



### Application of an Effective Hierarchical Deep-Learning-Based Object Detection Model Integrated with Image-Processing Techniques for Detecting Speed Limit Signs, Rockfalls, Potholes, and Car Crashes

Yao-Liang Chung D

Article

check for updates

Citation: Chung, Y.-L. Application of an Effective Hierarchical Deep-Learning-Based Object Detection Model Integrated with Image-Processing Techniques for Detecting Speed Limit Signs, Rockfalls, Potholes, and Car Crashes. *Future Internet* 2023, *15*, 322. https://doi.org/10.3390/fi15100322

Academic Editor: Paolo Bellavista

Received: 28 August 2023 Revised: 26 September 2023 Accepted: 26 September 2023 Published: 28 September 2023



**Copyright:** © 2023 by the author. 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/). Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University, Keelung City 202301, Taiwan; ylchung@email.ntou.edu.tw

Abstract: Against the backdrop of rising road traffic accident rates, measures to prevent road traffic accidents have always been a pressing issue in Taiwan. Road traffic accidents are mostly caused by speeding and roadway obstacles, especially in the form of rockfalls, potholes, and car crashes (involving damaged cars and overturned cars). To address this, it was necessary to design a real-time detection system that could detect speed limit signs, rockfalls, potholes, and car crashes, which would alert drivers to make timely decisions in the event of an emergency, thereby preventing secondary car crashes. This system would also be useful for alerting the relevant authorities, enabling a rapid response to the situation. In this study, a hierarchical deep-learning-based object detection model is proposed based on You Only Look Once v7 (YOLOv7) and mask region-based convolutional neural network (Mask R-CNN) algorithms. In the first level, YOLOv7 identifies speed limit signs and rockfalls, potholes, and car crashes. In the second level, Mask R-CNN subdivides the speed limit signs into nine categories (30, 40, 50, 60, 70, 80, 90, 100, and 110 km/h). The images used in this study consisted of screen captures of dashcam footage as well as images obtained from the Tsinghua-Tencent 100K dataset, Google Street View, and Google Images searches. During model training, we employed Gaussian noise and image rotation to simulate poor weather conditions as well as obscured, slanted, or twisted objects. Canny edge detection was used to enhance the contours of the detected objects and accentuate their features. The combined use of these image-processing techniques effectively increased the quantity and variety of images in the training set. During model testing, we evaluated the model's performance based on its mean average precision (mAP). The experimental results showed that the mAP of our proposed model was 8.6 percentage points higher than that of the YOLOv7 model—a significant improvement in the overall accuracy of the model. In addition, we tested the model using videos showing different scenarios that had not been used in the training process, finding the model to have a rapid response time and a lower overall mean error rate. To summarize, the proposed model is a good candidate for road safety detection.

**Keywords:** YOLOv7; Mask R-CNN; Canny edge detection; object detection; speeding; road safety detection

#### 1. Introduction

Cars are ubiquitous in modern society thanks to technological advancements. In Taiwan, a pressing issue is that the rate of road traffic accidents has been rising over the last few years. Table 1 shows the number of road traffic accidents over the past seven years based on the statistics of the National Police Agency, Ministry of the Interior [1]. In 2016 and 2017, there were around 300,000 accidents; in 2018, there were 320,000; in 2019, there were more than 340,000; in 2020 and 2021, there were around 360,000; and in 2022, there were more than 370,000. There is growing public awareness concerning measures to reduce the number of road traffic accidents through prevention. Reducing the number of speeding offences is also an effective means of reducing the number of road traffic accidents. In terms of traffic violations, speeding is historically the leading violation, with Table 2 showing the number of speeding violations in Taiwan. In 2016, there were 2.78 million violations (or approximately 8000 drivers on average per day were fined for speeding), which rose to over 2.81 million in 2017, then to over 2.95 million in 2018, over 3.13 million in 2019, over 3.16 million in 2020, drastically to over 3.57 million in 2021, and finally to over 3.62 million in 2022. It is thus obvious that the number of speeding violations has been growing steadily, year on year.

Year	Total
2016	305,556
2017	296,826
2018	320,315
2019	341,972
2020	362,393
2021	358,221
2022	375,632

**Table 1.** Number of road traffic accidents in Taiwan over the past seven years [1].

Table 2. Number of speeding violations in Taiwan over the past seven years [1].

Year	Total
2016	2,783,751
2017	2,816,540
2018	2,953,940
2019	3,130,010
2020	3,169,631
2021	3,572,665
2022	3,621,383

One of the main reasons for road traffic crashes in Taiwan is the amount of precipitation, which varies by season (northeasterly winds in winter, typhoons in summer). This abundance of rainfall causes erosion and rainwater accumulation in the mountainous topography of Taiwan, resulting in crashes caused by uneven roads due to rockfalls and potholes, especially in hilly areas after rainfall events. Therefore, one way to prevent road traffic crashes would be to detect rockfalls and potholes early and to alert drivers to the occurrence of these conditions ahead.

From the perspective of information and communication technology, deep-learningbased object detection [2], utilizing deep convolutional neural networks (CNNs) as the backbone and convolution as the basis of calculation, offers a feasible and effective solution for the problems outlined above. Specifically, deep-learning-based object detection technologies can be used alongside image-preprocessing techniques to train object detection systems. Sensor components can be used to observe and recognize speed limit signs, rockfalls, potholes, and car crashes (involving damaged cars and overturned cars) on roads. High-performance computers could then effectively calculate the information to generate real-time reminders to drivers to pay attention to the road conditions ahead, thus preventing violations, car crashes, or secondary car crashes.

However, to date, the concurrent use of the two typical types of models in the deeplearning-based object detection field, namely, You Only Look Once (YOLO) series models (e.g., YOLO [3], YOLO9000 [4], YOLOv3 [5], YOLOv4 [6], YOLOv5 [7], YOLOv6 [8], YOLOv7 [9]) and region-based CNN (R-CNN) series models (e.g., R-CNN [10], Fast R-CNN [11], Faster R-CNN [12], Mask R-CNN [13]) with image-processing techniques for road safety detection applications still has room for research. Moreover, there is a paucity of research available on the concurrent detection of speed limit signs, rockfalls, potholes, and car crashes as well as on resolving the speed limits of different vehicles. Compared to the R-CNN series models, the YOLO series models have undergone rapid developments. Interestingly, in January 2023, Ultralytics released the all-new YOLOv8 [14], which is an integration of its seven predecessors and features several new functions such as segmentation, pose estimation, and tracking. In May 2023, Deci further optimized YOLOv8 and released the YOLO-neural architecture search (NAS) model [15], which outperforms the eight previous YOLO models. However, YOLOv8 and YOLO-NAS are more suitable for solving problems in larger and more complex settings, which explains why they have higher computing costs and complexities than the seven previous YOLO models. In other words, the excellence of YOLOv8 and YOLO-NAS are more pronounced when solving problems in larger and more complex settings. These models are unnecessary in this study due to the smaller scale of the problem settings and the limited hardware resources. It should be noted that these models are all two-dimensional (2D) object detection models that are trained using images. A substantial body of research has been conducted in recent years on the detection of three-dimensional (3D) objects using light detection and ranging (LiDAR) point clouds [16,17]. Due to the fact that LiDAR point clouds essentially do not vary with lighting, and LiDAR-based detection is extremely precise, these models are widely applied to self-driving cars [18]. However, compared to 2D object detection, 3D object detection has higher computing costs and longer delays, which are barriers to the low delay requirements in self-driving cars. Furthermore, compared to 2D visual sensors (e.g., video cameras and dashcams), LiDAR is more expensive and is incompatible with most car models. Hence, a consensus must be attained between all LiDAR manufacturers to significantly reduce the price of the technology so that it can meet various market demands and be more accessible on the market. Based on the abovementioned observations and reasons, we proposed an effective detection model that appropriately combines the YOLOv7 and Mask R-CNN algorithms with image-processing techniques for application in various traffic scenarios.

