International Journal of Geo-Information

MDPI

Article

An Efficient and Scene-Adaptive Algorithm for Vehicle Detection in Aerial Images Using an Improved YOLOv3 Framework

Xunxun Zhang [1,2,†] and Xu Zhu [3,*,†]

1 School of Civil Engineering, Xi'an University of Architecture and Technology, No. 13, Yanta Road, Xi'an 710055, China; zxx@xauat.edu.cn
2 National Experimental Teaching Center for Civil Engineering Virtual Simulation (XAUAT), No. 13, Yanta Road, Xi'an 710055, China
3 School of Electronic and Control Engineering, Chang'an University, Middle Section of Nan Erhuan Road, Xi'an 710064, China
* Correspondence: zhuxu_1987@sina.com or zx@chd.edu.cn
† These authors contributed equally to this work.

Received: 2 September 2019; Accepted: 21 October 2019; Published: 25 October 2019

Abstract: Vehicle detection in aerial images has attracted great attention as an approach to providing the necessary information for transportation road network planning and traffic management. However, because of the low resolution, complex scene, occlusion, shadows, and high requirement for detection efficiency, implementing vehicle detection in aerial images is challenging. Therefore, we propose an efficient and scene-adaptive algorithm for vehicle detection in aerial images using an improved YOLOv3 framework, and it is applied to not only aerial still images but also videos composed of consecutive frame images. First, rather than directly using the traditional YOLOv3 network, we construct a new structure with fewer layers to improve the detection efficiency. Then, since complex scenes in aerial images can cause the partial occlusion of vehicles, we construct a context-aware-based feature map fusion to make full use of the information in the adjacent frames and accurately detect partially occluded vehicles. The traditional YOLOv3 network adopts a horizontal bounding box, which can attain the expected detection effects only for vehicles with small length–width ratio. Moreover, vehicles that are close to each other are liable to cause lower accuracy and a higher detection error rate. Hence, we design a sloping bounding box attached to the angle of the target vehicles. This modification is conducive to predicting not only the position but also the angle. Finally, two data sets were used to perform extensive experiments and comparisons. The results show that the proposed algorithm generates the desired and excellent performance.

Keywords: vehicle detection; aerial image; improved YOLOv3 framework; context-aware-based feature map fusion; sloping bounding box

1. Introduction

Vehicle detection in aerial images is a vital component of an intelligent transportation system (ITS), which is particularly useful for traffic information gathering and road network planning. Compared with fixed ground cameras, the cameras equipped onto unmanned aerial vehicles (UAVs) have a broader perspective with 24-h, all-weather monitoring [1,2]. Recently, vehicle detection in aerial images has become a popular topic in the field of computer vision. However, because aerial images are low contrast and provide little vehicle information, vehicle detection in aerial images is difficult. Additionally, high efficiency is required because of the large number of vehicles in aerial images.

Moreover, vehicles that are close to each other are liable to cause lower accuracy and a higher detection error rate [3].

Vehicle detection in aerial images has been researched in abundance [4–7]. For instance, Cheng et al. designed a dynamic Bayesian network classification algorithm to detect vehicles in aerial images [8]. Michael et al. combined an adaptive boosting (AdaBoost) classifier and the sliding window to detect the vehicles [9]. Koga et al. [10] firstly chose the most informative training data by applying hard example mining (HEM) to stochastic gradient descent (SGD), and then they trained a convolutional neural network (CNN) for the vehicle detection in aerial images. Girshick proposed a fast region-based convolutional network method (Fast R-CNN) to efficiently classify targets using deep convolutional networks [11]. Overall, the main approach used in these methods is to first extract the features of the region of interest and then locate the position using a classifier. Therefore, such methods are collectively called two-stage target detection. They have high detection accuracy and strong generalization ability, but the determination of the region of interest requires high-volume computation, which leads to a poor ability to achieve real-time detection.

To improve the detection efficiency, Redmon et al. [12] proposed the YOLO network to regard object detection as a regression problem, which directly utilized the grids for the regression of the target position. Compared with the current convolutional neural networks, the YOLO network markedly increases the speed of detection velocity and is widely applied in many fields [13–15]. Compared with the R-CNN, the YOLO network locates the target vehicle with less accuracy and a lower recall rate. To improve the location accuracy and recall rate and thus improve detection accuracy, Redmon and Farhadi proposed the YOLO9000 network [16]. Unfortunately, the performance of the YOLO9000 network is still not satisfying for small targets [17]. To further improve the detection accuracy, YOLOv3 was developed using incremental improvement, which greatly enhanced the detection accuracy of small targets [18]. Because the vehicles in aerial images are relatively small, YOLOv3 is an ideal choice.

However, the YOLOv3 network cannot be directly employed for vehicle detection in aerial images. Firstly, the current YOLOv3 network always focuses on one frame image for vehicle detection but does not refer to the correlation issue of the adjacent frames. Adjacent frames represent the previous and following of the current frame, and they are spatially and temporally adjacent. In the process of capturing the aerial images, the camera keeps moving. In this situation, a shadowed or occluded vehicle in a frame is usually hard to detect. If we can introduce the information of adjacent frames, then occluded vehicles may be accurately detected. On the other hand, the YOLOv3 network adopts a horizontal bounding box, so it attains the expected detection effects only for the vehicles with small length–width ratio. For vehicles such as big trucks, oil tank trucks, and car carriers, which have high length–width ratios, the YOLOv3 network has difficulty describing the size of the target precisely [19]. For example, the ratio of the target and its circumscribed bounding box may be less than 50% when the slop angle of the target is 45°. Furthermore, if two vehicles are quite close to each other, then noise is generated by the high overlapping of the two horizontal bounding boxes for the two vehicles. In other words, the YOLOv3 network is prone to recognizing vehicles at a high density as one vehicle, leading to a low recall rate and a high miss rate. Therefore, it is essential to reform the bounding box of the YOLOv3 network to further improve the accuracy of vehicle detection in aerial images.

Recently, some researchers have constructed a sloping bounding box for remote sensing, and several achievements have resulted from this approach [20,21]. Mostly, they utilize the rotated region extraction and the pooling of the rotation region of interest to detect the rotated bounding box. For instance, Gong et al. firstly extracted the feature of the rotating rectangular candidate region and then achieved the label classification [22]. Liu et al. introduced the rotated region-based CNN for the target detection in aerial images, which can extract features of the rotated regions and locate rotated targets precisely [23]. Yang et al. proposed the rotation-dense feature pyramid networks (R-DFPN) to detect targets in different scenes effectively. Worrall et al. proposed a rotation convolution ensemble module to ensure the rotation invariance of the target detection [24]. On the whole, these methods are mostly based on two-stage target detection, where the rotated region is extracted and the rotation

region of interest is pooled to detect the rotated bounding box. Inevitably, the cascading position regression causes repeated calculation, leading to low detection efficiency.

