remote sensing

MDPI

Article

A Dual-Input Moving Object Detection Method in Remote Sensing Image Sequences via Temporal Semantics

Bo Wang [1,2], Jinghong Liu [1,2,*], Shengjie Zhu [1,2], Fang Xu [1] and Chenglong Liu [1]

[1] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; wangbo202@mails.ucas.ac.cn (B.W.); zhushengjie19@mails.ucas.ac.cn (S.Z.); xufang59@126.com (F.X.); liuchenglong@ciomp.ac.cn (C.L.)

[2] University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: liujinghong@ciomp.ac.cn

Abstract: Moving object detection in remote sensing image sequences has been widely used in military and civilian fields. However, the complex background of remote sensing images and the small sizes of moving objects bring great difficulties for effective detection. To solve this problem, we propose a real-time moving object detection method for remote sensing image sequences. This method works by fusing the semantic information from a single image extracted by the object detection branch with the motion information of multiple frames extracted by the motion detection branch. Specifically, in the motion detection branch, we design a motion feature enhancement module (MFE) to improve the interframe motion information. Then, we design a Motion Information Extraction network (MIE) to extract motion information. Finally, the moving object information is directly output by fusing the motion and semantic information extracted by the object detection branch. Based on the experimental results of the two datasets, the proposed method achieves an accuracy rate of 93.21%, a recall rate of 92.72%, an average frame rate of 25.25 frames (fps), and a performance of 96.71% in terms of AP@0.5. The performance of the proposed method is better than that of other methods, and the overall detection effect is better; therefore, it meets the needs of the detection task.

Keywords: remote sensing image sequences; moving camera; moving object detection (MOD); convolutional neural network (CNN)

check for updates

Citation: Wang, B.; Liu, J.; Zhu, S.; Xu, F.; Liu, C. A Dual-Input Moving Object Detection Method in Remote Sensing Image Sequences via Temporal Semantics. *Remote Sens.* **2023**, *15*, 2230. https://doi.org/10.3390/rs15092230

Academic Editor: Pedro Melo-Pinto

Received: 23 March 2023

Revised: 17 April 2023

Accepted: 19 April 2023

Published: 23 April 2023

1. Introduction

With the rapid development of aerospace remote sensing technology, the application range of remote sensing images has become extensive. Moving object detection (MOD) in aerial remote sensing image sequences is one of the most critical tasks in military surveillance and civil navigation. The MOD task refers to the detection of change regions in an image sequence and the extraction of the moving object from the background image. The extracted moving object can provide a reference area for subsequent tasks, such as action recognition [1,2], tracking [3–7], behavior analysis [8], and intelligent video surveillance [9,10]. Recently, advances in UAV technology have made it easy to acquire remote sensing images, and relatively inexpensive UAVs have a wide range of applications. Therefore, as the use of moving cameras grows, so does the need to detect moving objects, which makes it critical to develop robust moving object detection methods for moving cameras.

MOD tasks can be divided into static and motion camera tasks depending on whether the camera is in motion. Currently, research on static camera tasks is more mature [11–13], but more research on motion camera tasks is required [14,15]. Compared with a static camera, a stationary object appears to be in motion in the image sequence obtained by a moving camera due to the movement of the camera. In remote sensing image sequences, camera motion relative to the ground can be characterized as translation, rotation, and scaling. When using a MOD algorithm designed for static cameras on an image sequence obtained from a moving camera, the algorithm will fail due to the appearance of motion in the image background. For motion cameras, MOD methods need to consider not only all problems that arise when the camera is stationary, but also the difficulties caused by camera motion compensation. Additionally, motion targets in remote sensing image sequences are typically far from the shooting platform, which leads to problems such as small target sizes and inadequate object color and texture features. Moreover, the target is susceptible to background interference, which leads to noise and motion blur, making detection difficult. Therefore, detecting motion targets in remote sensing image sequences involves many challenges and difficulties. Here, we present these challenging problems in detail.

Moving background: In a static camera, only the moving objects appear to be moving, but in the case of a moving camera, everything seems to be moving as the camera moves. Additionally, parts of the scene disappear over time. In this case, separating the moving objects from the stationary background becomes more difficult.

Shadows: The shadow of a moving object is often treated as the background, rather than the moving object itself. However, many classical moving object detection algorithms fail to separate the shadow from the moving object itself.

Few target pixels: In the remote sensing image sequence, the field of view is large. The moving target occupies very few pixels, requiring the moving target detection algorithm to focus more on small targets.

Complex background: There are many irrelevant backgrounds with similar characteristics to moving targets, as well as static targets with the same characteristics as moving targets in remote sensing images. Therefore, it is necessary to use motion and the semantic features of moving objects for moving object detection.

In recent decades, many MOD algorithms for stationary cameras have been proposed, but there has been relatively little research on MOD algorithms that directly target moving cameras. For motion target detection under camera motion conditions, traditional MOD methods include three main types: the frame difference method, the optical flow method, and the background subtraction method [16–20]. The frame difference method usually compensates for moving backgrounds before detecting them using a static camera. This method is simple to calculate, but it is susceptible to changes in lighting conditions and has high requirements for background motion compensation algorithms. The optical flow method is accurate, as it can calculate the magnitude and direction of the displacement of each pixel in the image. However, it requires a lot of computation and cannot achieve real-time calculation. The background subtraction method identifies the moving target by subtracting the background model from the current frame. However, it requires the construction of a robust background model. The method is the most commonly used motion target detection method. It achieves motion target detection by constructing a universal background, such as a mixture of Gaussian models and the kernel density estimation [21,22]. However, its application effect in the case of background movement is not ideal. These algorithms cannot adapt well to the needs of complex scenes and require a better balance between robustness and real-time performance. In addition, these methods can only detect the contours of the motion target and cannot obtain the precise coordinates and categories of each motion target.