The rest of this paper is divided as follows: Section 2 is a literature review; Section 3 describes the model design, including the scenario description and the design of the overall process framework; Section 4 presents the test results of the proposed model and discusses and analyzes the results; Section 5 concludes by summarizing the findings of the study and presenting directions for future research.

#### 2. Related Work

We collected and consolidated a large number of studies on road safety. The studies were divided into two categories, related to: (1) the detection of traffic signs; and (2) the detection of objects other than traffic signs (such as traffic lights, vehicles, pedestrians, vehicle lanes, rockfalls, potholes, car crashes, etc.). Note that only a handful of studies have detected both types of objects, such as that of Güney et al. [19], in which vehicles, pedestrians, and traffic signs were detected. Thus, the study was categorized in the second category.

#### 2.1. Studies Related to the Detection of Traffic Signs

Yan et al. [20] used the YOLOv4 model alongside the K-means algorithm for traffic sign detection. The results showed that the frame rate was 25.3 frames per second (FPS), and the mean average precision (mAP) was 1.9% higher than that of the original YOLOv4. Wang et al. [21] developed an improved K-means algorithm based on YOLOv4-Tiny for traffic sign detection. By generating appropriate threshold values, the model increased the detection accuracy for long-distance and small targets. The authors also proposed an improved non-maximum suppression algorithm to identify the prediction box and avoid deleting the prediction results of different targets. The results showed that the improved algorithm achieved a 5.73% higher mAP at 87 FPS compared to the original YOLOv4-Tiny. Yang and Tong [22] proposed a model that specifically detected small traffic signs by adding a convolutional layer into the YOLOv4 backbone to obtain feature maps with

richer information about small objects. In addition, as an effective means of integrating multiscale feature maps, the authors proposed two attention modules—upsampling and downsampling-to assist the network in focusing on useful features. The approach's mAP was 2.2% higher than the original YOLOv4. Jiang et al. [23] performed filtering and data augmentation on raw datasets to improve the accuracy of the small target detection of five common detection algorithms—Faster R-CNN, single shot multi-box detector (SSD), RetinaNet, YOLOv3, and YOLOv5—and optimized the network parameters. The experimental results showed that the test accuracies of Faster R-CNN, RetinaNet, YOLOv3, and YOLOv5 were all above 98.2%. Regarding the use of Faster R-CNN in small object detection, Wang et al. [24] developed a sampling method that achieved network optimization by selecting high-quality proposals. Other post-processing solutions have also been proposed for network optimization through resampling. Using Res2net as the backbone of the proposed network, more distinctive features were obtained, allowing it to outperform the accuracy of the Faster R-CNN by 4%. Yang and Zhang [25] compared small target detection through YOLOv4 and YOLOv3 using a dataset of 4000 traffic signs that were manually labeled by the authors. The detection results revealed that YOLOv4 outperformed YOLOv3 by 2.83% in terms of accuracy. Lin et al. [26] proposed an improved YOLOv4 algorithm for small target detection by adding the Inception structure to the regression network. The improved algorithm had a 3.6% higher mAP than the original YOLOv4 network, although the recognition speed decreased marginally. Chen et al. [27] proposed an improved YOLOv5 model for traffic sign recognition by modifying the feature pyramid network (FPN) from three layers to four layers and the path aggregation network (PAN) from two layers to three layers. As a result, the neck of the model additionally generated larger feature maps for identifying small targets, and the mAP was improved by 10% compared to the original YOLOv5.

Bhatt et al. [28] presented a CNN model for traffic sign detection and compared its test results against a German dataset and an Indian dataset. The results revealed that the model achieved a 99.85% accuracy for the former, 91.08% for the latter, and 95.45% for the combined datasets. Tabernik and Skočaj [29] employed a Mask R-CNN method that resolved the complete process of detection and recognition through automatic end-to-end learning. The method was able to detect 200 traffic signs. Barodi et al. [30] proposed a method that involved image processing for detecting triangular, square, and rectangular traffic signs, thus validating the various shapes of traffic signs. Liu et al. [31] suggested a YOLOv4-based traffic sign detection approach by combining a CNN with a multilayer perceptron architecture, thus strengthening the object detection capabilities of the YOLOv4 network. Then, the authors added a same-level connection within the PAN to enhance the network's acquisition of feature information. The experimental results revealed that the improved YOLOv4 algorithm increased the mAP from 72.95% to 78.84%. Liu and Li [32] presented an improved YOLOv4-based algorithm called TSnet, which resulted from restructuring and adjusting the network structure of the YOLOv4. To enhance feature extraction, the authors used DenseNet to replace the residual unit in the backbone network with a dense connection unit. The experimental results indicated that TSnet had a 3.83% higher mAP than YOLOv4. Gan et al. [33] proposed an improved YOLOv4 model for traffic sign recognition derived from adding a cross-layer connection to the YOLOv4 network and adjusting the weight of the transferred feature map, thus enhancing the network's feature extraction capabilities. The method was able to detect a wider range of traffic signs, with a 1.03% mAP improvement over the original YOLOv4. Zhang and Gao [34] developed an improved Canny edge detection method and analyzed its suitability for traffic sign edge detection under various external interference factors. In addition, to address the insufficiency of traffic sign data, Dewi et al. [35] proposed a data augmentation method in which synthetic images were created using least-squares generative adversarial networks (LSGANs), thus enlarging the original image dataset. After mixing the original images with the synthesized LSGANs-generated images, the recognition performance, as tested by YOLOv3 and YOLOv4, had improved, with the former achieving an 84.9% accuracy and the latter an 89.33% accuracy.

Contrastingly, some researchers have focused on network architecture designs with faster detection speeds. Gu and Si [36] presented a lightweight real-time traffic sign detection model that augmented the information sharing between all levels. An optimized network was used to improve the speed of feature extraction and reduce the computational time and hardware requirements. The results showed that the FPS of the new model was 31 units higher than the YOLOv4. Gong et al. [37] similarly enhanced the detection speed of YOLOv4 by improving the YOLOv4 backbone extraction network through separable convolutions. The difference between the mAP of the improved and original models was only 0.88%, but the improved model's detection speed was nearly three times higher. Kong et al. [38] presented a lightweight traffic sign recognition algorithm based on cascaded CNN. Compared to the YOLOv2-Tiny algorithm, the new algorithm had a 55% lower computational complexity and 32% shorter central processing unit (CPU) computational time, while maintaining the mAP at a similar level. Abraham et al. [39] proposed a crossstage partial YOLOv4 model that improved the original YOLOv4 through the use of CNNs. The improved model had a mAP of 79.77% at 29 FPS, which was extremely beneficial for identifying labels in continuously detected videos. Prakash et al. [40] proposed an extended LeNet-5 CNN model, using the Gabor-based kernel followed by the normal convolutional kernel after the pooling layer. The hue and saturation value color space features had faster detection speeds and fewer impacts from illumination.