Considering the above insights, we aimed to establish an efficient and scene-adaptive algorithm for vehicle detection in aerial images. Therefore, we designed an improved YOLOv3 framework. First, we proposed an improved YOLOv3 network composed of fewer layers compared with the traditional YOLOv3 network because only vehicles need to be detected in our work. Next, a context-aware-based feature map fusion was constructed to make full use of the information of the adjacent frames. Thus, the feature maps of the adjacent frames are combined to construct a feature map for the current frame. Furthermore, to ensure the rotation invariance of vehicle detection, a sloping bounding box attached to the angle of the target vehicles is proposed.

As a note, the proposed vehicle detection algorithm is applied for the aerial images, which includes not only the still images but also videos composed of consecutive frame images. The remainder of our work is organized as follows. The workflow of the vehicle detection algorithm in aerial images is presented and detailed in Section 2. Section 3 shows the experimental and comparisons to demonstrate the validity and superiority of the proposed vehicle detection algorithm. Finally, the conclusion is summarized in Section 4.

2. Methodology

As previously described, the YOLOv3 network is the most representative technology for vehicle detection in aerial images. First, it extracts the features using the Darknet-53 network to obtain the feature map, and then it sets the grid cell to the same size as the feature map. For each grid cell, three bounding boxes are predicted. However, since only vehicles need to be detected in our work, a very deep neural network is not essential. Therefore, as illustrated in Figure 1, we constructed an improved YOLOv3 framework that has fewer layers and is composed of three components: feature map generation, feature map fusion, and prediction.

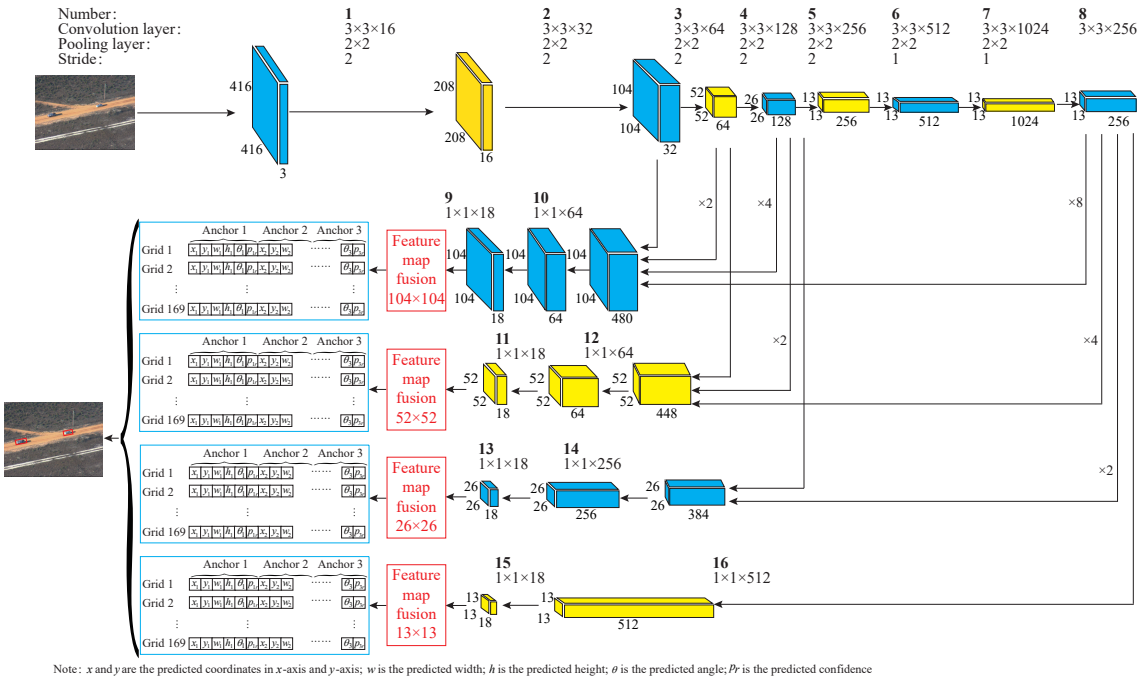

Figure 1. The diagram of the improved YOLOv3 framework with four scales.

For the feature map generation, we put forward an improved YOLOv3 framework with only 16 layers to meet real-time demands.

Additionally, to further improve the accuracy of detecting small vehicles, we constructed feature maps of four scales: i.e., 13 × 13, 26 × 26, 52 × 52, and 104 × 104. Then, a context-aware-based

feature map fusion was constructed to combine the information of the adjacent frames to form a new feature map that is conducive to improving the detection accuracy, especially for the occluded vehicles. Moreover, to ensure rotation invariance, we designed the sloping bounding box to obtain the prediction result, which contains the angle that guarantees rotation invariance.

Compared with the traditional YOLOv3 network, a more simple structure with fewer layers is constructed. Besides, the feature map fusion is added to the improved YOLOv3 framework. For the output, the traditional YOLOv3 network outputs only the location information (i.e., x, y, w, h, and p_r), while the improved YOLOv3 framework outputs not only the location information but also the angle information (i.e., x, y, w, h, θ, and p_r).

2.1. Feature Map Generation

As shown in Figure 1, the improved YOLOv3 framework has four scales and is composed of only 16 layers. A convolutional layer of 3×3 is utilized to extract the feature map, and a pooling layer is used to reduce the dimensionality of the feature information.

Compared with the YOLOv3 network in [18], where a neural network with 53 layers was built, the improved YOLOv3 framework can further reduce the computational burden by using fewer layers. Furthermore, in the traditional YOLOv3 network, there are three scales of feature maps for object detection, namely, 13×13, 26×26, and 52×52. However, vehicles in aerial images are relatively small. Therefore, we expanded the number of scales to four: 13×13, 26×26, 52×52, and 104×104. By drawing on the idea of Densenet [25], the feature fusion of different layers is carried out by upper sampling, and the original image was resized into $416 \times 416 \times 3$ as the input. As shown in Figure 1, "$\times 2$", "$\times 4$", and "$\times 8$" indicate that the step size of the corresponding upper sampling is 2, 4, and 8, respectively.

Furthermore, the K-means clustering method is utilized to determine the anchor boxes of the improved YOLOv3 framework. In the proposed framework, there are 12 anchor boxes: (12, 16), (16, 24), (21,32), (21, 41), (24, 51), (33, 51), (28, 62), (39, 64), (35,74), (44, 87), (53, 105), and (64, 175). By introducing 12 anchor boxes, vehicles that are too small or too large can be effectively detected.

However, the occlusion problem is difficult to overcome if we directly utilize the four feature maps to predict the vehicles in aerial images. In this situation, a vehicle in a frame that is shadowed or occluded is usually hard to detect. Introducing the information of adjacent frames allows for the accurate detection of occluded vehicles. Therefore, the information of the adjacent frames should be added to the current frame. Next, context-aware-based feature map fusion is detailed.

2.2. Context-Aware-Based Feature Map Fusion

In this part, we focus on the context-aware-based feature map fusion, which aims to make full use of the information of the adjacent frames. First, we need to determine whether two frames should be fused, and the basis of this decision is the fusion critical condition. Then, feature map fusion is realized by the linear iteration.

