather conditions. The RetinaNet also shows high generalization ability under different weather scenarios but is very limited in snowy, rainy, and low-lighting conditions, especially nighttime. The SSD shows low generalization ability, especially in foggy, snowy, rainy, and low-lighting conditions.

From Table 6, SSD can be considered a fast object detection algorithm in two-stage and one-stage detection models without any additional hardware requirements. The detection speed of YOLOv4 is almost the same compared to SSD with GPU requirements. The detection speed of Mask R-CNN is slow because of using an additional branch by the Roi Align module, which increases the computation time for final classification and bounding box regression. As per the evaluation of the above experimental results, Table 8 summarizes the comparison of various object detection algorithms in terms of backbone architecture

RETRACTED

used in this experimentation, sensitivity, overall specificity, accuracy, and model complexity using low, medium, and high as an indicator to measure the sensitivity, specificity, accuracy, and complexity of various object detection algorithms.

Table 8. Performance of object detection models on BDD100K dataset.

Model	Backbone	Sensitivity	Specificity	Accuracy	Layers
R-FCN [26]	ResNet-101	100	99.02	52.86	101
Mask R-CNN [27]	ResNet-101-FPN	76.65	69.20	67.61	101
SSD [28]	VGG-16	92.75	96.2	41.52	16
RetinaNet [29]	ResNet-101-FPN	76.65	75.69	62.57	101
YOLOv4 [30]	CSPDarkNet-53	69.61	68.52	88.67	53

In summary, the YOLOv4 model has shown outstanding accuracy with fast detection speed, satisfying the current standards of real-time object detection, and can be deployed to the road driving environment for actual vehicles as compared to other discuss models. The two-stage detector Mask R-CNN shows good detection accuracy and high generalization under various low-lighting scenarios but cannot be applied to real-time applications and is more suitable for applications, such as identifying drivable areas, lane detection, etc. The RetinaNet shows good accuracy but has low detection speed and cannot be applied to real-time applications when the vehicle is running unless its hardware requirements are fulfilled. It is more suitable for low-speed scenarios, such as automatic parking, etc. The R-FCN has shown low accuracy and is more suitable for applications like traffic detection, obstacle detection, etc. The SSD shows extremely fast detection speed without any additional hardware requirements but low detection accuracy and can be suitable for deployment on various embedded devices and mobile platforms due to its low power requirements, such as Unmanned Aerial Vehicles (UAVs) that can perform real-time detection with limited computing power.

4.3. Discussion

In earlier two-stage and one-stage target detector models, the two-stage models always had better accuracy and slow speed than the one-stage models. Tables 6 and 8 show that the one-stage detection algorithm YOLOv4 achieves high detection accuracy while maintaining fast detection speed outperforming the two-stage detection algorithm Mask R-CNN because of a new backbone network, CSPDarkNet-53. In addition, YOLOv4 uses SAT, BoF, and BoS methods to increase detection accuracy during detector training. The Mask R-CNN has shown high generalization ability because of using the Roi Align module for accurate localization. But it cannot be applied to real-time applications because of its highly complex architecture and slow detection speed. The RetinaNet has shown good detection accuracy because it uses a special loss function called, Focal Loss to optimize the weights of easy samples; therefore, the model concentrates more on classifying difficult samples during training. With RPN, RetinaNet can be suitable for applications that require good detection accuracy with certain hardware requirements, such as GPU for real-time performance. The R-FCN has low accuracy and cannot be applied to real-time applications due to its slow detection speed. Through comparative analysis, this article finds that even though SSD is an old object detection algorithm, it still has a huge advantage in real-time detection compared to other object detection algorithms without any additional hardware requirements. The SSD uses a simple feature extraction network that effectively reduces the model complexity. The SSD has low computation requirements and lightweight architecture. This article considers SSD suitable for deployment on embedded devices and mobile platforms, such as UAVs, to perform real-time environment perception instead of traditional object detection algorithms. The YOLOv4 has low computation requirements, but the network introduces certain complexity in the model depending on the selection of architecture and other features. This article finds that the YOLOv4 may accurately

RETRACTED

detect difficult road target objects under complex road scenarios and weather conditions. It is worth mentioning that YOLOv4 sometimes fails to perform detections on very small occluded and truncated objects.

