Article

A Real-Time Vehicle Detection System under Various Bad Weather Conditions Based on a Deep Learning Model without Retraining

Xiu-Zhi Chen, Chieh-Min Chang, Chao-Wei Yu and Yen-Lin Chen *

Department of Computer Science and Information Engineering, National Taipei University of Technology, 1, Sec. 3, Chung-hsiao E. Rd., Taipei 10608, Taiwan; t107599006@ntut.edu.tw (X.-Z.C.); t107598023@ntut.org.tw (C.-M.C.); david741002@ntut.edu.tw (C.-W.Y.)

* Correspondence: ylchen@csie.ntut.edu.tw; Tel.: +886-02-2771-2171 (ext. 4239)

Received: 8 August 2020; Accepted: 7 October 2020; Published: 9 October 2020

Abstract: Numerous vehicle detection methods have been proposed to obtain trustworthy traffic data for the development of intelligent traffic systems. Most of these methods perform sufficiently well under common scenarios, such as sunny or cloudy days; however, the detection accuracy drastically decreases under various bad weather conditions, such as rainy days or days with glare, which normally happens during sunset. This study proposes a vehicle detection system with a visibility complementation module that improves detection accuracy under various bad weather conditions. Furthermore, the proposed system can be implemented without retraining the deep learning models for object detection under different weather conditions. The complementation of the visibility was obtained through the use of a dark channel prior and a convolutional encoder–decoder deep learning network with dual residual blocks to resolve different effects from different bad weather conditions. We validated our system on multiple surveillance videos by detecting vehicles with the You Only Look Once (YOLOv3) deep learning model and demonstrated that the computational time of our system could reach 30 fps on average; moreover, the accuracy increased not only by nearly 5% under low-contrast scene conditions but also 50% under rainy scene conditions. The results of our demonstrations indicate that our approach is able to detect vehicles under various bad weather conditions without the need to retrain a new model.

Keywords: intelligent traffic system; vehicle detection system; various bad weather conditions

1. Introduction

Numerous vehicle detection methods have been proposed to obtain trustworthy traffic data for the development of intelligent traffic systems. Briefly, we can split these methods into two categories: the first category adopts the classical machine learning models trained from manually defined features [1–6], such as Haar-like features and histogram of gradients (HoG) features with AdaBoost classifiers and support vector machines (SVM) classifiers; the other category comprises deep learning model-based approaches constructed from large amounts of labeling data [7–11]. Both categories can currently satisfy real-time requirements, yet the detection and classification accuracies of deep learning model-based approaches are still more robust than machine learning models constructed from manually defined features. Compared to the machine learning techniques in earlier decades, deep learning is more powerful due to its detection ability in abstraction.

Composed of multiple layers, the deep learning structure is able to extract potential features from training data and develop a more robust strategy, such as object detection tasks; it obtains higher detection rates than traditional machine learning models. Region-based Convolutional Neural Networks (R-CNN) [12] is one successful application of deep learning for object detection and classification tasks; it is comprised of a region proposal, feature extractor and classifier. The region proposal was accomplished through selective searching [13,14], the feature was extracted by AlexNet deep convolutional neural network (CNN) and the linear support vector machine was trained. To reduce the detection time, a spatial pyramid pooling network and region of interest pooling layer (RoI pooling) were adopted; then, the linear SVM was replaced by a fully connected layer, resulting in a new model named Fast R-CNN [15]. For further improvement, Faster R-CNN [16] completed the region proposal task through a new concept module, the Region Proposal Network (RPN), that functions as an attention mechanism. Although the new concept efficiently improved the computational time, it still only reaches a rate of 17 fps on a desktop computer platform, which is not sufficient for real-time applications.