2.2.1. Fusion Critical Condition

Before the feature map fusion, we should first determine whether the current frame and the adjacent frame (i.e., the previous frame or the next frame) can be fused. This work applies the histogram equalization method to compute the similarity between the current and the adjacent frames. For this purpose, the Bhattacharyya distance [26] is utilized to compute the similarity according to histogram equalization:

$$D(p, p') = \sum_{i=1}^{N} \sqrt{p(i)p'(i)} \tag{1}$$

where $D(p, p')$ represents the similarity between the current frame and the adjacent frame, and p and p' are the histogram equalizations of the current frame and the adjacent frame, respectively. N

represents the number of pixels in each frame, and $p(i)$ and $p'(i)$ are the corresponding ith histogram equalizations of the current frame and the adjacent frame, respectively. If the similarity $D(p, p')$ is smaller than the pre-defined threshold value d^t, then the corresponding frames can be fused; otherwise, they are not fused.

2.2.2. Feature Map Fusion

After the derivation of the fusion critical condition, linear iteration is applied for feature map fusion. For the kth frame F_k, the fused feature map is shown in Figure 2:

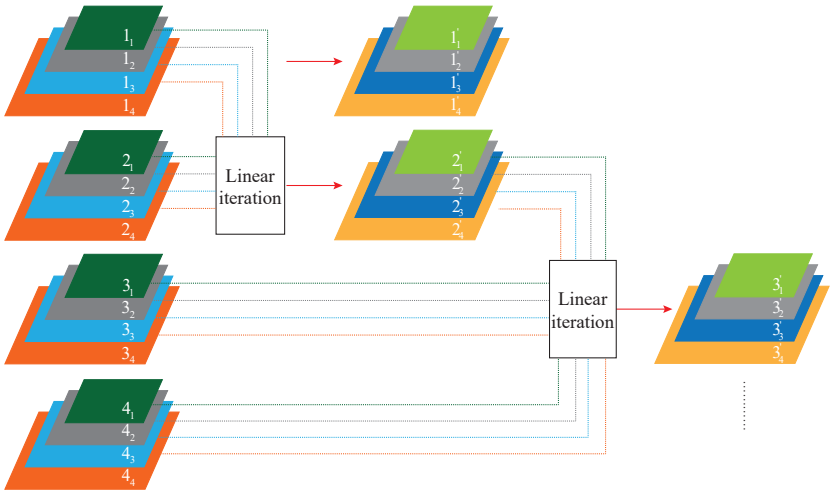

Figure 2. The diagram of the feature map fusion.

$$\overline{F}_k^l = \begin{cases} F_1^l & k = 1 \\ F_2^l + \omega \times F_1^l & k = 2 \\ \omega \times F_{k+1}^l + F_k^l + \omega \times \overline{F}_{k-1}^l & k \geq 3 \end{cases} \tag{2}$$

where l denotes the scale of the improved YOLOv3 framework. F_k^l represents the feature map of the kth frame with the scale l. ω is the fusion weight between the current frame and the adjacent frames. $\overline{F}_k^l$ represents the fused feature map of the kth frame with the scale l.

In Equation (2), the fusion weight ω plays a crucial role in the feature map fusion. In our algorithm, we utilize mutual information entropy to determine the fusion weight ω.

$$\omega = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \cdot \omega_d \tag{3}$$

where μ and σ represent the mean value and standard deviation of the Gauss function, respectively, x denotes the mean value of the similarity for an aerial image sequence, and ω_d is the weight.

Then, mutual information entropy is used to determine the value of x. For two images F_a and F_b, their information entropies can be represented as:

$$H_a = -\sum_{i=0}^{255} P_a(i)\log_2 P_a(i), \quad H_b = -\sum_{i=0}^{255} P_b(i)\log_2 P_b(i) \tag{4}$$

where H_a and H_b represent the information entropies of the images F_a and F_b, respectively. $P_a(i)$ and $P_b(i)$ denote the probability of the gray scale i in the two images F_a and F_b, respectively.

$$H(a,b) = -\sum_{a,b} P_{ab}(a,b)\log_2 P_{ab}(a,b) \tag{5}$$

where H_a and H_b represent the information entropies of the images F_a and F_b, respectively. $P_{ab}(a, b)$ denotes the intensity joint probability distribution.

Mutual information entropy can be computed as:

$$I(a, b) = H_a + H_b - H(a, b) \tag{6}$$

According to Equations (5) and (6), $H(a, b)$ presents the shared information in both F_a and F_b. The less the shared information is, the smaller the value of $H(a, b)$ is, and the bigger the value of $I(a, b)$ is.

Next, to obtain the parameter x, we carry out normalization for mutual information entropy:

$$x = I(a, b)/I(a, a) \tag{7}$$

So far, context-aware-based feature map fusion is completed by using the fusion critical condition and feature map fusion. For feature map fusion, mutual information entropy is utilized to determine the fusion weights.

2.3. Sloping Bounding Box Based Prediction

In this section, to overcome the flaws of the traditional horizontal bounding box, we introduce the sloping anchor box, which is intended for the prediction of the target position and angle. Introducing the sloping anchor box renders the proposed YOLOv3 module more sensitive to the angle. First, the sloping bounding box is presented, followed by a multi-task loss function with angle classification punishment.

2.3.1. Sloping Bounding Box

The YOLOv3 network utilizes the horizontal bounding box to locate the position of the targets. Generally, the bounding box is represented by a four-dimension vector $\{b_x, b_y, b_w, b_h\}$, where (b_x, b_y) represents the center of the bounding box, and b_w and b_h are the width and height, respectively. Specifically, b_x, b_y, b_w, and b_h are computed as:

$$\begin{cases} b_x = \sigma(t^x) + c_x \\ b_y = \sigma(t^y) + c_y \\ b_w = p_w e^{t_w} \\ b_h = p_h e^{t_h} \end{cases} \tag{8}$$

where $\sigma(\cdot)$ represents the sigmoid function, p_w and p_h are the width and height of the predefined anchor box, respectively, and (c_x, c_y) are the coordinates of the top left corner of the anchor box. However, it is difficult to describe the rotated target if the length–width ratio is large. On the other hand, when the two vehicles are close to each other, the bounding boxes of two vehicles that are close to each other are prone to interlocking. Therefore, a sloping bounding box is used to locate the vehicles accurately.

As illustrated in Figure 3, we redefine a sloping bounding box in the shape of a red rectangle, which is represented as a five-dimension vector $\{b_x, b_y, b_w, b_h, \theta\}$. (b_x, b_y) represents the centre of the anchor box, b_w and b_h are the width and height, respectively, and θ is the orientation angle. In the 5-dimension vector, θ is defined as the angle between x-axis and the side with positive slope.

For simplicity, the prediction of the angle can be transformed into a classification problem. In other words, the predicted angle should be selected from a flash of candidate angles belonging to $[0, \pi)$. A crucial step is the determination of candidate angles.