#### 2.2. Studies Related to the Detection of Objects Other Than Traffic Signs

Pavani and Sriramya [41] performed vehicle detection using the CNN, k-nearest neighbor (KNN), Haar cascade, and YOLO algorithms. The accuracy of YOLO based on five different video tests was 93%, which was higher than that for the CNN, KNN, and Haar cascade (accuracies of 59%, 89%, and 59%, respectively). This highlights YOLO's superior performance in vehicle detection. He [42] compared the performances of Faster R-CNN, SSD, and YOLOv4 using the same road settings and found that the mAP of each model was 73.5%, 81.5%, and 87.2%, respectively. This demonstrates that YOLOv4 had a higher efficiency than the other two models. Yang and Gui [43] presented a high-precision YOLOv4based model for vehicle detection. The vehicle width and height were determined by deepening the CNN and combining the anchor mechanism with the K-means++ algorithm. Feature fusion was performed using FPN+PAN. Lastly, the complete intersection over union was used as the loss function for the coordinate prediction. The results showed that the mAP of the new model was higher than that of the original YOLOv4 by 3%. Hu et al. [44] developed an improved YOLOv4-based video stream vehicle target detection method that resolved the problem of slow detection speeds. The results revealed that the FPS of the improved algorithm was about six units higher than the original model, and the detection outcomes were not lowered when the improved model was used in conjunction with the Camshift tracking algorithm. Yang et al. [45] applied TensorRT, MobileNetv3, and the channel-pruning method to improve the detection speed of YOLOv4. Wang and Zang [46] proposed a YOLOv4-based model that improved the detection speed and accuracy through the use of the MobileNetv1 network. The  $13 \times 13$  prediction frame was replaced with a  $104 \times 104$  prediction frame to increase the precision of small target detection. Meanwhile, the K-means algorithm was used in cluster analysis to generate the anchor box of the network. The results showed that the mAP of the new model was 90.32% at 35 FPS. Even though the mAP was 2.66% less than the original YOLOv4, the model size was only 23.70% of YOLOv4, and the detection speed was 1.66 times higher. Wu et al. [47] proposed the YOLOv5-Ghost model, which involved modifying the network architecture of YOLOv5, thus lowering the computational complexity. The mAP of YOLOv5 was 83.36% at 28.57 FPS, whereas the mAP of YOLOv5-Ghost was 80.76% at 47.62 FPS. This shows that the latter is more suitable for being placed in embedded devices. Güney et al. [19] proposed a YOLOv5-based automatic driver and driver assistance detection system for detecting traffic signs, vehicles, and pedestrians. The system was tested on three embedded platforms

(Jetson Xavier AGX, Jetson Xavier Nx, and Jetson Nano), with the results showing that Jetson Xavier AGX achieved the fastest detection and highest detection precision.

Altaf et al. [48] analyzed and tested the influence of video quality on target detection in self-driving scenarios. Gaussian noise and motion blur were added to the videos to simulate foggy, cloudy, and rainy environments. Zhu et al. [49] proposed an improved YOLOv4-Tiny-based method for detecting targets under poor weather conditions. The model increased the recognition of small targets in street scenarios and was more robust under both rainy and foggy conditions, with a mAP of 68.88% compared to the 64.75% of the original YOLOv4-Tiny. Yu and Marinov [50] reviewed the current developments in obstacle detection systems in automatic vehicles, determining that a combination of different obstacle detection techniques (such as radar, vision cameras, ultrasonic sensors, and infrared camera monitoring) achieved a better representation of driving environments. Hng et al. [51] used an unmanned aerial vehicle camera alongside a proprietary MATLAB algorithm combined with YOLOv4 to detect accident sites and the extent of vehicle damage. The level of traffic congestion was determined based on the pixels of the moving vehicles. Chen et al. [52] proposed an improved algorithm that integrated the PReNet and YOLOv4 networks, which effectively reduced additional convolutional layers and resolved the problem of unideal small target detection results caused by higher network layers. When used alongside the K-means algorithm, the model was able to allocate the targets in different feature maps to more suitable view frames in multiscale detection. Compared to the original YOLOv4, the improved algorithm had a 29.74% higher mAP and 16.26 higher FPS. Wang et al. [53] presented a YOLOv3-based algorithm that used the Retinex image enhancement algorithm to augment the image quality and resolve the issues in vehicle collision detection during poor weather. The model was also able to rapidly detect collision types in mixed traffic flow settings; the detection rate of the proposed algorithm was 92.5%.

Yan et al. [54] presented a traffic light detection method, using YOLOv5 as its algorithmic core to generate anchors through K-means. Using a traffic light dataset, the mAP achieved by the model was 63.3% at a detection speed of 143 FPS. Shubho et al. [55] proposed a traffic offense detection system based on YOLOv4 and YOLOv4-Tiny. The former was used for vehicle detection and had an accuracy of 86%, and the latter was used for helmet detection and had an accuracy of 92%. The DeepSORT algorithm was used to track vehicles though camera modules. Yu and Qiu [56] developed an intelligent vehicle lane detection system that could be used for various road conditions. The images were preprocessed by gray-scaling the images using the weighted-average method and with binarization using the Otsu algorithm. The Canny edge detection algorithm was used as the edge extraction operator. Detection of straight and curved lanes was accomplished using an improved Hough transform and the least-squares method. The detection accuracies for after rain, congested lane, and road surface conditions were 97.715%, 96.313%, and 94.611%, respectively. Trivedi and Negandhi [57] also proposed a method for lane detection in which computer vision was integrated with the Sobel algorithm, thus resolving the error when switching lanes, with YOLO also used to recognize vehicles and other obstacles to assist drivers in their decision making. Chen et al. [58] proposed a lane-marking detector that used CNNs to capture and record lane-marking features while reducing the system complexity and maintaining a high precision. The model achieved a 65.2% mean intersection-over-union accuracy at a detection speed of 34.4 FPS on the CamVid dataset. Wu et al. [59] focused on improving the dashcam storage space and object recognition rate. The experiments revealed that the compressed sensing method of the iterative shrinkage thresholding algorithm with the network was able to reduce the storage space by 60% while maintaining image resolution. In addition, YOLOv4 was able to overcome complex environments by achieving a minimum 80% recognition rate in  $480 \times 480$  pixels. Chung and Lin [60] proposed a YOLOv3-based deep-learning model integrated with Canny edge detection to detect and classify highway accidents, and after that, Chung and Yang [61] proposed a Mask R-CNN-based model integrated with the Retinex image enhancement algorithm for detecting rockfalls and potholes ahead of vehicles traveling on hill roads.

#### 3. Model Design

#### 3.1. Scenario Description

The scenario considered in this study is shown in Figure 1. Car A is traveling on a road (hill road or level street) or highway when the intelligent sensor on the car's dashcam (as depicted in the "Dashcam video screen" inset in Figure 1) detects, and alerts the driver to, the road conditions ahead—for example, the presence of a speed limit sign, a rockfall, pothole, or car crash. The system can be integrated with existing speed limit warning apps to achieve a dual reminder function. When a speed limit sign is detected, the system alerts the driver to slow down so that they can focus on other road conditions or their driving route and prevent a car crash. When rockfalls, potholes, or car crashes are detected, the information is transferred through the Internet of Vehicles (IoV) to the drivers of the vehicles behind (i.e., Car B and Car C), prompting them to slow down to prevent a secondary car crash or further congestion. In addition, the information is simultaneously transferred, via fifth generation (5G) base stations, to the competent authorities (police, fire brigade, paramedics) firsthand, so that they can arrive at the car crash scene quickly to manage it. In this study, we focused on the detection of speed limit signs, rockfalls, potholes, and car crashes as well as the classification of the detected objects. The topic regarding the information transmission through wireless communication technologies could be further explored in future studies (the transmission technologies designed by Chung [62] and Chung and Wu [63] could be revised and applied here).



Figure 1. Schematic showing the object detection scenario.