YOLO (You Only Look Once) [17], a well-known model structure for its high-speed inference that is able to complete object detection and classification in real-time, has been published. The model first splits the input image into grid cells in which each cell is responsible for predicting the object of the one whose bounding box center is located in it. Following its main concept, the research team of YOLO adopted different technique concepts from other existing models, such as pre-defined anchor boxes, batch normalization, feature pyramid network, etc., and developed the second [18] and third versions of the model [19]. With an input size of 320×320 , YOLOv3 [19] is able to complete the inference in 22 ms at 28.2 mAP, which is three times faster than Single Shot MultiBox Detector (SSD) [20] with the equivalent accuracy. Under common scenarios, such as sunny or cloudy days, most of the above-mentioned deep learning-based detection methods can perform sufficiently well; however, the accuracy drastically decreases under various bad weather conditions such as rainy days or days with glare, which normally happens during sunsets, as shown in Figure 1. This issue may cause an intelligent traffic system to obtain incorrect data and make incorrect decisions that lead to serious problems, such as traffic accidents, improper control of traffic lights, etc. Smart traffic has become a modern trend, the development of which countries pay a great deal of attention to; however, under the influence of climate change, most countries are suffering from extreme weather conditions. For this reason, a detection system which has high accuracy is needed worldwide.

Figure 1. Extreme weather conditions.

As for the issue of vehicle detection under various bad weather conditions, there are a large number of existing research studies about image restoration for object detection. Li et al. [21] proposed a method that imposes priors, which are based on Gaussian mixture model learning, for background and rain streak layers. The main idea of this method is to decompose the rain streak from the captured image and obtain the recovered background. Luo et al. [22] introduced the shape prior technique to avoid the incorrect decomposition of the rain streak layer. Gary and Nayar [23] observed that the brightnesses of raindrops are approximately similar regardless of the background intensity. Based on this phenomenon, the candidate raindrops were detected and filtered by predefined constraints.

After computing the spatio-temporal correlation between the previous and following frames and determining the pixel locations of raindrops by the directions of correlation, the target pixels are restored by the interpolation of the pixel value from the previous and following frame. Shen and Xue [24] also located the positions of raindrops through intensity variation but restored the pixel value not only on the target pixel but also its neighborhood and obtained a greater smoothing result. Although the existing methods obtain satisfactory results in terms of removing the rain stripes, raindrops on lenses still seriously affect the vehicle appearance on the image frame. Furthermore, for a real-time system, only past frames can be taken as the reference frames; as a result, the restoration methods are not suitable for our application. Another approach for this issue is to collect specific data to retrain the deep learning model. Leung et al. [25] collected nighttime images by themselves and labeled all of them to develop a nighttime vehicle dataset. With the dataset, they trained the neural network and obtained a model that achieved the purpose of detecting vehicles with multiple classes. To deal with the nighttime scenario, the creation of a dataset for a specific purpose may work; however, there are many different extreme weather conditions, and building datasets for each of the cases does not tackle the problem at its roots. In addition, the hardware requirement of training a new deep learning model is much higher than just taking deep learning model as inference tasks. Further, training a deep learning model for all possible bad weather conditions suffers from huge computational times to converge the corresponding model weights and parameters; therefore, a visibility complementation module that can be combined with the exist vehicle detection method and can improve the detection accuracy for different weather conditions without retraining the classification model or deep learning model is needed.

To resolve the above-mentioned issues of vehicle detection under various bad weather conditions without the cost of completely retraining the deep learning models for different weather conditions, this paper proposes a new vehicle detection system with a visibility complementation module that is integrated with multiple deep learning techniques. The complementary module assesses the visibility and adopts different visibility correction methods depending on the assessment result. The module improves the detection accuracy under various bad weather conditions based on the unrestrained deep learning model. The considered parameters were not specific for local applications, such as the image contrast value and V value in the Hue-Saturation-Value (HSV) color space; as a result, the proposed method can be adapted in various traffic scenes and environments in different countries. We validated our system on multiple surveillance videos of different various bad weather conditions and proved that our system is able to detect vehicles under various bad weather conditions without retraining a new model; in other words, the proposed system overcomes the difficulty of detecting vehicles under various bad weather conditions through a pretrained deep learning model.

The remainder of this paper is organized as follows. In Section 2, we present details of the proposed vehicle detection system, including the vehicle detection module and visibility complementation module. In Section 3, we present the experimental and comparative results with both visualization and numerical examples. In addition, the computational time is also analyzed to demonstrate the real-time performance. Finally, the conclusion is given in Section 4.