For the candidate angles, the selection criteria are to ensure fewer classifications and smaller IoU, which is defined as:

$$IoU = \frac{area(P \cap G)}{area(P \cup G)} \tag{9}$$

where P and G represent the bounding box and the ground truth, respectively, and $area(\cdot)$ denotes the area of the corresponding region.

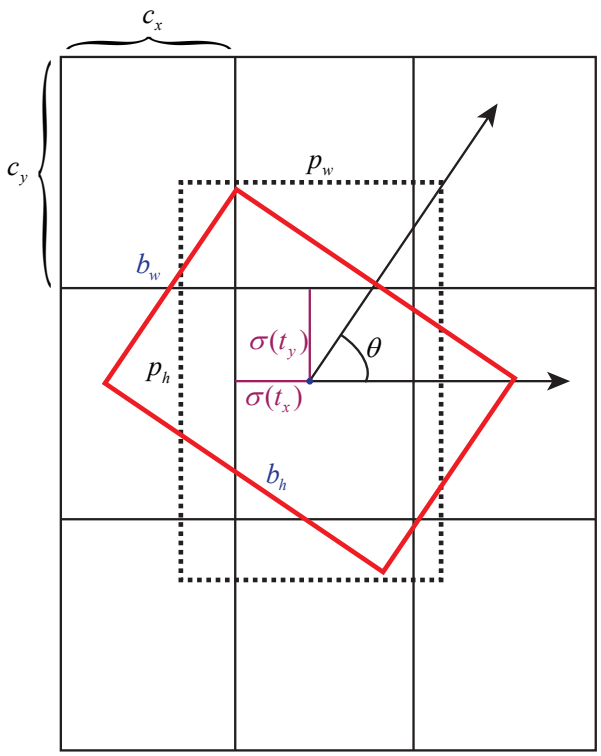

Figure 3. The sloping anchor box.

Moreover, the step length is defined as $\pi/step$, and $step$ should be more than 4. This are mainly because there are at least four directions for the vehicles based on the cluster analysis of the vehicle angle.

By introducing the sloping bounding box, the prediction result of our proposed framework can be represented as a six-dimension vector $\{b_x,\ b_y,\ b_w,\ b_h,\ \theta,\ p_r\}$, where p_r is the predicted confidence.

2.3.2. Multi-Task Loss Function with Angle Classification Punishment

For the target detection network, designing the loss function is crucial. The improved YOLOv3 network establishes a multitask loss function to optimize target detection. Therefore, compared with the traditional YOLOv3 network, two improvements are implemented in the proposed network. On one hand, the angle classification punishment is added to the multi-task loss function, which aims to punish the error of the predicted angle and ground truth. Furthermore, the multi-task loss function is computed by combining the initial output and the final output. On the whole, we designed the multi-task loss function by combining the networks of YOLOv3, R2CNN, and PCN. Specifically, position and confidence errors are represented by the loss of mean square error (MSE), while the angle classification error is represented by the loss of the cross entropy.

$$\begin{aligned} L(b_x,\ b_y,\ b_w,\ b_h,\ \theta,\ p_r) = {} & \lambda_{crd} \sum_{i=1}^{s^2} \sum_{j=1}^{A} m_{ij}^{obj} [(x_i^g - x_i)^2 + (y_i^g - y_i)^2] \\ & + \lambda_{crd} \sum_{i=1}^{s^2} \sum_{j=1}^{A} m_{ij}^{obj} [(w_{ij}^g - w_{ij})^2 + (h_{ij}^g - h_{ij})^2] \\ & + \lambda_{prb} \sum_{i=1}^{s^2} \sum_{j=1}^{A} m_{ij}^{obj} [p_{ij}^g \log p_{ij} + (1 - p_{ij}^g) \log(1 - p_{ij})] \\ & + \lambda_{cls} \sum_{i=1}^{s^2} \sum_{j=1}^{A} m_{ij}^{obj} CrossEntropy(\theta_i^g,\ \theta_i) \end{aligned} \tag{10}$$

where s^2 represents the number of the grids in the feature map; A is the number of the anchor boxes; λ_{crd}, λ_{prb}, and λ_{cls} are the corresponding weights; and m_{ij}^{obj} is a Boolean variable. If the center of the ground truth (x_i^g, y_i^g) is in the ith anchor box, then $m_{ij}^{obj} = 1$; otherwise, $m_{ij}^{obj} = 0$. $CrossEntropy(\cdot)$ represents the loss of the cross entropy for the angle classification and is computed as:

$$CrossEntropy(\theta_i^g, \theta_i) = -\sum \theta_i^g \log\theta_i \tag{11}$$

where θ_i^g and θ_i represent the labelled sample and the output of the network, respectively.

In a word, by introducing the context-aware-based feature map fusion, the detection accuracy can be effectively improved. Furthermore, the sloping bounding box is designed to ensure the rotation invariance of vehicle detection. In this way, the efficient and scene-adaptive vehicle detection in aerial images can be achieved.

3. Experiments and Analysis

As described in this section, the proposed vehicle detection algorithm was tested on two data sets, namely, the DARPA VIVID and OIRDS data sets. The DARPA VIVID data set was collected at Eglin Air Base in the DARPA VIVID program, and it is composed of five visible image sequences. The OIRDS data set, it is composed of aerial images instead of image sequences. All the experiments described in this section were implemented in MATLAB on an Intel 8 Core-i7 (3.20 GHz) system with 16 GB of RAM. Noticeably, these aerial images were captured in different conditions so that they can meet the demands of the experimental verification.

In the vehicle detection algorithm, we firstly aim to determine the pre-defined threshold for the feature map fusion. Then, the angle step should be defined for the angle prediction. Next, the experiments on the two data sets are presented to verify the effectiveness and acceptability of the proposed vehicle detection algorithm in aerial images.

3.1. Determination of the Fusion Weight for the Feature Map Fusion

In our vehicle detection algorithm, feature map fusion plays a vital role, and the fusion weight ω in Equation (2) is the key factor. In the determination of the fusion weight ω, the first frame is chosen as the current frame, and then mutual information entropy is utilized to compute mutual information entropy. Figure 4 presents the mutual information entropy between the current frame and other frames.

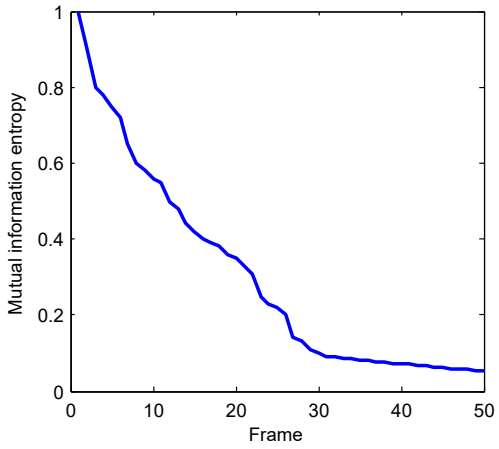

Figure 4. Mutual information entropy between the current frame and other frames.