With the rise of deep learning, the object detection method based on the convolutional neural network (CNN) has been widely used in various fields [23–25]. Two-stage algorithms, such as R-CNN, Fast R-CNN, and Faster R-CNN [26–28], which achieve state-of-the-art performances in terms of accuracy, and the YOLO series [29–34] of end-to-end algorithms, which achieve the highest detection speed, have been proposed.

Motivated by the success of CNNs in target detection tasks, CNNs have been applied to MOD tasks with some good results. Compared with traditional methods, CNN-based MOD methods show significant advantages in terms of their accuracy and robustness. In [35], F. LATEEF et al. proposed a CNN-based moving target to detect vehicular video sequences. This consisted of two deep learning networks, an encoder–decoder-based network (ED-Net), and a semantic segmentation network (Mask R-CNN). Mask R-CNN detects the objects of interest, and ED-Net classifies their motion (moving/static) in two consecutive frames. Finally, the two networks are combined to detect the motion targets. In [36], C. Xiao et al. proposed a motion target detection method for satellite video. They proposed a two-stream detection network divided into dynamic and static fusion networks (DSF-Net). The DSF-Net extracts static contextual information from single frames and dynamic motion cues from consecutive frames. Finally, the two networks are fused in layers for moving target detection. In [37], H. ZHU et al. proposed a framework consisting of coarse-grained detection and fine-grained detection. A connected region detection algorithm is used to extract moving regions, and then a CNN is used to detect more accurate coordinates and identify classes of objects. In [38], J. ZHU et al. combined a background compensation method to detect moving regions with a neural network-based approach to localize moving objects accurately, and finally, fused the two results to determine the moving target. In [39], D. Li et al. used CNNs exclusively for single-strain estimation and target detection. Motion information is used in consecutive frames as an additional input to the target detection network to effectively improve the detection performance.

As mentioned above, traditional motion detection methods can achieve good detection accuracy levels. However, they are time-consuming and can only obtain regional information instead of specific target information, such as the object class and location. The CNN-based approach solves this problem by enabling the network to determine the specific location and target class of the moving target. Most CNN-based algorithms divide the MOD task into two steps: first, they extract motion information, compensate for or extract optical flow information from the moving background, and then input it into the object detection network to obtain motion targets. These methods have achieved good results, but they require additional preprocessing, resulting in a complex detection process. The computation cost of optical flow information is expensive, making it unsuitable for remote sensing image sequences. Therefore, under stationary camera conditions, MOD methods cannot be well applied to moving camera scenes. Traditional image-registration-based MOD methods rely heavily on the accuracy of the registration algorithm. In contrast, deep-learning-based algorithms often need to extract motion information as an auxiliary input to the network.

To address the above problems, inspired by the two-stream network combining motion and semantic information, we propose an end-to-end MOD method for an aerial remote sensing image sequence, which fuses enhanced motion information based on dual frames with semantic information based on a single frame. Specifically, we introduce a Motion Feature Enhancement (MFE) module to extract and enhance motion information between two frames preliminarily. Then, the enhanced motion features are fed into a 3D convolutional Motion Information Extraction (MIE) network to further extract motion features at different convolution depths. Finally, the extracted motion information is fused with semantic features based on a single frame at corresponding feature levels and decoded by a detection head to identify moving targets. In summary, the main contributions are as follows:

(1) We propose a feasible MOD model, motion feature enhancement module, and motion information extraction network (MFE-MIE), which enhances motion features by feature differences and then extracts multilayer motion information using 3D convolution and finally fuses the multilayer motion information with the corresponding semantic information to accurately detect moving targets. The proposed method is powered entirely by CNN and implements end-to-end detection. Only two frames are input to realize the detection without additional auxiliary information. In addition, it meets the requirements of MOD task real-time detection.

(2) We propose an MFE module, which highlights motion features between two frames and calculates the difference between features. This difference is used as a weight to suppress the background to highlight the foreground. The MFE module can enhance the motion features and provide clues for subsequent processing.

(3) We designed an MIE network that uses three groups of lightweight improved 3D convolutional layers to extract further motion information from the enhanced motion information. Compared with directly using 3D convolution to extract motion information, our MIE network achieves more lightweight detection with a better detection effect.

(4) We conducted a series of experiments to validate the effectiveness of the proposed method. Experiments on two benchmark datasets demonstrated that our MFE-MIE outperforms other state-of-the-art models, achieving real-time detection and meeting the needs of the detection tasks.

The rest of this paper is organized as follows: Section 2 presents the details of the proposed MOD model. In Section 3, we first introduce the adopted dataset and the data processing methods used, and then the proposed method is evaluated and discussed through several experiments. Section 4 presents the discussion. Finally, the conclusions are given in Section 5.

2. Methods

In this section, we first present the overall structure of the proposed method. Then, we introduce the proposed MFE module and the MIE network.