On the contrary, this article also tests various deep learning-based algorithms under different road scenarios. The YOLOv4 has excellent generalization ability in multiple road environmental scenarios. Though YOLOv4 has a medium complex architecture, it satisfies the current requirements for real-time applications with strong detection robustness, proving its feasibility in practical engineering. The YOLOv4 model introduces many new features, and through the combination of various such features, state-of-the-art results can be achieved. To the best of our knowledge, these types of study are very rare in which to show the performance of one-stage detection models exceeds the two-stage detection models on a real large-scale dataset. This article selects five models based on significant improvements over the decade and trains the individual models with Mosaic data augmentation [53] to increase the learning ability under complex scenarios. Researchers deeply passionate about analyzing and applying deep learning-based algorithms in ITS can get inspiration from this comparative study. For instance, the researchers can apply various new features to existing models, such as Mosaic training, SAT, and CmBN, to efficiently train the model at a low cost and improve the detection performance. The selection of optimal hyper-parameters, modifying the loss function, and hard training difficult samples instead of adding more layers in the model are additional possibilities that researchers can explore. Moreover, accurate detection of very small target objects, such as traffic lights under low-lighting conditions, is still challenging to solve in the future of deep learning.

5. Development Trends

Object detection procedures generally involve an initial step before producing the final output, known as post-processing. This operation is computationally inexpensive when compared to the detection time. In most cases, the highest prediction result of the detected object is used for the accuracy calculation. An effective post-processing method can improve the performance of many objects detection models with minimum computational requirements. A common post-processing method, such as non-maximal suppression and improvements, can eliminate high IoU and high classification confidence objects, resulting in incorrect detection and classification. Therefore, exploiting a simpler, effective, and accurate post-processing technique is another direction for researchers in the object detection domain.

Recently, many researchers have proposed several deep learning-based models for object detection, but the solutions are limited to a strict local environment due to the high complexity of dynamic scenarios. The pre-existing object detectors, i.e., two-stage focus on high localization and precision, and one-stage focus on high inference speed, both have advantages and disadvantages in the practical engineering field. Further, they use multiscale anchors for learning bounding box coordinates to improve accuracy; still, it is difficult to select optimal parameters of anchors. Therefore, to address this issue and fully inherit the advantages of both types of detectors while overcoming their limitations, advanced anchor-free detectors have attracted much research in recent times. Although these methods achieve better efficiency, they usually compromise accuracy. Therefore, maintaining the balance between accuracy and computational complexity remains a big challenge for many researchers.

In the past few years, several improvements have been proposed to many existing object detection models, but it is found that the existing models can be very difficult to improve under the original framework. Therefore, a new set of object detection models, named EfficientDet [59], was proposed by M. Tan et al. in 2020. EfficientDet [59] uses a weighted Bi-directional Feature Pyramid Network (BiFPN) and EfficientDet [59] backbones to achieve better accuracy and efficiency. EfficientDet-D7 achieved state-of-the-art accuracy of 53.7 on the MS-COCO dataset [59]. Designing new features with fewer parameters for

RETRACTED

backbone and detector can achieve state-of-the-art results, which can be an interesting challenge for researchers.

6. Conclusions

This article presents a comparative study on five independent deep learning-based algorithms (R-FCN, Mask R-CNN, SSD, RetinaNet, YOLOv4) for road object detection. The BDD100K dataset is used to train, validate, and test the individual deep learning models to detect four road objects: vehicles, pedestrians, traffic signs, and traffic lights. The comprehensive performance of the models is compared using three parameters: (1) precision rate, recall rate, and *AP* of four object classes on the BDD100K dataset; (2) *mAP* on the BDD100K dataset; and (3) CPU and GPU computation time on the BDD100K dataset.

The experimental results show that the YOLOv4 in the one-stage detection model achieves the highest detection accuracy for target detection at all levels, while in the two-stage detection model, Mask R-CNN shows better detection accuracy over RetinaNet, R-FCN, and SDD. The SDD shows the lowest *AP* among all the models at all levels. However, the computation time of the models for object detection is different from accuracy. We conclude that the one-stage detector SSD is faster than other one/two-stages detection models. The YOLOv4 shows almost the same detection speed as compared to SSD on GPU. The R-FCN is faster than Mask R-CNN and RetinaNet on CPU and GPU. The Mask R-CNN is slower than other one/two-stages detection models on GPU. The RetinaNet is slower than other one/tow-stage detection models on CPU.

This work also considers different complex weather scenarios to intuitively evaluate the applicability of individual algorithms for target detection. This article provides a benchmark illustrating researchers' comparison of popular deep learning-based object detection algorithms for road target detection. With the fast developments in smart driving, deep learning-based object detection is still a subject worthy of study. To deploy more accurate scenarios for real-time detection, the need for high accuracy and efficient detection systems is becoming increasingly urgent. The main challenge is to balance accuracy, efficiency, and real-time performance. Although recent achievements have been proven effective, much research is still required for solving this challenge.