As illustrated in Figure 4, the mutual information entropy is larger when the current frame and some frame is close. In other words, when the two frames are spatially and temporally close to each other, the corresponding mutual information entropy is large.

The fusion weight ω is obtained by first computing the mutual information entropies and their normalized values for the two adjacent frames. Then, according to Table 1, the mean value of the normalized values is assigned to x. Finally, the fusion weight can be obtained according to Equation (3). In our work, $\sigma = 0.6$, $\mu = 0.6$, and $\omega_d = 1$. Therefore, $x = 0.7109$ and $\omega = 0.6536$.

Table 1. Mutual information entropies and their normalizations for the two adjacent frames.

Frames	Mutual Information Entropy	Normalization
1	5.3673	0.7886
2	4.2117	0.7967
3	4.7344	0.8552
4	5.3724	0.8725
5	4.1674	0.8336
...	...	...
72	5.5330	0.7083
73	3.7809	0.8037

3.2. Determination of the Angle Step

In the improved YOLOv3 framework, we introduce the sloping bounding box to predict not only the position but also the angle of the target vehicles. The angle prediction is treated as a classification problem. There are two main reasons: First, YOLOv3 network is essentially a multi-label classification. It is more convenient to construct the improved YOLOv3 framework by treating the angel prediction as a classification problem people need games; second, YOLOv3 network is composed of multiple convolution layers. Regression formulation brings high computation complexity due to its complex model. Besides, the accuracy of regression formulation relies heavily on large samples. Therefore, the angle prediction is treated as a classification problem. In other words, the predicted angle θ should be selected from the set $0 : step : \pi$, where $step$ denotes the step length. A step that is too large may reduce the accuracy of vehicle detection, and one that is too small requires excessive computation and leads to low efficiency. Therefore, the approach to selecting the step becomes a key factor. The selection criteria of the step for the candidate angles are a low number of classifications and a small IoU.

In our work, there are two criteria: one is to minimize the number of classifications, and the other is to ensure that $IoU < 0.3$ in a step. Moreover, the step length $step$ should be smaller than $\pi/4$.

As illustrated in Figure 5, IoU is still larger than 0.3 in a step even when the length–width ratio is 7:1. Therefore, the chosen step is $\pi/5$, and the angle set is $\{0,\ \pi/5,\ 2\pi/5,\ 3\pi/5,\ 4\pi/5\}$.

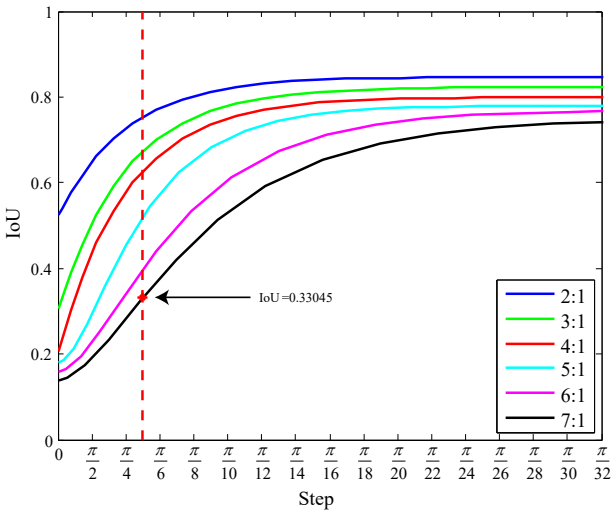

Figure 5. Relations between the step and IoU with different length–weight ratios.

3.3. Performance on the Two Data Sets

The proposed vehicle detection algorithm was tested on two data sets, namely, the DARPA VIVID and OIRDS data sets. First, the improved YOLOv3 framework is trained, which is an essential step. Owing to the high sampling frequency, the adjacent frames of the video sequences are quite similar. If all the aerial images are utilized for training, the vehicle samples are redundant. Therefore, we selected 400 aerial images to train the YOLOv3 framework. Specifically, the images are different in vehicle number, color, and shape.

For testing, we chose eight aerial images from the two data sets: egtest01, egtest02, egtest03, and egtest04 from the DARPA VIVID data set, and dataset1, dataset4, dataset10, and dataset11 from the OIRDS data set. The performance of the proposed vehicle detection algorithm on the eight test aerial images is partially presented in Figures 6 and 7, where the ground truth and the detected vehicles are indicated by green and red rectangles, respectively. Figures 6 and 7 compare the detection results of the proposed algorithm with those of the Fast-CNN and YOLOv3 modules. It is observed that the detection results of our algorithm are better than those of the others. In particular, as shown in Figure 7, our proposed vehicle detection algorithm can detect vehicles at a high density, while the other methods show poor performances.