In our work, we propose a moving object detection method named MFE-MIE in which motion information based on two frames and object information based on a single frame are fused. The overall framework of the method is shown in Figure 1. It is composed of three main parts: motion detection, object detection, and temporal–space fusion. For the motion detection module, a pair of registered images, $\mathrm{T_k}$ and $\mathrm{T_{k-n}}$, are first integrated with motion information by the proposed motion feature enhancement (MFE) module and then encoded by the proposed motion information extraction (MIE) network, which consists of improved 3D convolutions and obtains three layers of motion features with different convolutional depths. The object detection module uses the generic CSPDarknet53 [31] object feature extraction network to extract three layers of object semantic information at different convolutional depths. The temporal–space fusion module fuses the motion information extracted by the motion detection branch and the object semantic information extracted by the object detection branch at the corresponding level to facilitate the subsequent extraction of moving objects. Finally, the fused information is decoded using two convolutional layers to output the motion target information.

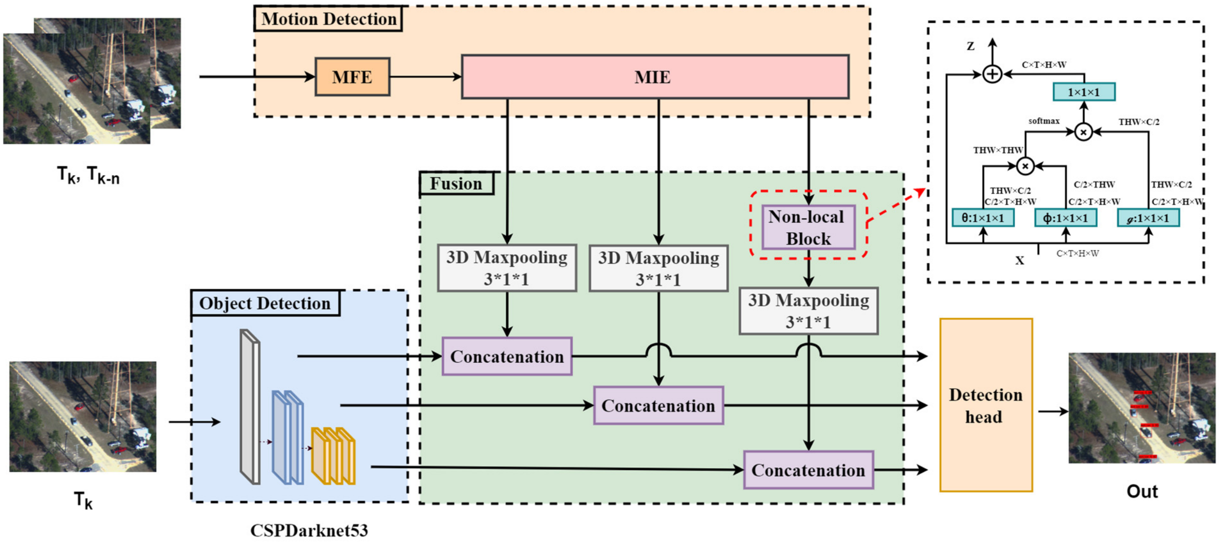

Figure 1. Overall architecture of the MFE-MIE framework.

2.1. Motion Detection Module

Previous motion detection methods usually only used the simple difference between two frames or the overhead of extracting the optical flow to obtain motion feature information during operation. However, these two operations are less adaptive to motion target detection tasks and often result in insufficient motion information extraction or a huge overhead. To solve this dilemma, a motion detection network consisting of the MFE module and the MIE network is proposed in this paper. It aims to effectively capture the motion information between two frames and provide detection cues to identify moving targets. The MFE module is responsible for integrating and enhancing the motion information between the two images. The MIE network is responsible for further extracting the motion information between the outputs of the features with the MFE module.

2.1.1. MFE Module

The framework of the MFE module is shown in Figure 2. It enhances the motion features between two frames, initially extracts the motion information, and suppresses the background information. It was inspired by the frame difference method [40]. We extend the difference between two frames of the frame difference method to the difference between the feature map of two frames to make it more applicable to the network. Note that our goal is to find a motion representation paradigm that can help to identify motion efficiently, rather than calculating the exact pixel displacement between two frames, as in optical flow. Therefore, we only use two RGB frames as the input and do not add precomputed optical flow images.

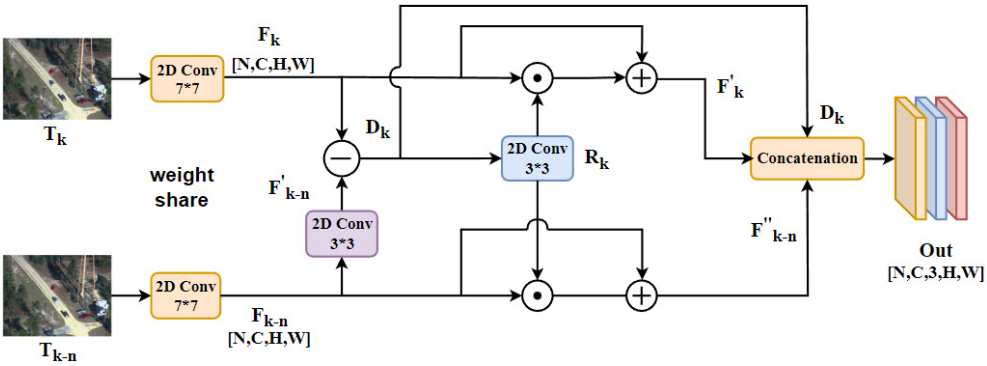

Figure 2. Motion feature enhancement (MFE) module.