In this study, the objects to be detected were first divided into speed limit signs (Class A) and rockfalls, potholes, and car crashes (Class B). Because speeding is strongly associated with car crashes, we focused more on Class A object detection by subdividing it into nine subcategories based on speed limits commonly seen on highways or normal roads—30 km/h (Class A\_30), 40 km/h (Class A\_40), 50 km/h (Class A\_50), 60 km/h (Class A\_60), 70 km/h (Class A\_70), 80 km/h (Class A\_80), 90 km/h (Class A\_90), 100 km/h (Class A\_110), and 110 km/h (Class A\_110). This reminded drivers not to exceed the speed limit and to pay attention to the road conditions ahead or the driving route as a means of preventing a speeding offense or a car crash.

In addition, we determined a method for distinguishing between different speed limit signs. In exceptional cases, two speed limit signs appear at the same time, which normally happens on expressways and freeways where vehicles of different sizes are subjected to different speed limits. Our solution for resolving this problem was as follows: when the moving vehicle was a small car, the model settings were set to detect the higher speed limit; conversely, when the moving vehicle was a large automobile, the model settings were set to detect the lower speed limit. The decision-making flow diagram is shown in Figure 2. The decision-making flow diagram is continuously implemented after the vehicle has begun to run and until it has come to a halt. Consider the example of choosing between 90 km/h and 110 km/h when both speed limit signs are detected at the same time. The 90 km/h speed limit would be applied to the larger automobiles, and the 110 km/h speed limit would be applied to the small cars.



Figure 2. Decision-making flow diagram where two speed limit signs are detected at the same time.

#### 3.2. Design of the Overall Process Framework

The overall process framework consisted of three stages, as shown in Figure 3, including data preparation, training, and detection.





## 3.2.1. Data Preparation Stage Image Collection

Prior to model training, a large number of images had to be prepared and then divided into a training set and a test set. In all cases, the resolution of the image was  $416 \times 416$ . The images were sourced from: (1) dashcam footage (including Car Crashes Time [64]); (2) the Tsinghua-Tencent 100K (TT100K) dataset [65]; (3) Google Street View; and (4) Google Images searches.

(1) Dashcam footage:

These were videos recorded by actual drivers on the road. The collected dashcam footage was trimmed into pictures that were saved as JPG files for training and later testing.

- (2) TT100K dataset: This dataset contains data collected by street-mapping vehicles and includes various weather conditions. We screened the dataset and extracted the required information.
- (3) Google Street View:From this, we screen-captured street imagery that contained speed limit signs.
- (4) Google Images searches: We used this resource to search for suitable images.

#### Image Processing

Rotation angle was used to simulate obscured, slanted, or twisted objects. Gaussian noise was used to simulate poor weather. Canny edge detection was used to strengthen the edges of the targets so that their features could be accentuated. The original images were processed separately using the four combinations involving the three image-processing techniques as follows: (1) processing through rotation angle, Gaussian noise, and Canny edge detection, successively; (2) processing through rotation angle and Canny edge detection, successively; (3) processing through Gaussian noise and Canny edge detection, successively; and (4) processing through Canny edge detection only. The tree structure in Figure 4 displays the corresponding images generated through these four image-processing combinations. Combinations 1 and 2 generate four additional images, and Combinations 3 and 4 generate two and one additional images. The joint use of these image-processing techniques effectively increased the number and variety of the images in the training set.



Figure 4. Various images generated by the four image-processing combinations.

#### Labeling Design

We used the LabelImg tool for labeling in YOLOv7 and the LabelMe tool for labeling in Mask R-CNN. In the latter, only the tens and hundreds digits were labeled. This means that for the speed limit signs bearing the numbers 30, 40, 50, 60, 70, 80, 90, 100, and 110, only the numbers 3, 4, 5, 6, 7, 8, 9, 10, and 11 were labeled, as shown in Table 3. This design was used for two reasons: (1) It is extremely rare to see a speed limit number that does not end with 0 (this was seen only on temporary speed limit signs in road sections under construction). Therefore, training and detection in solely the tens and hundreds digits not only reduced the complexity of the training and detection process, but also prevented the problem of indistinguishable features that arose when the number ended with a 0; and (2) detecting the tens and hundreds digits prevented detection errors when the ones digits were obscured (see the 70 km/h speed limit in Table 3), and also for this reason, we used the Mask R-CNN instead of YOLOv7 to classify the speed limit signs. Tables 4–14 visually present all image classes after labeling.

Table 3. Labeling design for different speed limit signs (in km/h) using Mask R-CNN.

Speed Limit	30	40	50	
Speed limit sign	30	40	50	
Speed Limit	60	70	80	
Speed limit sign	60	(7)	80	
Speed Limit	90	100	110	
Speed limit sign	90	100	(110)	

**Table 4.** Labeling Class A scenarios. (Image sources: dashcam footage, Google Street Views of National Freeway No. 1, Expressway 64, and Huanhe Expressway in Taipei City).

**Table 5.** Labeling Class B scenarios. (Image sources: Car Crashes Time and rockfall and pothole images on Google).



**Table 6.** Labeling Class A\_30 scenarios. (Image sources: dashcam footage and Google search of sections with 30 km/h speed limits).



**Table 7.** Labeling Class A\_40 scenarios. (Image sources: TT100K dataset, Google Street Views of Gongguan Road in Taipei City and Daoxiang Road in New Taipei City).



**Table 8.** Labeling Class A\_50 scenarios. (Image sources: TT100K dataset and Google search of sections with 50 km/h speed limits).



**Table 9.** Labeling Class A\_60 scenarios. (Image sources: TT100K dataset, Google Street Views of Yida 2nd Road and National Freeway No. 3A).



**Table 10.** Labeling Class A\_70 scenarios. (Image sources: TT100K dataset, Google Street Views of Expressway 64, Provincial Highway 2B, and Huanhe Expressway in Taipei City).



**Table 11.** Labeling Class A\_80 scenarios. (Image sources: TT100K dataset, Google Street Views of Huanhe Expressway in Taipei City and Hsuehshan Tunnel).



**Table 12.** Labeling Class A\_90 scenarios. (Image sources: TT100K dataset, Google Street Views of Provincial Highway No. 61 and Taichung Loop Freeway).



**Table 13.** Labeling Class A\_100 scenarios. (Image sources: TT100K dataset, Google Street Views of Taichung Loop Freeway and National Freeway No. 4).



**Table 14.** Labeling Class A\_110 scenarios. (Image sources: TT100K dataset, Google Street Views of National Freeway No. 3 and National Freeway No. 1).



#### 3.2.2. Model Training

Because there were so many data categories in this study, the recognition rate was undoubtedly lowered when multiple objects were detected. We performed model training using the YOLOv7 and Mask R-CNN models, and we combined both models into a twolevel approach. The objects were divided into Class A or Class B using YOLOv7, and the speed limit signs were further subdivided into Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_80, Class A\_90, Class A\_100, and Class A\_110 using Mask R-CNN based on the contours of the signs. For each image inputted (note that though only a single image is mentioned here for the sake of brevity, in reality, the images contained original images and processed images), the first level was tasked with training the YOLOv7 model, and an image containing a speed limit sign was processed in the second level, which was tasked with training the Mask R-CNN model. If the image did not contain a speed limit sign, the next image was processed, and the aforementioned process was repeated. In our settings, when the Mask R-CNN had trained 15,000 epochs, the weights of YOLOv7 and Mask R-CNN were saved, and the saved weights were used in the final detection stage. Note that based on the training set used in this study, we have performed various tests under various epochs and concluded that convergence can be attained through 15,000 epochs.