Author Contributions: Conceptualization, M.H.; methodology, M.H.; software, M.H.; validation, M.H. and A.G.; formal analysis, M.H.; investigation, M.H.; resources, M.H.; data curation, M.H.; writing—original draft preparation, M.H.; writing—review and editing, A.G.; visualization, M.H.; supervision, A.G.; project administration, M.H. and A.G.; funding acquisition, A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the AGH University of Science and Technology, Grant No: 16.16.120.773.

Data Availability Statement: BDD100K dataset is available at: https://bdd-data.berkeley.edu/ (accessed on 20 October 2020).

Conflicts of Interest: The authors declare no conflict of interest.

RETRACTED

References

1. Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S. A Survey of Deep Learning Applications to Autonomous Vehicle Control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 712–733. [CrossRef]
2. Sivaraman, S.; Trivedi, M.M. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1773–1795. [CrossRef]
3. Ahangar, M.N.; Ahmed, Q.Z.; Khan, F.A.; Hafeez, M. A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges. *Sensors* **2021**, *21*, 706. [CrossRef]
4. Huang, Y.; Chen, Y. Autonomous Driving with Deep Learning: A Survey of State-of-Art Technologies. *arXiv* **2020**, arXiv:2006.06091.
5. Haris, M.; Hou, J. Obstacle Detection and Safely Navigate the Autonomous Vehicle from Unexpected Obstacles on the Driving Lane. *Sensors* **2020**, *20*, 4719. [CrossRef] [PubMed]
6. Jang, C.H.; Kim, C.S.; Jo, K.C.; Sunwoo, M. Design Factor Optimization of 3D Flash Lidar Sensor Based on Geometrical Model for Automated Vehicle and Advanced Driver Assistance System Applications. *Int. J. Automot. Technol.* **2017**, *18*, 147–156. [CrossRef]

7. Maqueda, A.I.; Loquercio, A.; Gallego, G.; García, N.; Scaramuzza, D. Event-Based Vision Meets Deep Learning on Steering Prediction for Self-Driving Cars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5419–5427.
8. Fries, C.; Wuensche, H.-J. Autonomous Convoy Driving by Night: The Vehicle Tracking System. In Proceedings of the 2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, USA, 11–12 May 2015; pp. 1–6.
9. Glowacz, A. Ventilation Diagnosis of Angle Grinder Using Thermal Imaging. *Sensors* **2021**, *21*, 2853. [CrossRef]
10. Kumar, S.; Shi, L.; Ahmed, N.; Gil, S.; Katabi, D.; Rus, D. Carspeak: A Content-Centric Network for Autonomous Driving. *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *42*, 259–270. [CrossRef]
11. Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J.Z.; Langer, D.; Pink, O.; Pratt, V. Towards Fully Autonomous Driving: Systems and Algorithms. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 163–168.
12. Urmson, C.; Anhalt, J.; Bagnell, D.; Baker, C.; Bittner, R.; Clark, M.N.; Dolan, J.; Duggins, D.; Galatali, T.; Geyer, C. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *J. Field Robot.* **2008**, *25*, 425–466. [CrossRef]
13. Maddern, W.; Pascoe, G.; Linegar, C.; Newman, P. 1 Year, 1000 Km: The Oxford Robotcar Dataset. *Int. J. Rob. Res.* **2017**, *36*, 3–15. [CrossRef]
14. Request for Investigation of Deceptive and Unfair Practices in Advertising and Marketing of the "Autopilot" Feature Offered in Tesla Motor Vehicles. Available online: https://www.autosafety.org/wp-content/uploads/2018/05/CAS-and-CW-Letter-to-FTC-on-Tesla-Deceptive-Advertising.pdf (accessed on 23 May 2019).
15. Novak, V. Google Self-Driving Car. Ph.D. Thesis, Vinnytsia National Technical University, Vinnytsia, Ukraine, 2016.
16. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A Review of Object Detection Based on Deep Learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [CrossRef]
17. Chen, G.; Wang, H.; Chen, K.; Li, Z.; Song, Z.; Liu, Y.; Chen, W.; Knoll, A. A Survey of the Four Pillars for Small Object Detection: Multiscale Representation, Contextual Information, Super-Resolution, and Region Proposal. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**. [CrossRef]
18. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Martinez-Gonzalez, P.; Garcia-Rodriguez, J. A Survey on Deep Learning Techniques for Image and Video Semantic Segmentation. *Appl. Soft Comput.* **2018**, *70*, 41–65. [CrossRef]
19. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]
20. Haris, M.; Glowacz, A. Lane Line Detection Based on Object Feature Distillation. *Electronics* **2021**, *10*, 1102. [CrossRef]
21. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
22. Sun, Z.; Bebis, G.; Miller, R. On-Road Vehicle Detection: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 694–711. [PubMed]
23. Wang, H.; Yu, Y.; Cai, Y.; Chen, X.; Chen, L.; Liu, Q. A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 82–95. [CrossRef]
24. Liu, Y.; Xie, Z.; Liu, H. An Adaptive and Robust Edge Detection Method Based on Edge Proportion Statistics. *IEEE Trans. Image Process.* **2020**, *29*, 5206–5215. [CrossRef]
25. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access* **2020**, *8*, 58443–58469. [CrossRef]
26. Dai, J.; Li, Y.; He, K.; Sun, J. R-Fcn: Object Detection via Region-Based Fully Convolutional Networks. *arXiv* **2016**, arXiv:1605.06409.
27. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
28. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8 October 2016; pp. 21–37.
29. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
30. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
31. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. Bdd100k: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2636–2645.
32. Lienhart, R.; Maydt, J. An Extended Set of Haar-like Features for Rapid Object Detection. In Proceedings of the international Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; Volume 1.
33. Rybski, P.E.; Huber, D.; Morris, D.D.; Hoffman, R. Visual Classification of Coarse Vehicle Orientation Using Histogram of Oriented Gradients Features. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA, 21–24 June 2010; pp. 921–928.
34. Chen, Z.; Chen, K.; Chen, J. Vehicle and Pedestrian Detection Using Support Vector Machine and Histogram of Oriented Gradients Features. In Proceedings of the 2013 International Conference on Computer Sciences and Applications, Wuhan, China, 14–15 December 2013; pp. 365–368.