First, the two input images are encoded by two weight-sharing 2D convolution modules as

$$F_k = Conv_{7\times7}(T_k),\ F_k \in \mathbb{R}^{N\times C\times 1\times H\times W} \tag{1}$$

$$F_{k-n} = Conv_{7\times7}(T_{k-n}),\ F_{k-n} \in \mathbb{R}^{N\times C\times 1\times H\times W} \tag{2}$$

where T_k and T_{k-n} denote the 3D tensor of the two input images, $Conv_{7\times7}(\cdot)$ represents a convolutional layer with a 7 × 7 filter, a batch normalization layer, and a ReLU activation function, F_k, F_{k-n} represents the feature map of the k frame and the feature map of the $(k-n)$ frame, respectively. Not only is the moving target moving relative to the background between the two frames, there is also relative motion between the background and the moving camera. We continue to use the convolutional layer transform feature to compensate for the background motion for the $(k-n)$ frame feature to obtain F'_{k-n}. This procedure can be denoted as

$$F'_{k-n} = Conv_{3\times3}(F_{k-n}) \tag{3}$$

where $Conv_{3\times3}(\cdot)$ denotes a convolutional layer with a 3 × 3 filter, a batch normalization layer, and a ReLU activation function. Then, the difference in the feature maps between two frames was used to approximately represent the motion significance. We calculate the difference between the two feature maps by an element-wise subtraction operation, followed by taking the absolute value of the difference as

$$D_k = \left| F_k - F'_{k-n} \right| \tag{4}$$

where D_k represents a motion difference feature, and $|\cdot|$ denotes an absolution operation. D_k contains rich motion information, so we further put it through a 2D convolutional layer with 3 × 3 filters followed by a sigmoid function, which transforms the motion features into weights R_k. Then, the motion features of the two feature maps F_k and F_{k-n} are enhanced with an element-wise multiplication operation. The original feature maps are added to the enhanced features to obtain the refined motion features, which suppress the background and highlight the foreground. This process can be illustrated as

$$R_k = Conv_{3\times3}(D_k) \tag{5}$$

$$F'_k = F_k \oplus (F_k \otimes R_k) \tag{6}$$

$$F''_{k-\mathrm{n}} = F_{k-\mathrm{n}} \oplus (F_{k-\mathrm{n}} \otimes R_k) \tag{7}$$

where $\oplus$ and $\otimes$ are element-wise addition and multiplication operations, respectively.

Finally, we concatenate F'_k, F''_{k-n}, and D_k along the time dimension. This can be represented as

$$Out = Cat(F'_k, F''_{k-n}, D_k),\ Out \in \mathbb{R}^{N\times C\times 3\times H\times W} \tag{8}$$

where $Cat(\cdot)$ is a feature concatenation operation along the time dimension, and Out denotes the output feature map of size $N \times C \times 3 \times H \times W$, where N is the batch size, C is the number of channels in the feature map, 3 is the number of feature maps, and H and W are the feature map's height and width.

Since the shallow features of the image mainly contain the precise position and motion information while the deep features mainly contain semantic information, the MFE module is only used for the shallow features and the refined motion features are extracted using an improved 3D convolution implementation. In addition, all filters used in the MFE module are essentially 2D convolutional kernels, so it is lightweight.

2.1.2. MIE Network

For the 3D convolution-based motion information extraction (MIE) network, as shown in Figure 3, the output of MFE module $Out \in \mathbb{R}^{N\times C\times 3\times H\times W}$ is fed to a self-developed 3D network, which consists of three 3D convolutional layers, to generate motion features $F_i \in \mathbb{R}^{N\times C_i\times 3\times H_i\times W_i}$ with three different convolutional depths. Since the computational overhead of 3D convolutional blocks is too expensive, to increase the speed and reduce the cost of computation, we use three convolutional layers with kernels 3 × 1 × 1, 1 × 3 × 3, and 1 × 1× 1, instead of each 3D convolutional block.

First, we decompose a 3D convolutional layer with 3 × 3 × 3 filters into a 1D temporal convolutional layer with 3 × 1 × 1 filters and a 2D spatial convolutional layer with 1 × 3 × 3 filters. Then, the features are integrated through a convolutional layer with a 1 × 1 × 1 filter. Finally, to further reduce the computational cost, the number of channels in the second and third 3D convolutions is halved and then adjusted using the convolutional layer with 1 × 1 × 1 filters. The details of the MIE network are shown in Table 1.

We denote the spatiotemporal size by $C \times T \times S^2$, where C is number of channels, T is the temporal length, and S is the height and width of a square spatial crop. The MIE network was inspired by SlowFast [41]. It takes $T = 3$ frames of feature maps as the input, which are the k frame feature, the $(k - n)$ frame feature, and the feature difference between them. No temporal down-sampling is performed in this example, because it would be harmful to do so when the number of input frames is small, so the output of each improved 3D convolutional block has a temporal dimension of $T = 3$.