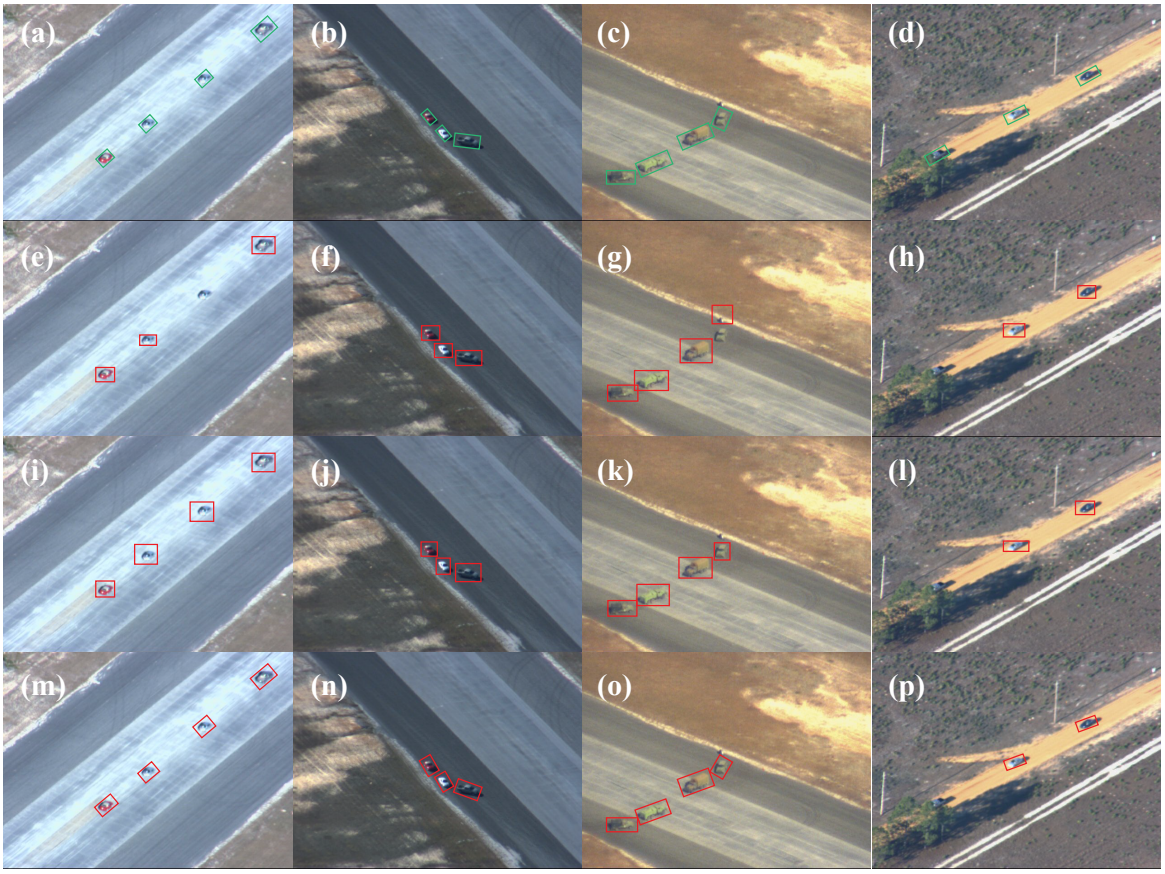

Figure 6. Comparison of the detection results for the VIVID data set. (**a**–**d**): The ground truth, (**e**–**h**): Fast convolutional neural network (Fast-CNN) module, (**i**–**l**): YOLOv3 module, (**m**–**p**): Improved YOLOv3 framework.

To further evaluate the performance of the proposed vehicle detection algorithm in aerial images, three universal measurements are introduced as:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

where TP and FP represent the true positive (the number of vehicles that are correctly detected) and false positive (the number of vehicles that are incorrectly detected), respectively. In other words, *Precision* means the percentage of the correctly detected vehicles over all the detected vehicles.

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

where FN represents the false negative (the number of other objects that are wrongly treated as vehicles), and *Recall* means the percentage of the correctly detected vehicles over all the true vehicles.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

where the $F1 - Score$ is a trade-off between *Precision* and *Recall*.

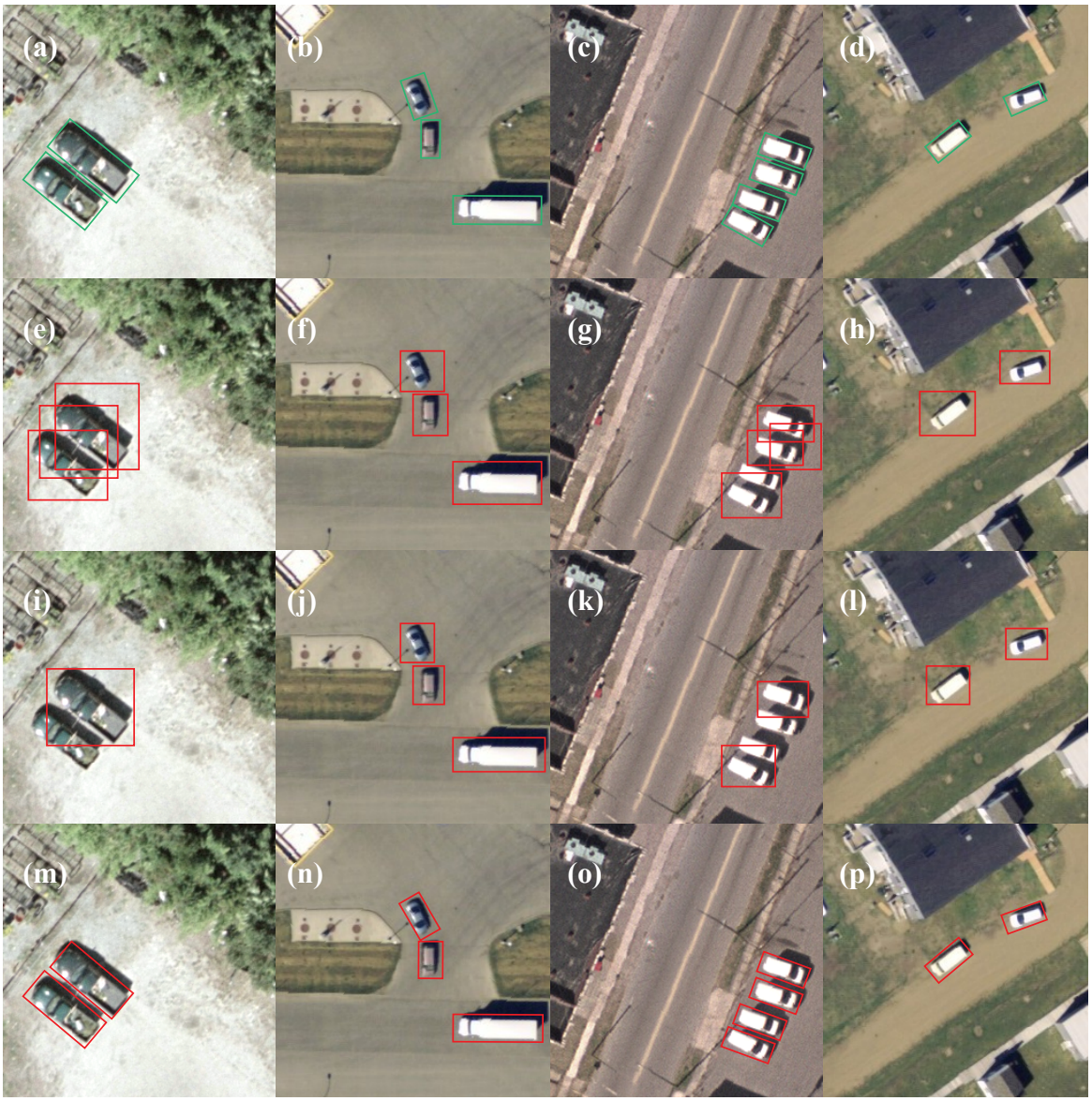

Figure 7. Comparison of the detection results for the OIRDS data set. (**a**–**d**): The ground truth, (**e**–**h**): Fast-CNN module, (**i**–**l**): YOLOv3 module, (**m**–**p**): Improved YOLOv3 framework.

As shown in Figures 6 and 7, only partial detection results are presented. To quantitatively evaluate the performance of the proposed algorithm, we perform the statistical analysis of *Precision*,

Recall, and $F1-Score$ based on Equations (12)–(14). The quantitative indicators on the eight subsets from the two data sets are displayed in Table 2.

Compared with the Fast-CNN method, the average *Precision*, *Recall*, and $F1-Score$ of our algorithm are increased by 8.34%, 9.06%, and 9.29%, respectively. Similarly, compared with the YOLOv3 module, the average *Precision*, *Recall*, and $F1-Score$ of our algorithm are increased by 4.57%, 5.06%, and 4.95%, respectively. This quantitative information demonstrates that the proposed vehicle detection algorithm has sufficient advantages on vehicle detection in aerial images.

Furthermore, to evaluate the effectiveness of sloping bounding box on the vehicle detection, we also perform the statistical analysis of *IoU* based on Equation (9). Table 3 shows the quantitative indicators on the eight subsets from the two data sets. Compared with Fast-CNN and YOLOv3 methods, the average *IoU* of our algorithm increase by 11.8% and 15.2%, respectively.