2. The Proposed Method

The proposed system includes a visibility complementation module and vehicle detection module, as shown in the flow chart in Figure 2. The visibility complementation module is responsible for assessing the visibility of the current image frame and corrects the image frame through different deep learning-based approaches depending on the assessment results. The vehicle detection module is purely responsible for obtaining the vehicle position and the bounding boxes in the current image frame implemented based on a YOLOv3 [19] structure. The acronyms of the parameters used in the proposed method are shown in Table 1. The details of the two modules are described below.

Figure 2. System flowchart.

Table 1. Table of used acronyms.

Acronyms.	Description
$Image_c$	The contrast value of each pixel position in the input image.
μ_c	The mean value of the input image contrast result.
T_c	The accepted region of the mean contrast value for a determined low-contrast image frame.
T_v	The threshold for the determination of a high V value in the Hue-Saturation-Value (HSV) color space.
R_{high_v}	The ratio of high V value pixels to the entire number of pixels in the image.
T_{high_v}	The threshold of the ratio for the determination of low-contrast image frames.
$T_{component}$	The threshold of the number of components for the determination of adherent raindrop conditions.
K	User-defined value of the size of the confirmation reference frame.

2.1. Visibility Complementation Module

This module comprises an assessment unit and a correction unit. By observation, the most common reasons for the accuracy to drastically decrease are that vehicles are covered by haze and glare and that raindrops adhere to the camera's lens; therefore, the assessment unit evaluates the input image frame to determine the quality state and locate the adherent raindrop locations, forming a visibility state result. Depending on the visibility state result, the input image frame will be corrected through different visibility correction approaches. The details of the two main parts are described below.

2.1.1. Visibility Assessment

With regard to the quality state, we are concerned with the contrast and the amount of the pixels with high values in the entire image frame. If one of the statements is satisfied, the current image frame will be determined as a low-contrast image frame. To obtain the contrast property, we adopted the Laplace operator. The Laplace operator takes not only the specific direction but all of the neighboring pixels' gradients into account. By applying the Laplace operator to the whole image frame pixel by pixel, we can obtain the resultant contrast image, $Image_c$; then, the standard deviation SD of the whole image frame will be calculated with Equation (1):

$$SD(Image_c) = E[(Image_c - \mu_c)^2]^{1/2} \quad (1)$$

where $E()$ indicates the mean operation and μ_c represents the mean value of $Image_c$. If the standard deviation is located in the defined threshold region T_c , the visibility of the image frame may be affected by haze. The defined threshold region T_c in our experiment was set as [95, 110]; the sample result of the contrast assessment is shown in Figure 3. To confirm whether the glare from the sunset affects the visibility or not, in this study, we directly counted the number of pixels in which value in the HSV color space was larger than the defined threshold T_v . This threshold value was determined based on our experiments with 3265 collected partitions affected by sunset, and the resultant average values were all about 120. Thus, the threshold T_v was set as 120 to obtain satisfactory results. The sample images of the partitions affected by sunset are shown as Figure 4. If the ratio of the pixels with high values to the whole image frame pixels R_{high_v} reached the defined ratio threshold T_{high_v} , the image frame may have been affected by glare caused by sunset. In order to reach real-time requirements, the defined ratio threshold T_{high_v} in our experiment was set as 80%, which means that, unless almost the

whole image frame is covered by glare, effects due to glare will not be corrected. The low-contrast image frame decision formula is defined in Equation (2).

$$\text{Low contrast} = \left(T_{\text{lower_bound}} < \text{SD}(\text{Image}_c) < T_{\text{upper_bound}} \right) \parallel (R_{\text{high}_v} > T_{\text{high}_v}) \quad (2)$$

Figure 3. Contrast property assessment result.

Figure 4. Collected sample images of the partitions affected by sunset.