To demonstrate the strengths of our model, we used three models as controls. For the sake of brevity, we have defined the four key components of the overall framework as follows: (1) add the rotation angle and Gaussian noise; (2) add the Canny edge detection algorithm; (3) train the YOLOv7 model; and (4) train the Mask R-CNN model. Table 15 shows the models in this study, denoted as M1, M2, M3, and M4. M4 is our proposed model, and M1, M2, and M3 are the controls. M1 used only YOLOv7 for model training. M2 used YOLOv7 and Mask R-CNN for model training, neglecting rotation angle and Gaussian noise as well as Canny edge detection. M3 used YOLOv7, Mask R-CNN, rotation angle, and Gaussian noise for model training, but Canny edge detection was not taken

into consideration. M4 used YOLOv7, Mask R-CNN, rotation angle, Gaussian noise, and Canny edge detection for model training. It is important to note that the training set of M1 included Class B, Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_80, Class A\_90, Class A\_100, and Class A\_110 objects. The M2, M3, and M4 models were separately trained on the joint YOLOv7-Mask R-CNN model.

Table 15. Models used in this study.

Model	M1	M2	M3	M4
Components	YOLOv7	YOLOv7 + Mask R-CNN	Rotation angle and Gaussian noise + YOLOv7 + Mask R-CNN	Rotation angle and Gaussian noise + Canny edge detection + YOLOv7 + Mask R-CNN

#### 3.2.3. Model Detection

The videos or images were inputted into the trained models for detection. For M1, we used YOLOv7 to perform the detection (the categories used were Class B, Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_80, Class A\_90, Class A\_100, and Class A\_110). For M2, M3, and M4, we first used YOLOv7 to perform the detection (the categories being Class A and Class B) and then used Mask R-CNN to further categorize the speed limit signs detected by YOLOv7 (the categories were Class A\_30, Class A\_40, Class A\_50, Class A\_50, Class A\_60, Class A\_80, Class A\_90, Class A\_100, and Class A\_50, Class A\_60, Class A\_70, Class A\_90, Class A\_90, Class A\_110).

#### 4. Experimental Results and Discussion

#### 4.1. Hardware

Table 16 presents the hardware configurations of the computer used in this study.

Table 16. Computer hardware configurations.

Processor	Intel(R)Core(TM)i5-8500CPU @ 3.00GHZ * 6		
Display card	NVIDIA GeForce RTX 3060 12 GB		
Operating system	Windows 10		
Memory	32 GB (RAM)		

#### 4.2. Training Set

The training set used in this study included various weather conditions (daytime, nighttime, rainy day, foggy day). Daytime was defined as sunny weather with clear visibility, nighttime as good weather but dim lighting, rainy day as rainy conditions, regardless of bright or dull weather, foggy day as poor visibility in the absence of rain. A total of approximately 3300 images were collected, around 2700 of which were Class A (around 300 each in Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_80, Class A\_90, Class A\_100, and Class A\_110) and around 600 were Class B. Table 17 shows the number of Class A and Class B images used in all four weather conditions. Furthermore, we randomly selected 10% of the images in the training set to serve as the validation set.

The images in the training set were mostly screen captures of dashcam footage. The training set was also enriched using images obtained from the TT100K dataset, Google Street View, and Google Images searches. The images captured from dashcam footage accounted for 70% of the training set, and the images from the TT100K dataset, Google Street View, and Google Images searches each accounted for 10%.

Ca	ategory Class A	Class B
Condition	(Speed Limit Signs)	(Rockfalls, Potholes, and Car Crashes)
Daytime	1854	252
Nighttime	441	147
Rainy day	273	160
Foggy day	132	37

**Table 17.** Statistics for the number of Class A and Class B images used in the training set with respect to weather condition.

#### 4.3. Test Set

A ratio of 9:1 was used between the training set and the test set, which resulted in 330 images in the test set (these images were not used in the training set). Table 18 shows the number of Class A and Class B images used in all four weather conditions. To better simulate the conditions during actual driving, all the images in the test set consisted of images captured from dashcam footage. The test set images were first labeled, and the mAP of each model was then calculated based on tests conducted using the post-training weights.

**Table 18.** Statistics for the number of Class A and Class B images used in the test set with respect to weather condition.

Category	Class A (Speed Limit Signs)	Class B (Rockfalls, Potholes, and Car Crashes)
Daytime	186	25
Nighttime	55	15
Rainy day	27	16
Foggy day	13	4

#### 4.4. Model Testing

4.4.1. Comparison of the mAPs of YOLOv7 and Mask R-CNN in Each Model

Table 19 shows the mAP results of YOLOv7 and Mask R-CNN in each model. The mAPs of YOLOv7 and Mask R-CNN in the proposed model (M4) were the highest among all models. The APs with respect to Class A and Class B were also the highest among all models.







Table 19. Cont.

The mAP of M1 was a relatively low 79.60%, which is decent if solely based on accuracy. It was, however, the lowest among all models, which shows that the design of this model was inferior to the others. A probable reason is because of the abundance of classifications (which were all handed by YOLOv7 only), as well as the lack of training on poor weather conditions and rotated images. The use of a more enriched dataset or image classification beforehand are two ways that could compensate for the shortcomings of this model. For M2, we divided the Class A (speed limit signs) and Class B (rockfalls, potholes, and car crashes) objects first using YOLOv7 and then subdivided Class A into Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_80, Class A\_90, Class A\_100, and Class A\_110 using Mask R-CNN. This method reduced the processing load of YOLOv7. The mAPs of YOLOv7 and Mask R-CNN in M2 were 74.80% and 83.10%, respectively, indicating that it was more efficient at recognizing speed limits than M1. By using the M2 training method in addition to Gaussian noise and rotating the angle to augment the variety of images, M3 simulated poor weather conditions as well as scenarios in which the detected objects were obscured, slanted, or twisted. The mAPs of YOLOv7 and Mask R-CNN in M3 rose to 79.30% and 87.30%, respectively. This shows that it is necessary to added noise and interference in order to successfully train a model. M4 was based on the M3 training method but also included Canny edge detection to enhance the image-processing training. Canny edge detection accentuated the contours of the detected objects, especially during rainy or nighttime conditions, thus enhancing the objects' features. The experiments showed that the mAPs of YOLOv7 and Mask R-CNN increased to 85.80% and 89.30%, respectively. This shows that the appropriate addition of noise and interference in the training set as well as the use of Canny edge detection increased not only the number of images in the training set but also their variety and completeness, thereby increasing the mAP. Furthermore, a detailed comparison of the mAPs of YOLOv7 and Mask R-CNN in M3 and M4 revealed that

based on the mAP of YOLOv7, M4 outperformed M3 by six-and-a-half percentage points, and based on the mAP of Mask R-CNN, M4 outperformed M3 by two percentage points. This implies that the addition of Canny edge detection to YOLOv7 resulted in greater benefits than Mask R-CNN. As YOLOv7 extracts objects through framing, and Mask R-CNN extracts objects by outlining their contours, the addition of Canny edge detection for contour feature enhancement complemented YOLOv7's functions more effectively. Therefore, the increase in mAP using YOLOv7 was greater (even though the new mAP of YOLOv7 was not higher than that of Mask R-CNN). Interestingly, when comparing the APs of M3 and M4 with respect to Class A and Class B, M4 outperformed M3 by five percentage points on the basis of Class A and by eight percentage points on the basis of Class B. Consequently, the addition of Canny edge detection was significantly beneficial to Class B in comparison to Class A. Because rockfalls, potholes, and car crashes may occur at more random locations (unlike speed limit signs, which are typically located on both sides of the road or above it), they are more likely to be partially obscured by other objects or to become poorly illuminated compared to speed limit signs. Therefore, the original contour features of rockfalls, potholes, and car crashes may be less prominent or incomplete than those of speed limit signs. It is important to note that because Canny edge detection emphasizes the enhancement of object contours, in terms of the increase in AP, the addition of Canny edge detection is relatively more conducive to the detection of rockfalls, potholes, and car crashes than the detection of speed limit signs (even though the new AP of rockfalls, potholes, and car crashes was not higher than that of speed limit signs).