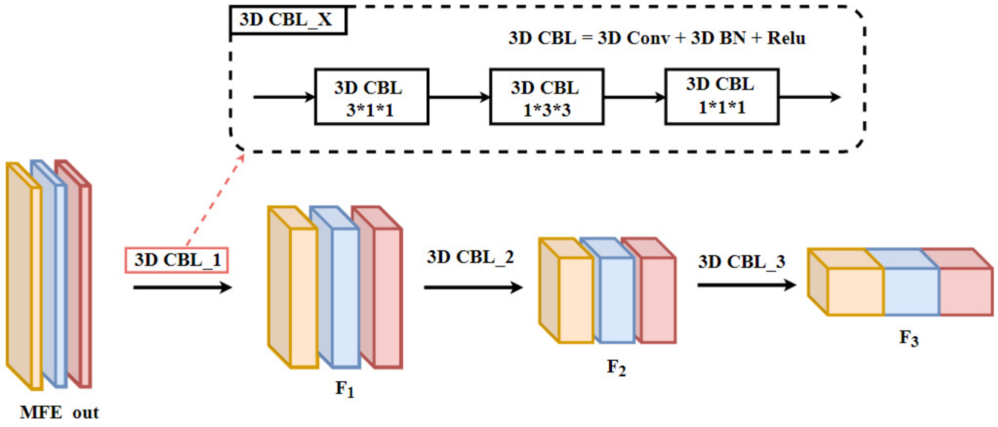

Figure 3. Motion information extraction network (MIE).

Table 1. The details of the MIE network.

Stage		**Kernel Dimension** $T \times S^2, C$	**Output Sizes** $C \times T \times S^2$
Input		\	$3 \times 2 \times 608^2$
MFE		\	$64 \times 3 \times 152^2$
MIE	3D CBL_1	$\begin{bmatrix} 3 \times 1^2 & 64 \\ 1 \times 3^2 & 64 \\ 1 \times 1^2 & 256 \end{bmatrix}$	$256 \times 3 \times 76^2$
	3D CBL_2	$\begin{bmatrix} 3 \times 1^2 & 256 \\ 1 \times 3^2 & 128 \\ 1 \times 1^2 & 512 \end{bmatrix}$	$512 \times 3 \times 38^2$
	3D CBL_3	$\begin{bmatrix} 3 \times 1^2 & 512 \\ 1 \times 3^2 & 256 \\ 1 \times 1^2 & 1024 \end{bmatrix}$	$1024 \times 3 \times 19^2$

Note that the designed MIE network is lightweight and computationally efficient, since it consists of only three layers of improved 3D convolution blocks. Moreover, since the 3D convolution can extract the motion information from the features, we can extract more effective motion cues from the designed MIE network.

2.2. Object Detection Module

For the MOD task, the motion features between multiple frames can help to discover the boundaries of moving targets. The semantic information of a single image can help to locate the locations of moving targets and identify object classes. Therefore, single-frame-based target semantic information is also vital for the MOD task. Since YOLO series algorithms have satisfactory results in object detection, we use the backbone network CSPDarknet53 in the object detection module to extract semantic information at three different convolutional depths from a single image frame. CSPDarknet53 consists of five CSP modules. Each CSP module is down-sampled by a convolutional layer with 3×3 filters, and we use the output of its last three layers as the output features of the object detection branch.

2.3. Temporal–Space Fusion Module

In this section, we fuse information from both branches of motion detection and target detection to identify moving targets using both motion and semantic information. Each of the two branches extracts three layers of feature maps with different convolutional depths.

First, we feed the third output F_3 of the motion detection into the nonlocal block [42] to further integrate the motion characteristics of the T channel. The nonlocal block is good for integrating information between channels, but it is expensive, so we only use it for the last motion feature. Then, the first two features of the motion branch and the motion information of the time dimension in the third motion feature of the nonlocal block fusion are fused into the channel. This study tested two different pooling methods to fuse motion information from the temporal dimension into the channels. Finally, we obtained three motion features with different convolutional depths, and the spatial dimensions of the three motion features corresponded to the three object features obtained by object detection. Since the corresponding features on the layer to be fused have the same spatial resolution, referring to the findings of C. Feichtenhofer [43], we tried to fuse the features using two methods to compute the sum of two feature maps at the same spatial location and stack features in the channel direction.

3. Dataset and Results

In this section, we first introduce the datasets, including multiple image sequences. Secondly, we present the evaluation metrics, evaluate the proposed method, and finally, compare it with several state-of-the-art MOD methods.

3.1. Dataset

Two datasets were used in this study, namely VIVID [44] and MDR105 [38]. The details of the datasets are given in Table 2.

Table 2. The datasets used for the experiment.

Dataset	Image Size	Number of Frames	Sequence Name
VIVID	640 × 480	10,019	EgTest01, EgTest02, EgTest03, EgTest04, EgTest05, Seq01
MDR105	640 × 360	13,931	Car01, Car03, Car04, Car05, Car06, Car07, Car08, Car10, Car12, Car13, Car15, Car16, Car17, Car19, Car21, Car22, Car23, Car26, Car27, Car28, Car30, Car33, Person00, Person02, Person07, Person08, Person13, Person11, Person22, Person30, Person31, Person33, Person34

(1) VIVID: The Defense Advanced Research Projects Agency Video Verification of Identity (VIVID) datasets, which are widely used in moving object detection algorithms, contain multiple video sequences containing a variety of moving vehicles captured by a moving platform. We selected six of these sequences and manually labeled each moving vehicle.