To locate the adherent raindrops, a scene base frame and several conditions are needed. The scene-based frame can be selected from any frame that is clear enough to recognize the objects in the scene, as shown in Figure 5. First, from observation, the partitions affected by adherent raindrops cause an increase in brightness, even becoming a bright spot on image frame; therefore, we obtained the proposal raindrop position through the difference between the current frame and scene base frame on the value plane in the HSV color space; the resulting image is shown in Figure 6b. Next, a medium blur operation was adopted for denoising so that the result could be more reliable; the result of this is shown in Figure 6c. Lastly, the numbers and locations of the prior raindrops were observed by applying a connecting component algorithm to the denoised image frame; the visualization result is shown in Figure 6d. Irrespective of the kind of scene, several scenarios that cause a variance of pixel values exist; as a result, a threshold $T_{\text{component}}$ for the number of components was set. Due to haze and fog appearing as a large connected area of pixel value variance, the number of the components calculated through the above steps was usually not very large; as a result, if the number of the prior raindrops was more than the defined threshold $T_{\text{component}}$, it was confirmed that some adherent raindrops existed on the camera lens. The threshold for $T_{\text{component}}$ was set as 3 in our experiment.

Figure 5. Sample image of the scene base frame.

Figure 6. Processing flow for locating adherent raindrops. (a) Input image frame with adherent raindrops. (b) Difference result between the current frame and scene base frame on the value plane under the HSV color space. (c) Denoising result through medium blur. (d) Connected component visualization result.

As we know, there are a large number of noises that might affect our location results, such as flashlights, moving trees and leaves, etc. To overcome this issue, we proposed a confirmation strategy that made our location result more robust. Based on the location results of each input frame, the confirmation strategy checked for the consistency against the previous $K - 1$ image frames, where K is an integer that represents how many frames will be confirmed together; unless K frames in the same position were all determined to show the presence of raindrops, the position was determined as a clear state pixel. Adherent raindrops will exist on the image frame for a long time, while others will

not; consequently, this method is efficient for locating the raindrops in a more accurate way. Figure 7 shows the result of assuming $K = 3$ for confirmation.

Figure 7. Concept of the confirmation strategy.

To ensure that the conditions and parameters in this research were widely supported, the proposed method was evaluated on multiple surveillance videos collected from roadside traffic surveillance systems set up in Kaohsiung, Keelung and Yangmingshan National Park in Taiwan. The analysis results are shown in Table 2. For this analysis experiment, we collected four scenes under different weather conditions, including glare, haze, sunny and rainy conditions. By applying the statement defined in the proposed method, the processed results demonstrated that the adopted parameters were able to obtain efficient results in various weather conditions.

Table 2. Parameters applied on different scenarios.

Scene Condition	Sequence	Frame Amount (Frames)	Avg. Variance	Ratio of Pixel Amount	Avg. Component Amount
Glare	Glare1	1425	11,035.14	3.1874%	1.5
	Glare2	1840	10,106.54	1.8145%	2
	Glare3	1780	9898.337	1.9105%	2.5
Haze	Haze1	1276	2151.5	91.4775%	1
	Haze2	135	2175.344	88.8172%	1.5
	Haze3	1276	943.6111	85.57%	2
Sunny	Sunny1	1780	7855.636	8.1367%	1.5
	Sunny2	900	14,106.04	14.9475%	1
	Rainy1	900	3871.039	31.5765%	4
Rainy	Rainy2	900	4926.419	32.217%	4
	Rainy3	900	4570.564	27.8319%	5

2.1.2. Visibility Correction

After receiving the visibility state result, the corresponding correction methods could be adopted. If the current image frame was noted as being low contrast, it was corrected through a dark channel prior [26]; to accelerate the correction processing, parallel processing was applied. The partitions affected by adherent raindrops were restored with deep learning restoration techniques. The details are described as below.

Image frames with haze or glare cover can be represented with the following equation:

$$I(x) = J(x)t(x) + A(1 - t(x)), \quad (3)$$

where I is the captured image frame, J is the scene information without noise, A is the global atmospheric light and t is the transmission from scene to camera. Fattal [26] proposed a scene information restoration method based on the formula above through Independent Component Analysis (ICA). Although the proposed method obtains good results, it might obtain unreliable results sometimes because of the non-satisfied statistically independent assumptions. The dark channel prior [26] estimates the transmission directly and improves the restoration reliability.