Table 2. Quantitative results on the four test image sequences.

	Fast-CNN Module			**YOLOv3 Module**			**The proposed Algorithm**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
VIVID_egtest01	87.0%	91.0%	89.0%	91.1%	94.2%	91.9%	95.3%	98.3%	96.5%
VIVID_egtest02	88.0%	90.4%	89.2%	90.9%	93.5%	92.9%	94.9%	98.7%	97.1%
VIVID_egtest03	90.4%	90.0%	90.2%	93.3%	93.8%	93.7%	97.7%	98.2%	97.6%
VIVID_egtest04	88.7%	89.2%	89.0%	91.9%	92.6%	93.3%	96.4%	97.1%	98.1%
OIRDS_dataset1	89.9%	88.1%	88.9%	92.5%	91.9%	92.8%	97.3%	96.3%	96.9%
OIRDS_dataset4	88.3%	89.0%	90.1%	91.3%	92.2%	93.7%	95.1%	96.6%	97.8%
OIRDS_dataset10	89.3%	88.7%	89.1%	92.1%	91.4%	93.1%	96.6%	97.7%	98.7%
OIRDS_dataset11	87.9%	89.1%	88.9%	91.7%	92.8%	92.3%	95.8%	96.8%	97.9%
Average	88.7%	89.4%	89.3%	91.9%	92.8%	93.0%	96.1%	97.5%	97.6%

Table 3. *IoU* of the Fast-CNN module, YOLOv3 module, and improved YOLOv3 framework (s).

Video Sequences	**Fast-CNN Module**	**YOLOv3 Module**	**The Proposed Algorithm**
VIVID_egtest01	0.77	0.75	0.85
VIVID_egtest02	0.72	0.69	0.83
VIVID_egtest02	0.80	0.75	0.87
VIVID_egtest02	0.76	0.77	0.89
OIRDS_dataset1	0.65	0.63	0.76
OIRDS_dataset4	0.59	0.55	0.68
OIRDS_dataset10	0.63	0.58	0.70
OIRDS_dataset11	0.55	0.52	0.62
Average	0.68	0.66	0.76

To test the efficiency of the proposed vehicle detection algorithm in aerial images, we compare its processing time with that of the Fast-CNN module [27]. Table 4 compares the processing time for the eight subsets from the two data sets. Compared with the Fast-CNN module and YOLOv3 module, the average processing time of our algorithm is shortened by 43.66% and 29.44%, respectively. In conclusion, according to Figures 6 and 7, and Tables 2 and 4, the proposed vehicle detection algorithm in aerial images is effective and efficient overall.

Moreover, in the improved YOLOv3 network, the context-aware-based feature map is introduced to detect the partially occluded vehicles. To verify its effectiveness, we choose four aerial images egtest04 from the DARPA VIVID. Then, the performance of the proposed vehicle detection algorithm on the four test aerial images is partially presented in Figure 8. As shown in Figure 8n, the vehicle in the lower-left corner, which is partially occluded by the trees, is detected via the context-aware-based feature map fusion. The similar issue is presented for the vehicle in the upper-right corner in Figure 8p. As a whole, compared with the Fast-CNN module and YOLOv3 network, the improved

YOLOv3 network can detect the partially occluded vehicles owing to the context-aware-based feature map fusion.

Table 4. Processing time of the Fast-CNN module, YOLOv3 module, and improved YOLOv3 framework (s).

Video Sequences	Fast-CNN Module	YOLOv3 Module	The Proposed Algorithm
VIVID_egtest01	14.5	11.2	10.6
VIVID_egtest02	28.6	28.2	16.3
VIVID_egtest02	35.4	27.0	19.2
VIVID_egtest02	39.3	28.6	20.7
OIRDS_dataset1	17.8	13.2	11.5
OIRDS_dataset4	26.3	18.6	13.5
OIRDS_dataset10	26.8	25.4	14.9
OIRDS_dataset11	25.9	19.3	14.3
Average	26.8	21.4	15.1

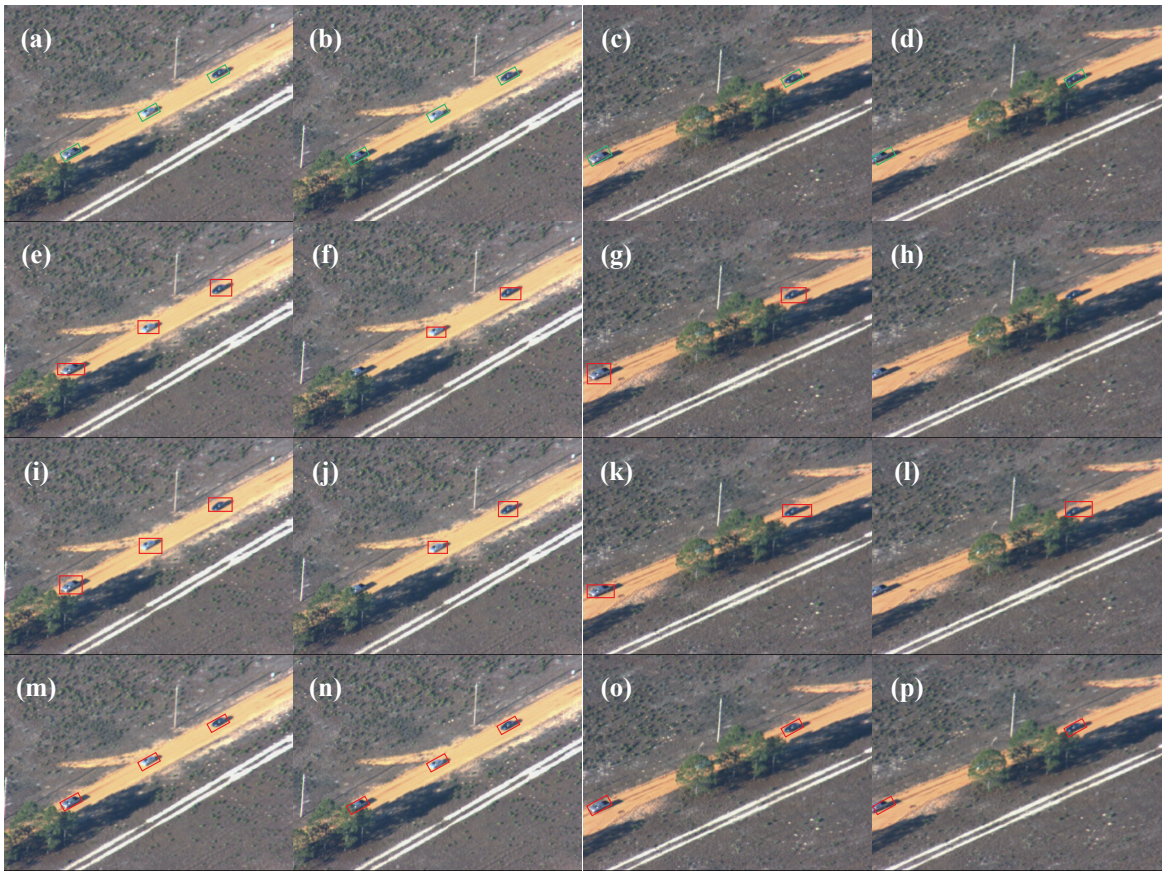

Figure 8. Comparison of partial occluded vehicles detection for the VIVID data set. (**a**–**d**): Ground truth, (**e**–**h**): Fast-CNN module, (**i**–**l**): YOLOv3 module, (**m**–**p**): Improved YOLOv3 framework.