#### 4.4.2. Comparison between the mAP of M4 and the Other Controls

In M2, M3, and M4, the images were first divided into Class A and Class B using YOLOv7, and then Class A was further subdivided into Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_80, Class A\_90, Class A\_100, and Class A\_110. In M1, the images were all divided into Class B, Class A\_30, Class A\_40, Class A\_60, Class A\_70, Class A\_90, Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_90, Class A\_100, and Class A\_50, Class A\_60, Class A\_70, Class A\_90, Class A\_100, and Class A\_110 using YOLOv7. Because M1 had a different number of categories than the other models (10 for M1 as opposed to 11 for the other three models), for the sake of impartiality, we calculated the overall weighted mAP (hereinafter denoted as the model mAP) of M2, M3, and M4 using the following equation:

Weighted mAP (denoted as model mAP) = 
$$\frac{(\text{mAP of Mask } R - CNN) \times 9 + \text{AP of Class } B}{10}$$
 (1)

The model mAP of each model derived through (1) is presented in Table 20. Even though the model mAPs of M2, M3, and M4 were lower than the mAP of their respective Mask R-CNN, the model mAP of M4 was still above 88% and was the highest among all models. This value is 8.6 percentage points higher than the mAP of M1, in which only YOLOv7 was used. The model mAP results further demonstrate the excellence of M4 in terms of detection accuracy.

Table 20. Model mAP for each model.

Model	M1	M2	M3	M4
Model mAP	79.60%	81.32%	85.60%	88.20%

4.4.3. Continuous Image Testing Using M4

In the aforementioned tests, all 330 images in the test set contained at least one object that we wished to detect. However, in reality, the images in the dashcam footage might only contain the objects we do not intend to detect. If such objects were detected (i.e., incorrect detections), the moving vehicle could potentially transmit the wrong information or even brake sharply, which can be dangerous in itself. To showcase the efficiency of M4, we further selected six pieces of dashcam footage (not used in the training set) for testing.

The features of the six videos were as follows: Video 1 [66] showed a vehicle stuck in a rush-hour traffic jam on a normal road in a city under bright and clear weather conditions; Video 2 [67] showed a vehicle traveling smoothly along a hilly road on a sunny day; Video 3 [68] showed a vehicle traveling smoothly along a freeway on a rainy night; Video 4 [69] showed a vehicle traveling along a hilly road on a rainy day; Video 5 [70] showed a vehicle traveling smoothly alon a rainy day; Video 5 [70] showed a vehicle traveling smoothly along a normal road on a rainy day; and Video 6 [71] showed a scooter traveling smoothly along a normal road on a foggy day. The footage was split into different frames for the detection tests, and the test results are presented in Table 21.

	Images That Did Not Contain Class A or Class B Objects	Images That Contained Class A Objects	Images That Contained Class B Objects	Number of Incorrectly Detected Images	Error Rate
Video 1	59	3	0	1	1.69%
Video 2	54	0	6	0	0%
Video 3	50	0	0	7	14%
Video 4	20	0	0	0	0%
Video 5	62	0	0	3	4.84%
Video 6	74	0	0	14	18.92%

Table 21. Error rates of M4 tested using different videos.

The test results showed that the processing time for each image was 0.28 s. This detection speed is adequate for real-time image recognition, because there were no missed detections caused by late responses. From Table 21, in Video 1, of the 59 images that did not include Class A or Class B objects, only one image was incorrectly detected; in Video 2, of the 54 images that did not include Class A or Class B objects, there were no incorrect detections; in Video 3, of the 50 images that did not include Class A or Class B objects, seven images were incorrectly detected; in Video 4, of the 20 images that did not include Class A or Class B objects, there were no incorrect detections; in Video 5, of the 62 images that did not include Class A or Class B objects, there were no incorrect detections; in Video 5, of the 61 images that did not include Class A or Class B objects, there were no incorrect detections; in Video 5, of the 62 images that did not include Class A or Class B objects, there were no incorrect detections; in Video 5, of the 62 images that did not include Class A or Class B objects, there were no incorrect detections; in Video 5, of the 62 images that did not include Class A or Class B objects, there were incorrectly detected; and in Video 6, of the 74 images that did not include Class A or Class B objects, 14 images were incorrectly detected. The error rate for each video can be calculated using the following equation:

# $Error rate = \frac{Number of images that were incorrectly detected}{Number of images that did not include Class A or Class B objects}$ (2)

From Table 21, Videos 3 and 6 had higher error rates. This is due to the insufficiency of samples for foggy weather and rainy nights in the training set (these samples were difficult to acquire). For the other four videos, the samples were sufficient, and therefore, the error rates were lower or even equal to 0. Therefore, the mean error rate of the six videos was only 6.58%, which attests to the excellence of M4.

#### 4.5. Discussion

Based on the experimental results described above, the two-level training method designed in this study demonstrated favorable results in terms of overall model accuracy when there are many classes of objects to be detected. This also proves that our method can be effectively applied in the detection of various objects. The images in the training set were mostly captured from the dashcam footage of moving vehicles. However, we noted that the training set contained few images of poor weather conditions, which is why we added Gaussian noise to simulate poor weather. We also added several random rotation angles to simulate obscured, slanted, or twisted objects. These treatments made up for the lack of images in these categories and expanded the variety of images. We also added Canny edge

detection to depict the contours of objects and accentuate their features, thus enhancing the model's recognition rate. The detection accuracy (expressed as the mAP) of M4 was 88.20%—significantly higher, by 8.6 percentage points, than that of M1 (79.60%), which only used a single model for detection. Furthermore, for a detection model, in addition to object type detection, it is also important to reduce incorrect detections. We performed tests using videos of different scenarios that had not been used in the training process, finding that M4 also had a quick response time, and the overall mean error rate was only 6.58%. This shows that M4 maintained a certain level of preventing incorrect detections. Overall, M4 is a good candidate model for driver safety detection.

#### 5. Conclusions and Directions for Future Research

We have successfully developed a deep-learning-based object detection model (M4) that jointly uses YOLOv7 and Mask R-CNN alongside image-processing techniques for detecting speed limit signs, rockfalls, potholes, and car crashes. The training was performed in two levels. First, the images were divided into Class A and Class B using YOLOv7. Then, the Class A images were further subdivided into Class A\_30, Class A\_40, Class A\_50, Class A\_60, Class A\_70, Class A\_80, Class A\_90, Class A\_100, and Class A\_110 using Mask R-CNN, thus effectively enhancing the model's detection efficiency. In addition, we employed Gaussian noise and image rotation to make up for the difficulty of acquiring images captured during poor weather conditions and with different object angles. We then used Canny edge detection to enhance the contours of the detected objects and the intensity of their features. Lastly, short videos from the dashcam footage of moving vehicles were mainly used as the training set. The experimental results showed that M4 outperformed all the other models, having a model mAP of 88.20%. The mAPs of YOLOv7 and Mask R-CNN in M4 were 85.80% and 89.30%, respectively—higher than those of the other models and a significant increase from those of M1. We also tested M4 using actual footage of various scenarios and found that the model had a rapid response time and an overall mean error rate of only 6.58%, thus attesting to its accuracy. Based on the relatively limited amount of training set images, M4 achieved a good overall performance, which demonstrates its superiority and efficiency.