He et al. [26] observed that at least one of the color channels exists in most of the non-sky patches from the image frame, and the color channel has a very low intensity patch. This discovery can be represented with the formula below:

$$J^{dark}(x) = \min_{c \in \{r, g, b\}} \left(\min_{y \in \Omega(x)} (J^c(y)) \right), \quad (4)$$

where J^c is a color channel of J and $\Omega(x)$ is a local patch centered at x . J^{dark} is the dark channel of J , which is close to zero. After making some derivation of applying dark channel prior into the scene image formula between sky and non-sky regions, a robust haze removal method was proposed.

For our experiment, a 7×7 patch size was the most suitable setting for searching for the lowest intensity value in the target patch between RGB color channels. To satisfy the real-time requirements, we assigned haze removal computations to a graphics processing unit (GPU) and shortened the processing time. The GPU acceleration was implemented through PyTorch 1.4.0 [27], an open source machine-learning library based on the Torch library. We loaded the image patches as a tensor structure and obtained its minimum intensity by using PyTorch Application Programming Interface (API). After calculating the dark channel parameters, all source images were applied to the formula above; after calculating parallels with the GPU, we finally obtained the de-hazed image frames. The whole processing flowchart is shown in Figure 8. Figure 9 presents a sample image frame of a low-contrast state, and Figure 10 illustrates the visualization result of dark channels. In this figure, we can clearly recognize that the partitions affected by haze were correctly detected.

Figure 8. Graphics processing unit (GPU) acceleration flowchart.

Figure 9. Sample image of haze scene.

Figure 10. Low-intensity visualization result.

The partitions affected by adherent raindrops were restored through a dual residual network. Liu et al. [28] proposed an image restoration method based on dual residual networks which can be applied to noise removal, motion blur removal, haze removal, raindrop removal and rain-streak removal. For rainy day scenes, the main negative effects result from raindrops; therefore, we adopted the raindrop removal method in our proposed system. The raindrop removal network structure is a convolutional encoder–decoder network with dual residual blocks, resulting in a Dual Residual Network-S-P (DuRN-S-P) structure, as shown in Figure 11. The attention mechanism generated an attention map of the input image frame, which showed the raindrops' locations. Focusing on the raindrops' locations and restoring them, we could finally obtain the restoration image frame and were ready to detect the vehicles in the frame. Figure 12 presents the sample image of a rainy day scene, and Figure 13 shows the attention map, which is the visualization result of the attention mechanism.

Figure 11. Raindrop removal network structure.

Figure 12. Sample image of a rainy day scene.

Figure 13. Attention map of the rainy day scene sample image.

2.2. Vehicle Detection Module

To accomplish the real-time requirements and obtain accurate detection results under large-scale conditions, this study adopted YOLOv3 [19] as a detection method. YOLOv3 [19] uses Darknet-53 as its backbone, and the residual blocks in it can deal with vanishing and exploding gradient problems. To improve the capability of detecting small vehicle objects, the Feature Pyramid Network (FPN) was applied in the network structure. Through FPN, the network combined the information of global features in the shallow part and local features in the deep part of feature maps and developed a feature map with less information loss that made the objects easier to detect. Due to the different view-points of the image between public datasets and our application scenario, we collected the vehicle image dataset by ourselves in this study. Our collected datasets contained seven types of vehicles, including sedans, motorcycles, pedestrians, bicycles, trucks, buses and special purpose vehicles, and a total of over 100,000 labeled instances in 34,090 images with 1280×720 resolutions were collected. A sample image of our self-collected dataset is shown in Figure 14. After setting the network input image size to 416×416 and modifying the output dimension of each detection scale, the YOLOv3 [19] structure of our proposed method was as shown in Figure 15.

Figure 14. Sample images from our self-collected dataset.

Figure 15. You Only Look Once (YOLOv3) [19] network structure.

3. Experimental Results

In this section, the visualization results of the visibility correction and detection improvement under three different bad weather conditions are shown, and the comparative numerical results between adopting the proposed method or not are also calculated. The experimental platform of the proposed system was implemented on a Ubuntu 16.04.4 LTS (Xenial Xerus, Canonical Ltd., London, UK) 64 bit operating system with Python 3.5. The hardware environment was as follows: Intel® Core™ i7-7700K, Double Data Rate Fourth Generation(DDR4) 32G and a GeForce GTX 1070 Ti GPU.