From the above, the proposed vehicle detection algorithm shows the competitive and superior detection performances in aerial images. Especially, based on the detection results in different frames, vehicle tracking can be promoted. Furthermore, by combining the positions in different frames and the motion of the UAV, the results can provide technical support for behavior analysis. In other words, vehicle detection is the basic part of traffic detection and information collecting, and it can promote vehicle tracking and behavior analysis.

4. Conclusions

In this study, we present an efficient and scene-adaptive algorithm for vehicle detection in aerial images using an improved YOLOv3 framework. Rather than directly using the YOLOv3 network, we constructed a new structure with fewer layers to improve the detection efficiency. By introducing the context-aware-based feature map fusion, we combined the feature maps of the adjacent frames with the current frame to increase the accuracy of vehicle detection. Rather than utilizing the traditional bounding box, a sloping bounding box is used to ensure the rotation invariance of the proposed framework, especially for vehicles at a high density and those with large length–width ratio. Finally, the experimental results demonstrate the superiority of our proposed algorithm for vehicle detection compared to the other two state-of-the-art methods.

Author Contributions: Xunxun Zhang conceived and designed the experiments; Xunxun Zhang performed the experiments and wrote the paper; Xu Zhu analyzed the data; Xu Zhu contributed reagents/materials/analysis tools.

Funding: This work was supported by the the China Postdoctoral Science Foundation [grant number 2017M613030].

Acknowledgments: The authors are grateful for the constructive suggestions from anonymous reviewers that significantly enhance the presentation of this paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsor has the role in the analyses, interpretation of data, and in the decision to publish the results.

References

1. Zhang, S.; He, G.; Chen, H.B.; Jing, N.; Wang, Q. Scale Adaptive Proposal Network for Object Detection in Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 864 –868. [CrossRef]
2. Tao, H.; Lu, X. Automatic smoky vehicle detection from traffic surveillance video based on vehicle rear detection and multi-feature fusion. *IET Intell. Transp. Syst.* **2019**, *13*, 252–259. [CrossRef]
3. Zhou, H.; Wei, L.; Lim, C.P.; Creighton, D.; Nahavandi, S. Robust Vehicle Detection in Aerial Images Using Bag-of-Words and Orientation Aware Scanning. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1–12. [CrossRef]
4. Lin, C.; Chen, S.Y.; Chen, C.C.; Tai, C.H. Detecting newly grown tree leaves from unmanned-aerial-vehicle images using hyperspectral target detection techniques. *ISPRS J. Photogramm. Remote Sens.* **2018**, *142*, 174–189. [CrossRef]
5. Wang, J.; Wang, X.; Ke, Z.; Cai, Y.; Yue, L. Small UAV Target Detection Model Based on Deep Neural Network. *J. Northwest. Polytech. Univ.* **2018**, *36*, 258–263. [CrossRef]
6. Sun, M.; Li, Y.; Qiang, C. Automatic Urban Vehicle Detection from Airborne LiDAR Data with Aerial Image. *Remote Sens. Technol. Appl.* **2014**, *29*, 886–890.
7. Hu, X.; Xu, X.; Xiao, Y.; Chen, H.; He, S.; Qin, J.; Heng, P.A. SINet: A Scale-Insensitive Convolutional Neural Network for Fast Vehicle Detection. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1010–1019. [CrossRef]
8. Cheng, H.Y.; Weng, C.C.; Chen, Y.Y. Vehicle detection in aerial surveillance using dynamic Bayesian networks. *IEEE Trans. Image Proc.* **2012**, *21*, 2152–2159. [CrossRef] [PubMed]
9. Teutsch, M.; Kruger, W. Robust and fast detection of moving vehicles in aerial videos using sliding windows. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 26–34.
10. Koga, Y.; Miyazaki, H.; Shibasaki, R. A CNN-Based Method of Vehicle Detection from Aerial Images Using Hard Example Mining. *Remote Sens.* **2018**, *10*, 124. [CrossRef]
11. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1–9.
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
13. Wang, H.; Zhang, Z. A vehicle real-time detection algorithm based on YOLOv2 framework. In Proceedings of the Real-time image and video processing in Industrial and IoT Applications using Big Data Analytics, Mantra, Gold Coast, Australia, 20–22 June 2018; p. 10670.

14. Sommer, L.W.; Schuchert, T.; Beyerer, J. Fast Deep Vehicle Detection in Aerial Images. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Shanghai, China, 18–21 May 2017; pp. 311–319.
15. Saribas, H.; Cevikalp, H.; Kahvecioglu, S. Car detection in images taken from unmanned aerial vehicles. In Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU), Cesme, Izmir, 2–5 May 2018; pp. 1–4.
16. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
17. Kim, K.J.; Kim, P.K.; Chung, Y.S.; Choi, D.H. Performance Enhancement of YOLOv3 by Adding Prediction Layers with Spatial Pyramid Pooling for Vehicle Detection. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; pp. 1–6.
18. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Shi, X.; Shan, S.; Kan, M.; Wu, S.; Chen, X. Real-Time Rotation-Invariant Face Detection with Progressive Calibration Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2295–2303.
20. Hsu, G.S.; Ambikapathi, A.M.; Chung, S.L.; Su, C.P. Robust license plate detection in the wild. In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
21. Ali, W.; Abdelkarim, S.; Zahran, M.; Zidan, M.; Sallab, A.E. YOLO3D: End-to-end real-time 3D Oriented Object Bounding Box Detection from LiDAR Point Cloud. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 1–12.
22. Gong, C.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415.
23. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 900–904.
24. Worrall, D.E.; Garbin, S.J.; Turmukhambetov, D.; Brostow, G.J. Harmonic Networks: Deep Translation and Rotation Equivariance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5028–5037.
25. Zhang, Z.; Liang, X.; Dong, X.; Xie, Y.; Cao, G. A Sparse-View CT Reconstruction Method Based on Combination of DenseNet and Deconvolution. *IEEE Trans. Med. Imaging* **2018**, *37*, 1407–1417. [CrossRef] [PubMed]
26. Ratha, D.; De, S.; Celik, T.; Bhattacharya, A. Change Detection in Polarimetric SAR Images Using a Geodesic Distance Between Scattering Mechanisms. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1066–1070. [CrossRef]
27. Liu, X.; Tao, Y.; Jing, L. Real-Time Ground Vehicle Detection in Aerial Infrared Imagery Based on Convolutional Neural Network. *Electronics* **2018**, *7*, 78. [CrossRef]