We recommend that future studies increase the number of cameras so that images from different angles can be captured. This would also reflect the distance of the detected objects in the images as well as the tradeoff between detection performance and power consumption. Furthermore, it is important to test different weather and road conditions and to increase the detection performance. Integrating this detection model with the IoV and 5G networks is also an issue that warrants further research, because this would increase the completeness and practicality of the system and facilitate the commercialization of the product.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

**Conflicts of Interest:** The author declares no conflict of interest.

#### References

- 1. National Police Agency, Ministry of the Interior. Available online: https://www.npa.gov.tw/en/app/data/view?module=wg055 &id=8026&serno=6f85cc4f-cd02-40d9-b4ea-2c0cc066c17b (accessed on 18 May 2023).
- 2. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. Proc. IEEE 2023, 111, 257–276. [CrossRef]
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
- Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
- 5. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767.
- Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:abs/2004.10934.

- 7. YOLOv5 Is Here: State-of-the-Art Object Detection at 140 FPS. Available online: https://blog.roboflow.com/yolov5-is-here (accessed on 12 December 2022).
- Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* 2022, arXiv:2209.02976.
- 9. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* 2022, arXiv:2207.02696.
- 10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
- 11. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
- 12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497v3. [CrossRef]
- 13. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. arXiv 2018, arXiv:1703.06870v3.
- 14. Ultralytics YOLOv8 Docs. Available online: https://docs.ultralytics.com/models/yolov8 (accessed on 6 February 2023).
- Deci Introduces YOLO-NAS—A Next-Generation, Object Detection Foundation Model Generated by Deci's Neural Architecture Search Technology. Available online: https://deci.ai/blog/yolo-nas-foundation-model-object-detection (accessed on 15 August 2023).
- 16. Yin, T.; Zhou, X.; Krähenbühl, P. Center-based 3D Object Detection and Tracking. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
- 17. Xia, Y.; Wu, Q.; Li, W.; Chan, A.B.; Stilla, U. A lightweight and detector-free 3D single object tracker on point clouds. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5543–5554. [CrossRef]
- 18. Wang, L.; Zhang, X.; Song, Z.; Bi, J.; Zhang, G.; Wei, H.; Tang, L.; Yang, L.; Li, J.; Jia, C.; et al. Multi-modal 3D object detection in autonomous driving: A survey and taxonomy. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3781–3798. [CrossRef]
- 19. Güney, E.; Bayilmiş, C.; Çakan, B. An implementation of real-time traffic signs and road objects detection based on mobile GPU platforms. *IEEE Access* 2022, *10*, 86191–86203. [CrossRef]
- 20. Yan, W.; Yang, G.; Zhang, W.; Liu, L. Traffic Sign Recognition using YOLOv4. In Proceedings of the 2022 7th International Conference on Intelligent Computing and Signal Processing, Xi'an, China, 15–17 April 2022.
- Wang, L.; Zhou, K.; Chu, A.; Wang, G.; Wang, L. An improved light-weight traffic sign recognition algorithm based on YOLOv4-Tiny. *IEEE Access* 2021, 9, 124963–124971. [CrossRef]
- 22. Yang, T.; Tong, C. Small Traffic Sign Detector in Real-time Based on Improved YOLO-v4. In Proceedings of the 2021 IEEE 23rd International Conference on High Performance Computing & Communications; the 7th International Conference on Data Science & Systems; the 19th International Conference on Smart City; the 7th International Conference on Dependability in Sensor, Cloud & Big Data Systems & Application, Haikou, China, 20–22 December 2021.
- 23. Jiang, J.; Yang, J.; Yin, J. Traffic Sign Target Detection Method Based on Deep Learning. In Proceedings of the 2021 International Conference on Computer Information Science and Artificial Intelligence, Kunming, China, 17–19 September 2021.
- 24. Wang, F.; Li, Y.; Wei, Y.; Dong, H. Improved Faster RCNN for Traffic Sign Detection. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, Rhodes, Greece, 20–23 September 2020.
- 25. Yang, W.; Zhang, W. Real-Time Traffic Signs Detection Based on YOLO Network Model. In Proceedings of the 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Chongqing, China, 29–30 October 2020.
- Lin, H.; Zhou, J.; Chen, M. Traffic Sign Detection Algorithm Based on Improved YOLOv4. In Proceedings of the 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference, Chongqing, China, 17–19 June 2022.
- 27. Chen, Y.; Wang, J.; Dong, Z.; Yang, Y.; Luo, Q.; Gao, M. An Attention Based YOLOv5 Network for Small Traffic Sign Recognition. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics, Anchorage, AK, USA, 1–3 June 2022.
- Bhatt, N.; Laldas, P.; Lobo, V.B. A Real-Time Traffic Sign Detection and Recognition System on Hybrid Dataset using CNN. In Proceedings of the 2022 7th International Conference on Communication and Electronics Systems, Coimbatore, India, 22–24 June 2022.
- 29. Tabernik, D.; Skočaj, D. Deep learning for large-scale traffic-sign detection and recognition. *IEEE Trans. Intell. Transp. Syst.* 2020, 21, 1427–1440. [CrossRef]
- Barodi, A.; Bajit, A.; Aidi, S.E.; Benbrahim, M.; Tamtaoui, A. Applying Real-Time Object Shapes Detection to Automotive Traffic Roads Signs. In Proceedings of the 2020 International Symposium on Advanced Electrical and Communication Technologies, Marrakech, Morocco, 25–27 November 2020.
- 31. Liu, Z.; Musha, Y.; Wu, H. Detection of Traffic Sign Based on Improved YOLOv4. In Proceedings of the 2022 7th International Conference on Intelligent Computing and Signal Processing, Xi'an, China, 15–17 April 2022.
- 32. Liu, S.; Li, H. Application of Chinese Traffic Sign Detection Based on Yolov4. In Proceedings of the 2021 7th International Conference on Computer and Communications, Chengdu, China, 10–13 December 2021.
- Gan, Z.; Wenju, L.; Wanghui, C.; Pan, S. Traffic Sign Recognition Based on Improved YOLOv4. In Proceedings of the 2021 6th International Conference on Intelligent Informatics and Biomedical Sciences, Oita, Japan, 25–27 November 2021.
- Zhang, T.; Gao, H. Detection Technology of Traffic Marking Edge. In Proceedings of the 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology, Weihai, China, 14–16 October 2020.