3.1. Visualization Results

We cooperated with the Department of Intelligent Systems at the Institute for Information Industry, Taiwan. They provided a large number of various bad weather scenario sequences which were collected from the road side units, except for the haze scenario. Figure 16 shows the glare scene captured from the sequences collected from Keelung, Taiwan. The contrast was affected by the glare, and Figure 17 shows the corrected result based on our proposed method; Figure 18 is the original image frame of the haze scene condition downloaded from the Yangmingshan National Park's website [18], and Figure 19 shows the corrected result through our proposed method. The visibilities of the corrected images were apparently increased.

Figure 16. Glare scene: original image.

Figure 17. Glare scene: correction result.

Figure 18. Haze scene: original image.

Figure 19. Haze scene: correction result.

Figure 20 shows the original rainy scene frame captured from the sequences collected from Kaohsiung, Taiwan. Some raindrops adhered to the camera lens and caused distortion. Figure 21 shows the corrected result based on our proposed method.

Figure 20. Rainy scene: original image.

Figure 21. Rainy scene: correction result.

3.2. Performance Evaluation

We applied our system to surveillance videos and compared the detection performance between the YOLOv3 [19] model trained on the same training dataset. The results indicated that our proposed method effectively increased the detection accuracy and was able to reach the real-time requirements under extreme weather conditions.

Figures 22 and 23 present the detection results of YOLOv3 applied on the low-contrast scene image frame. Figures 24 and 25 show the detection results of our proposed system. The results showed that, under this condition, YOLOv3 could only detect a few vehicles; however, with our proposed system, the number of detected vehicles was significantly increased.

Figure 22. Low-contrast scene image frame detected by YOLOv3.

Figure 23. Low-contrast scene image frame detected by YOLOv3.

Figure 24. Low-contrast scene image frame detected by our proposed system.

Figure 25. Low-contrast scene image frame detected by our proposed system.

Figure 26 shows the detection result of YOLOv3 applied to a haze scene image frame, and Figure 27 shows the detection result of our proposed system. We can easily observe that YOLOv3 could not detect any vehicle in the image frame; in contrast, our proposed method could detect the black and white vans in the image frame. Under the same scene condition, the detection result of YOLOv3 did not detect any vehicle; in contrast, our proposed system detected a white van and marked it properly, and the details are shown in Figures 28 and 29.

Figure 26. Haze scene image frame detected by YOLOv3.

Figure 27. Haze scene image frame detected by our proposed system.

Figure 28. Haze scene image frame detected by YOLOv3.

Figure 29. Haze scene image frame detected by our proposed system.

Figures 30 and 31 show the detection results of YOLOv3 applied to the rainy scene image frame. Figures 32 and 33 show the detection result of our proposed system. As the figure shows, after the visibility correction, the vehicles far away and those covered by raindrops were able to be detected.

Figure 30. Rainy scene image frame detected by YOLOv3.

Figure 31. Rainy scene image frame detected by YOLOv3.

Figure 32. Rainy scene image frame detected by our proposed system.

Figure 33. Rainy scene image frame detected by our proposed system.

In terms of operation time, we tested our proposed system on three different high-definition (HD) sequences for each extreme weather condition. Glare and rainy testing sequences were downloaded from the surveillance system database, and haze testing sequences were downloaded from the Yangmingshan National Park's website [29]. Table 3 shows the testing results; most of the sequences could reach 30 fps and satisfied the real-time application requirements. Table 4 shows the comparative detection results of applying our proposed method or not on object detection using YOLOv3 [19] with the same pre-trained model.