RETRACTED

35. Broggi, A.; Cardarelli, E.; Cattani, S.; Medici, P.; Sabbatelli, M. Vehicle Detection for Autonomous Parking Using a Soft-Cascade AdaBoost Classifier. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 912–917.
36. Aziz, L.; bin Haji Salam, S.; Ayub, S. Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review. *IEEE Access* **2020**, *8*, 170461–170495. [CrossRef]
37. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
38. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
39. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
40. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–10 February 2017.
41. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
42. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; Volume 2017, pp. 6517–6525.
43. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
44. Hnewa, M.; Radha, H. Object Detection under Rainy Conditions for Autonomous Vehicles. *arXiv* **2020**, arXiv:2006.16471.
45. Husain, A.A.; Maity, T.; Yadav, R.K. Vehicle Detection in Intelligent Transport System under a Hazy Environment: A Survey. *IET Image Process.* **2020**, *14*, 1–10. [CrossRef]
46. Meng, C.; Bao, H.; Ma, Y. Vehicle Detection: A Review. *Proc. J. Phys. Conf. Ser.* **2020**, 12107. [CrossRef]
47. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
48. Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; Yang, R. The Apolloscape Dataset for Autonomous Driving. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 954–960.
49. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, 2–4 November 2016; Volume 41, pp. 11621–11631.
50. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. NuScenes: A Multimodal Dataset for Autonomous Driving. *Comput. Vis. Pattern Recognit.* **2020**, 11621–11631. [CrossRef]
51. Chang, M.-F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D. Argoverse: 3d Tracking and Forecasting with Rich Maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8748–8757.
52. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
53. Ultralytics/Yolov3: V9.5.0—YOLOv5 v5.0 Release Compatibility Update for YOLOv3. 2021. Available online: https://zenodo.org/record/4681234#.YRJXoYgzaM8 (accessed on 12 April 2021).
54. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
55. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
56. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
57. Ma, W.; Wu, Y.; Cen, F.; Wang, G. MD*FN*: Multiscale Deep Feature Learning Network for Object Detection. *Pattern Recognit.* **2020**, *100*, 107149. [CrossRef]
58. Jetson AGX Xavier: Deep Leasrning Inference Benchmarks | NVIDIA Developer. Available online: https://developer.nvidia.com/embedded/jetson-agx-xavier-dl-inference-benchmarks (accessed on 25 July 2021).
59. Tan, M.; Pang, R.; Le, Q. Efficientdet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.

REJECTED