- 35. Dewi, C.; Chen, R.-C.; Liu, Y.-T.; Jiang, X.; Hartomo, K.D. Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. *IEEE Access* **2021**, *9*, 97228–97242. [CrossRef]
- Gu, Y.; Si, B. A Novel Lightweight real-time traffic sign detection integration framework based on YOLOv4. Entropy 2022, 24, 487. [CrossRef] [PubMed]
- Gong, Y.; Peng, J.; Jin, S.; Li, X.; Tan, Y.; Jia, Z. Research on YOLOv4 Traffic Sign Detection Algorithm Based on Deep Separable Convolution. In Proceedings of the 2021 IEEE International Conference on Emergency Science and Information Technology, Chongqing, China, 22–24 November 2021.
- Kong, S.; Park, J.; Lee, S.-S.; Jang, S.-J. Lightweight Traffic Sign Recognition Algorithm Based on Cascaded CNN. In Proceedings of the 2019 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Republic of Korea, 15–18 October 2019.
- Abraham, A.; Purwanto, D.; Kusuma, H. Traffic Lights and Traffic Signs Detection System Using Modified You Only Look Once. In Proceedings of the 2021 International Seminar on Intelligent Technology and Its Applications, Surabaya, Indonesia, 21–22 July 2021.
- Prakash, A.S.; Vigneshwaran, D.; Ayyalu, R.S.; Sree, S.J. Traffic Sign Recognition using Deeplearning for Autonomous Driverless Vehicles. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication, Erode, India, 8–10 April 2021.
- Pavani, K.; Sriramya, P. Comparison of KNN, ANN, CNN and YOLO Algorithms for Detecting the Accurate Traffic Flow and Build an Intelligent Transportation System. In Proceedings of the 2022 2nd International Conference on Innovative Practices in Technology and Management, Gautam Buddha Nagar, India, 23–25 February 2022.
- 42. He, H. Yolo Target Detection Algorithm in Road Scene Based on Computer Vision. In Proceedings of the 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, Dalian, China, 14–16 April 2022.
- Yang, K.; Gui, X. Research on Real-Time Detection of Road Vehicle Targets Based on YOLOV4 Improved Algorithm. In Proceedings of the 2022 3rd International Conference on Electronic Communication and Artificial Intelligence, Zhuhai, China, 14–16 January 2022.
- 44. Hu, X.; Wei, Z.; Zhou, W. A Video Streaming Vehicle Detection Algorithm Based on YOLOv4. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference, Chongqing, China, 12–14 March 2021.
- 45. Yang, F.; Zhang, X.; Zhang, S.; Li, C.; Hu, H. Design of Real-Time Vehicle Detection Based on YOLOv4. In Proceedings of the 2021 International Conference on Control, Automation and Information Sciences, Xi'an, China, 14–17 October 2021.
- 46. Wang, H.; Zang, W. Research on Object Detection Method in Driving Scenario Based on Improved YOLOv4. In Proceedings of the 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference, Chongqing, China, 4–6 March 2022.
- Wu, T.-H.; Wang, T.-W.; Liu, Y.-Q. Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network. In Proceedings of the 2021 3rd World Symposium on Artificial Intelligence, Guangzhou, China, 18–20 June 2021.
- Altaf, M.; Rehman, F.U.; Chughtai, O. Discernible Effect of Video Quality for Distorted Vehicle Detection using Deep Neural Networks. In Proceedings of the 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, 27–30 September 2021.
- 49. Zhu, D.; Xu, G.; Zhou, J.; Di, E.; Li, M. Object Detection in Complex Road Scenarios: Improved YOLOv4-Tiny Algorithm. In Proceedings of the 2021 2nd Information Communication Technologies Conference, Nanjing, China, 7–9 May 2021.
- 50. Yu, X.; Marinov, M. A Study on Recent Developments and Issues with Obstacle Detection Systems for Automated Vehicles. *Sustainability* 2020, *12*, 3281. [CrossRef]
- Hng, T.J.; Weilie, E.L.; Wei, C.S.; Srigrarom, S. Relative Velocity Model to Locate Traffic Accident with Aerial Cameras and YOLOv4. In Proceedings of the 2021 13th International Conference on Information Technology and Electrical Engineering, Chiang Mai, Thailand, 14–15 October 2021.
- 52. Chen, T.; Yao, D.-C.; Gao, T.; Qiu, H.-H.; Guo, C.-X.; Liu, Z.-W.; Li, Y.-H.; Bian, H.-Y. A fused network based on PReNet and YOLOv4 for traffic object detection in rainy environment. *J. Traffic Transp. Eng.* **2022**, *22*, 225–237.
- 53. Wang, C.; Dai, Y.; Zhou, W.; Geng, Y. A vision-based video crash detection framework for mixed traffic flow environment considering low-visibility condition. *J. Adv. Transp.* 2020, 2020, 9194028. [CrossRef]
- 54. Yan, S.; Liu, X.; Qian, W.; Chen, Q. An End-to-End Traffic Light Detection Algorithm Based on Deep Learning. In Proceedings of the 2021 International Conference on Security, Pattern Analysis, and Cybernetics, Chengdu, China, 18–20 June 2021.
- Shubho, F.H.; Iftekhar, F.; Hossain, E.; Siddique, S. Real-time Traffic Monitoring and Traffic Offense Detection Using YOLOv4 and OpenCV DNN. In Proceedings of the TENCON 2021—2021 IEEE Region 10 Conference (TENCON), Auckland, New Zealand, 7–10 December 2021.
- 56. Yu, G.; Qiu, D. Research on Lane Detection Method of Intelligent Vehicle in Multi-road Condition. In Proceedings of the 2021 China Automation Congress, Beijing, China, 22–24 October 2021.
- 57. Trivedi, Y.; Negandhi, P. An Advanced Driver Assistance System Using Computer Vision and Deep-Learning. In Proceedings of the 2018 Second International Conference on Intelligent Computing and Control Systems, Madurai, India, 14–15 June 2018.
- Chen, P.-R.; Lo, S.-Y.; Hang, H.-M.; Chan, S.-W.; Lin, J.-J. Efficient Road Lane Marking Detection with Deep Learning. In Proceedings of the 2018 IEEE 23rd International Conference on Digital Signal Processing, Shanghai, China, 19–21 November 2018.
- Wu, J.-W.; Wu, C.-C.; Cen, W.-S.; Chao, S.-A.; Weng, J.-T. Integrated Compressed Sensing and YOLOv4 for Application in Image-storage and Object-recognition of Dashboard Camera. In Proceedings of the 2021 Australian & New Zealand Control Conference, Gold Coast, Australia, 25–26 November 2021.

- 60. Chung, Y.-L.; Lin, C.-K. Application of a model that combines the YOLOv3 object detection algorithm and Canny edge detection algorithm to detect highway accidents. *Symmetry* **2020**, *12*, 1875. [CrossRef]
- Chung, Y.-L.; Yang, J.-J. Application of a Mask R-CNN-Based Deep Learning Model Combined with the Retinex Image Enhancement Algorithm for Detecting Rockfall and Potholes on Hill Roads. In Proceedings of the 11th IEEE International Conference on Consumer Electronics—Berlin, Berlin, Germany, 15–18 November 2021.
- 62. Chung, Y.-L. ETLU: Enabling efficient simultaneous use of licensed and unlicensed bands for D2D-assisted mobile users. *IEEE Syst. J.* **2018**, *12*, 2273–2284. [CrossRef]
- 63. Chung, Y.-L.; Wu, S.-H. An effective toss-and-catch algorithm for fixed-rail mobile terminal equipment that ensures reliable transmission and non-interruptible handovers. *Symmetry* **2021**, *13*, 582. [CrossRef]
- Car Crashes Time YouTube Channel. Available online: https://www.youtube.com/user/CarCrashesTime (accessed on 17 September 2023).
- 65. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-Sign Detection and Classification in the Wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
- 66. On a Normal Road in the Daytime under a Rush-Hour Traffic Jam Condition. YouTube. Available online: https://youtu.be/ xMHtpM6rSmk (accessed on 16 January 2023).
- 67. On a Hill Road in the Daytime. YouTube. Available online: https://youtu.be/a1\_DcM5tGYM (accessed on 16 January 2023).
- GS980D Front View Footage in 4K on a Highway on a Rainy Night. YouTube. Available online: https://youtu.be/gCry3oTFIc0 (accessed on 16 January 2023).
- Please Drive Safely on Hill Roads When It's Raining. YouTube. Available online: https://youtu.be/RO9-R\_EN3nY (accessed on 1 February 2023).
- 70. Rainy Day Dashcam Footage. YouTube. Available online: https://youtu.be/iQkxeghxAvQ (accessed on 1 February 2023).
- Mio Mivue M733 WIFI Dashcam Footage on a Scooter on a Foggy day. YouTube. Available online: https://youtu.be/CUn27eD0 JaQ (accessed on 1 February 2023).

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.