3.3. Comparative Results and Discussion

Through our proposed visibility complementation module, the visibility problem in various bad weather conditions was compensated; as Table 4 shows, under glare scene conditions, the recall rate increased from 85.22% to 89.82%, and the false classification ratio decreased from 14.71% to 9.97%. Furthermore, the recall rate increased over 20% under rainy scene conditions. Under the haze condition, our proposed method could significantly improve the detection ability of traffic objects more than seven-fold; the amount of detected objects increased from 80 to 602. Although the computational time

under our testing sequences was able to reach the real-time requirements, it might decrease under high-resolution or complex raindrop conditions, as depicted in Table 3. The computational costs of the image contrast take a great deal of time for high-resolution image frames, and the large number of raindrops under complex raindrop conditions will cause iterative inference through the deep learning model and cost more computational time. The limited effectiveness was caused by the additional noise created during the generation of the non-noise scene; such additional noise affects the clear partitions in the original image frame and results in some bulk false detections and classifications. The background and vehicles were confused by the deep learning model in the generated scene image. As a result, a new complementary method for such haze scene conditions, which is able to preserve the recognition ability of the pre-trained deep learning model, is necessary for vehicle detection system applications in real-world traffic situations.

Table 3. Computation time of the system applied to different weather conditions.

Weather Condition	Low Contrast	Adherent Raindrops	Sequence Name	Frame Amount (Frames)	Total Time (Seconds)	Frames per Second
Glare	☒	☒	20191002_072701.avi	1779	61.25	29.06
	☒	☒	20191002_073357.avi	1779	60.56	29.41
	☒	☒	20191002_073456.avi	1779	59.46	30.30
Haze	☒	☒	media7.mp4	425	12.97	32.75
	☒	☒	media8.mp4	425	13.00	32.69
	☒	☒	media12.mp4	353	10.89	32.40
Rainy	☒	☒	20190813_083935.avi	899	29.15	31.05
	☒	☒	20190813_084746.avi	899	29.19	30.79
	☒	☒	20190813_090641.avi	899	29.40	30.57

Table 4. Detection performance comparison between the non-corrected image and corrected image.

	Scene	Bounding Box Amount	Positive	False Detection	False Classification
Glare	Without proposed	5702	4859 (85.22%)	4 (0.07%)	839 (14.71%)
	With proposed	8093	7269 (89.82%)	17 (0.21%)	807 (9.97%)
Haze	Without proposed	97	80 (82.47%)	10 (10.31%)	7 (7.22%)
	With proposed	760	602 (79.21%)	102 (13.41%)	65 (8.55%)
Rainy	Without proposed	650	435 (66.92%)	172 (26.46%)	43 (6.62%)
	With proposed	998	871 (87.27%)	52 (5.21%)	75 (7.52%)

4. Conclusions

The adoption of deep learning techniques for the development of intelligent traffic applications is a main trend in recent research studies, such as vehicle detection and recognition. Although the existing methods perform sufficiently well under general scenarios, the effects under various bad weather conditions are not ideal. As a result, the method for complementing these effects under such scenarios is important. This study proposed a vehicle detection system, including a visibility complementation module, which is able to detect and recognize vehicles under various bad weather conditions without retraining models and is useful for the development of intelligent traffic systems. The improvements in detection and recognition for most of the various bad weather scene conditions were quite obvious from our experiment. Our proposed system was able to increase the accuracy not only by nearly 5% under low-contrast scene condition but also by 50% under rainy scene conditions, which satisfied the real-time requirement. For some specific cases, such as haze conditions, more effort is needed to develop a more efficient method. In future, we will devote more time and energy to the creation of a better correction method which is general and adaptive enough to solve the great majority of bad weather conditions without mass predefined conditions.

Author Contributions: Investigation, X.-Z.C.; Methodology, C.-M.C.; Project administration, X.-Z.C., C.-M.C. and C.-W.Y.; Software, X.-Z.C. and C.-M.C.; Supervision, X.-Z.C., C.-W.Y. and Y.-L.C.; Validation, C.-M.C.; Visualization, C.-M.C.; Writing—original draft, X.-Z.C.; Writing—review & editing, Y.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the “New Generation Transport Deep Learning Intelligent System (2/4) Project” of the Institute for Information Industry, which is subsidized by the Ministry of Economic Affairs of the Republic of China; additionally, this research was funded by the Ministry of Science and Technology MOST-108-2218-E-027-017 and MOST-108-2221-E-027-066.

Acknowledgments: The collection of sequences for this research was supported by the members of the Embedded Vision Intelligent Laboratory at National Taipei University of Technology. Their contributions are greatly appreciated. This manuscript was edited by Chun-Ting, Chen. The addition of experiments in the manuscript revision was done by Chen-Pu, Cheng.