(2) MDR105: The MDR105 dataset contains 105 videos acquired by UAVs and surveillance cameras, divided into three categories. We selected 33 video sequences containing both human and vehicle categories acquired by UAVs as part of the experimental data.

The proposed method was compared with classical algorithms on the VIVID dataset and the state-of-the-art deep learning-based methods on the MDR105 dataset. We used the first 80% of all video sequences as the training set, 10% as the validation set, and the last 10% as the test set. Since our method requires two consecutive images to be input simultaneously, frames k and $k - 4$ of the same sequence were fed into the network as a group according to the dataset's frame rate and movement speed. At the same time, after pairing, the data were shuffled during training. In addition, we used random affine transformation, HSV enhancement, and random inversion in data augmentation and randomly paired the k frames themselves during the pairing process. At this point, its label was empty, since no motion occurred between the two identical images.

3.2. Evaluation Metrics

In this study, we evaluated the performance of moving object detection algorithms using precision (*Pr*), recall (*Re*), average precision (*AP*), and *F*1 score, which are defined as

$$\mathrm{Pr} = \frac{\text{number of true detections}}{\text{number of detected objects}} = \frac{TP}{TP + FP} \tag{9}$$

$$\mathrm{Re} = \frac{\text{number of true detections}}{\text{number of existing objects}} = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = 2 \times \frac{\mathrm{Pr} \times \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}} \tag{11}$$

where *FP* refers to incorrectly detected boxes, *TP* refers to true detections, and *FN* refers to ground truth boxes that were missed by the method. A ground truth bounding box was considered as *TP* if it intersected with 50% or more of the detected bounding boxes. *FP* refers to the fact that the detected bounding boxes do not overlap, even in 50% of the cases.

The average precision (*AP*) measures how well the model performs for a given category and is defined as the area under the Pr–Re curve, which was calculated using interpolation. The formula for calculating *AP* is as follows:

$$AP = \sum_{k=1}^{N} \max_{\widetilde{k} \geq k} P(\widetilde{k}) \Delta R(k) \tag{12}$$

The proposed method was tested and evaluated on a computer with an AMD Ryzen 5 3600X 6-Core Processor CPU, 16 GB of computer memory, and a GeForce RTX 3060Ti GPU with 8 GB memory, implemented using the open-source Pytorch framework. During training, the stochastic gradient descent [45] (SGD) was used to optimize the parameters. The weights were initialized with the Kaiming distribution [46]. The IoU threshold of nonmaximum suppression (NMS) was 0.5 for inference.

3.3. Ablation Experiments

In this section, we present ablation experiments that were conducted to verify the effectiveness of the proposed MFE module and MIE network and then compare our proposed MFE-MIE method with other moving object detection algorithms.

Table 3 shows the effectiveness of each part of the MFE-MIE algorithm. Table 3 is divided into the object detection and motion detection branches. The motion detection branch has two structures: the MFE module and the MIE network. A check mark (√) indicates that the corresponding module is enabled, and a fork (×) indicates that the corresponding module is not used.

Table 3. Effectiveness of each part of the MFE-MIE.

	Object	**Motion**		**Pr**	**Re**	**F1**	**AP@0.5**
		MFE	**MIE**				
MFE-MIE	×	√	√	90.64%	36.92%	0.52	64.04%
	√	×	×	90.65%	79.46%	0.85	90.34%
	√	×	√	90.02%	89.23%	0.90	93.78%
	√	√	√	**93.21%**	**92.72%**	**0.93**	**96.71%**

The best results are presented in bold.

Firstly, the proposed algorithm has a low recall rate (36.92%) when only the motion detection branch is available, indicating that by using only motion information without the moving target's semantic information the boundary information of the motion can be found, but the specific object information cannot be located. In the same way, with only the object detection branch, the algorithm can detect all objects in the image, but it cannot distinguish whether the object is moving or not, so it can obtain a higher recall rate (79.46%). Only by combining the object detection branch with the motion detection branch can we achieve good detection results and obtain the highest recall rate (92.72%) and the highest F1 score (0.93).

At the same time, the effectiveness of our proposed MFE module and MIE network was demonstrated through experiments, as presented in in Table 3. Rows 2 and 3 of Table 3 show a 10 percent improvement in recall when using the MIE network to extract motion information compared with using only the object detection branch. Rows 3 and 4 of Table 3 show that using the MFE module is crucial for the motion detection branch. This module integrates the motion information interframe well and plays a crucial role in providing better extraction of motion information for the subsequent MIE network. The improvements in Pr, Re, F1, and AP@0.5 were 3.19%, 3.49%, 0.03, and 2.93%, respectively.

We also tried to vary the position of the background-compensated convolutional layer in the MFE module to investigate the effect on motion target detection. The results are shown in Table 4. The first position (denoted by A in Table 4) uses the approach presented in Equation (7) to multiply the weight R_k with the F_{k-n} without background compensation to obtain the enhanced features. The second position (denoted by B in Table 4) multiplies R_k with the background-compensated F'_{k-n} to get the enhanced features. Meanwhile, the other settings of the network remain unchanged.

Table 4. The position of the background-compensated convolution layer in the MFE.