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

References

1. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001.
2. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 20–25 June 2005.
3. Lee, C.; Kim, J.; Park, E.; Lee, J.; Kim, H.; Kim, J.; Kim, H. Multi-feature Vehicle Detection Using Feature Selection. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013.
4. Sajib, M.S.R.; Tareeq, S.M. A feature based method for real time vehicle detection and classification from on-road videos. In Proceedings of the 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 22–24 December 2017.
5. Moghimi, M.M.; Nayeri, M.; Pourahmadi, M.; Moghimi, M.K. Moving Vehicle Detection Using AdaBoost and Haar-Like Feature in Surveillance Videos. *Int. J. Imaging Rob.* **2018**, *18*, 94–106.
6. Weia, Y.; Tianb, Q.; Guo, J.; Huang, W.; Cao, J. Multi-vehicle detection algorithm through combining Haar and HOG features. *Math. Comput. Simul.* **2019**, *155*, 130–145. [\[CrossRef\]](#)
7. Zhao, M.; Jia, J.; Sun, D.; Tang, Y. Vehicle detection method based on deep learning and multi-layer feature fusion. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018.
8. Li, S.; Lin, J.; Li, G.; Bai, T.; Wang, H.; Pang, Y. Vehicle type detection based on deep learning in traffic scene. *Procedia Comput. Sci.* **2018**, *131*, 564–572.
9. Song, H.; Liang, H.; Li, H.; Dai, Z.; Yun, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. *Eur. Transp. Res. Rev.* **2019**, *11*, 1–16. [\[CrossRef\]](#)
10. Murugan, V.; Vijaykumar, V.R.; Nidhila, A. A deep learning RCNN approach for vehicle recognition in traffic surveillance system. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 4–6 April 2019.
11. Mouna, B.; Mohamed, O. A vehicle detection approach using deep learning network. In Proceedings of the 2019 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), Tunis, Tunisia, 20–22 December 2019.
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
13. Uijlings, J.; van de Sande, K.; Gevers, T.; Smeulders, A. Selective Search for Object Recognition. *Int. J. Comput. Vision* **2013**, *104*, 154–171. [\[CrossRef\]](#)
14. Wang, X.; Yang, M.; Zhu, S.; Lin, Y. Regionlets for Generic Object Detection. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 17–24.
15. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015.

17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the CVPR, Las Vegas, NV, USA, 26 June–1 July 2016.
18. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
19. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016.
21. Li, Y.; Tan, R.T.; Guo, X.; Lu, J.; Brown, M.S. Single Image Rain Streak Decomposition Using Layer Priors. *IEEE Trans. Image Process.* **2017**, *26*, 3874–3885. [[CrossRef](#)] [[PubMed](#)]
22. Luo, Y.; Zhu, J.; Ling, J.; Wu, E. Fast Removal of Rain Streaks from a Single Image via a Shape Prior. *IEEE Access* **2018**, *6*, 60067–60078. [[CrossRef](#)]
23. Gary, K.; Nayer, S.K. Detection and Removal of Rain from Videos. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004.
24. Shen, M.; Xue, P. A fast algorithm for rain detection and removal from videos. In Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, Barcelona, Spain, 11–15 July 2011.
25. Leung, H.-K.; Chen, X.-Z.; Yu, C.-W.; Liang, H.-Y.; Wu, J.-Y.; Chen, Y.-L. A Deep-Learning-Based Vehicle Detection Approach for Insufficient and Nighttime Illumination Conditions. *Appl. Sci.* **2019**, *9*, 4769. [[CrossRef](#)]
26. He, K.; Sun, J.; Tang, X. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2341–2353. [[PubMed](#)]
27. PyTorch. Available online: <https://pytorch.org/> (accessed on 6 July 2020).
28. Liu, X.; Suganuma, M.; Sun, Z.; Okatani, T. Dual Residual Networks Leveraging the Potential of Paired Operations for Image Restoration. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
29. Yangmingshan National Park. Available online: https://www.ym SNP.gov.tw/main_ch/index.aspx (accessed on 2 June 2020).

© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).