	Position	**Pr**	**Re**	**F1**	**AP@0.5**	**FPS**
MFE	A	93.21%	92.72%	0.93	96.71%	25.25
	B	92.84%	90.95%	0.92	96.63%	25.25

Table 4 shows that the convolutional layer for background compensation is better detected when placed at position A than at position B. This is because multiplying the compensated features F'_{k-n} using the weights R_k artificially weakens the motion information between the subsequent F_k and F'_{k-n}. This is not conducive to the further extraction of motion information by the subsequent MIE network. In addition, R_k is a relatively coarse background compensation weight that is available for feature enhancement. However, artificially compensating the motion background before the MIE network will cause it to lose some motion information. Our original intention in designing the MFE module was to make the network pay more attention to the motion information between frames.

Furthermore, as shown in Figure 4, we visualized the results of the frame difference method, image registered frame difference method, and image feature difference method to illustrate the advantages of using feature differences to enhance motion features in the proposed MIE model. The results of feature difference were obtained by visualizing six randomly selected channel features from D_k.

Figure 4a is from frame 15 and frame 20 in the rg01 sequence of the VIVID dataset. It shows five moving vehicles. The object moving distance between two frames is small, and the background motion is small. There is shadow interference. Figure 4b is from frames 281 and 286 of the Car08 sequence of the MDR105 dataset. It illustrates a black car parked on the side of the road and a white moving car. The background motion amplitude is large, and the vehicle is moving fast.

Figure 4 shows that the difference image obtained using the frame difference method has the phenomenon of ghosting, and the background of the difference image is cluttered. Due to the movement of the background, the contours of the detected moving objects are relatively large, and the shadows are also detected as targets. This is because the camera's motion causes the whole image to be in motion, and the moving background causes the frame difference method to fail. The registration algorithm can overcome the influence of the moving background to some extent, but there are still some issues, such as target holes and large contours. By using convolutional layers to compensate for background motion, the proposed MFE model makes the network focus on moving objects (white cars) and suppresses objects with similar semantic features but without motion (black cars). It makes object extraction more complete. In addition, the effect of the shadow of the moving target itself is eliminated.

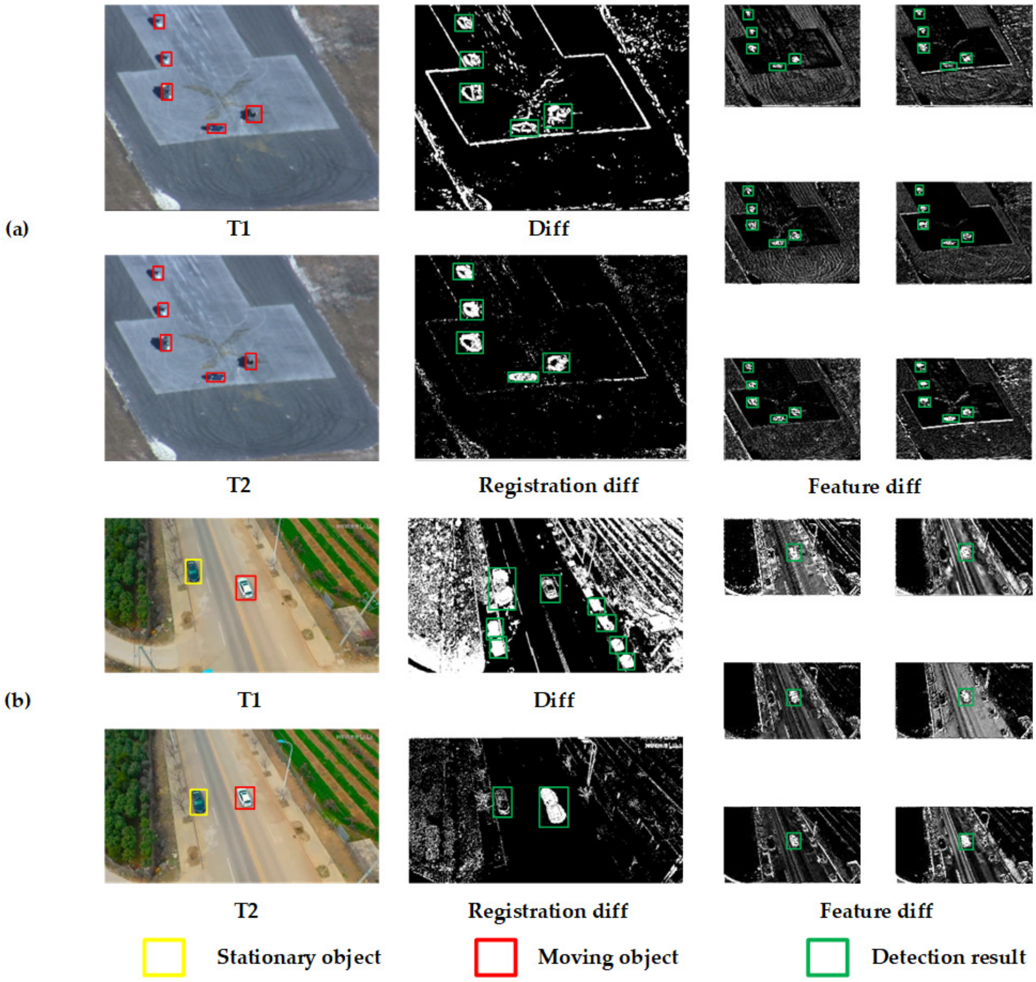

Figure 4. Visualization of the frame difference method (denoted by Diff in Figure 4), the image registration difference method (denoted by Registration diff in Figure 4), and the feature difference in the MFE module (denoted by Feature diff in Figure 4). (**a**) Frames 15 and 20 of the eg01 sequence in the VIVID dataset; (**b**) Frames 281 and 286 of the Car08 sequence in the MDR105 dataset.

In addition, we verified the effectiveness of decomposing 3D convolution into 1D temporal convolution and 2D spatial convolution for MOD tasks, as shown in Table 5. Compared with full 3D convolution, the improved 3D convolution improves the AP@0.5 by 2.61% and the FPS by 5.19 frames with a 2.67 M reduction in parameters and a 16.71 G reduction in computation. This is because the 3D convolutional decomposition introduces additional nonlinear corrections, while we did not destroy the relevance of its motion information in the temporal dimension. Three-dimensional convolution can simultaneously extract interframe temporal and intraframe semantic information, while the improved 3D convolution focuses more on extracting motion information. Therefore, the proposed model enhances the representativeness of the extracted motion information while reducing the computational cost.

To be consistent with the temporal dimension of the object detection branch, we needed to fuse the temporal dimension information of the output motion features of the MIE network with the channel dimension. Therefore, we also evaluated the influences of different pooling methods on the temporal dimension of the fused motion features on the detection results. The results in Table 6 show that, compared with average pooling, using the maximum pooling method can better retain motion information and is more suitable for moving object detection, with increases of 6% in recall and 0.04 in F1 score. This is because the motion information mainly contains the edge change information of the moving object. At the same time, the average pooling method will weaken the motion information by averaging the motion information in the temporal dimension with other irrelevant information in the neighborhood. However, the maximum pooling method will retain abrupt edge changes and suppress the noise, thus achieving a better performance than the average pooling method.

Table 5. Improved 3D convolution and full 3D convolution.

MFE-MIE	Params(M)	FLOPs(G)	Pr	Re	F1	AP@0.5	FPS
Full 3D	27.39	51.39	92.31%	88.98%	0.91	94.10%	20.06
Improve 3D	24.72	34.68	93.21%	92.72%	0.93	96.71%	25.25

Table 6. Pooling approach for the temporal dimension of the motion features.

Pooling Type	Pr	Re	F1	AP@0.5
Avg Pooling	91.76%	86.94%	0.89	94.70%
Max Pooling	93.21%	92.72%	0.93	96.71%

Based on the conclusion of [43], we tested two methods of fusing motion and semantic information: element-wise summation (denoted by Sum in Table 7) and concatenation (denoted by Cat in Table 7) along the channels. The results are shown in Table 7, where the first number under Channel represents the third feature channel number of the object detection branch, and the second number represents the third feature channel number of the motion detection branch.

Table 7. Types of fusion of motion and semantic features.

Fusion Type	Channel		Pr	Re	F1	AP@0.5
	Object	Motion				
Sum	512	512	**96.12%**	73.68%	0.83	89.98%
	1021	1024	93.48%	**79.86%**	**0.86**	**91.20%**
Cat	512	512	**97.17%**	70.79%	0.82	91.03%
	512	1024	91.10%	90.68%	0.91	94.81%
	768	1024	93.21%	92.72%	**0.93**	**96.71%**
	1024	1024	91.59%	**94.36%**	**0.93**	**96.71%**

The best results are presented in bold.

The results show that the concatenation method has a better fusion effect than the element-wise addition method. Moreover, it can be found from Table 7 that when the number of channels of the two branches increases, the fusion effect improves. For example, in the case of the concatenation method, when the numbers of channels are 512 and 1024, the recall rate is increased by 23.57%, and F1 is increased by 0.11. This is because more channels retain more information, which also leads to a large computational cost. When the number of object detection branch channels is 768, and the number of motion detection branch channels is 1024, the calculation amount and fusion effect reach a balance. The detection effect at this time is equivalent to that when the number of the two branch channels is 1024. This situation occurs because, in the object detection branch, we can achieve good results by using the CSPDarknet53 backbone to extract the semantic information from the target. However, the information extraction ability of the motion detection branch needs to be improved. Therefore, to balance the computational burden and the detection effect, it is wise to extend the number of channels of the motion detection branch to retain more motion information.

3.4. Comparison with State-of-the-Art Methods

In this section we compare our proposed method with several state-of-the-art methods on two datasets, VIVID and MDR105. According to the results presented in Section 3.2, our method uses the maximum pooling approach to fuse motion features in the temporal dimension and the concatenation method to combine the motion and object detection branches.

3.4.1. Experiments on the VIVID Dataset

We present the results of several other classical algorithms on the VIVID dataset to evaluate our proposed algorithm objectively. The first algorithm (denoted as Diff in Table 7) is the frame difference method, which is widely used due to its high computational efficiency in the case of a stationary camera. It subtracts the current frame image directly from the previous one to obtain the difference image between the two frames and considers that the moving object causes the main difference. The second method (denoted as Registration diff in Table 8) detects moving targets by registering the acquired image sequences and then using the frame difference method, which is modified from the frame difference method. It uses the registration strategy to compensate for the background of interframe motion. The third method proposed in [47] increases the image quality evaluation link, further improving the detection performance using the edge detection algorithm and obtaining a more accurate motion target. The fourth method is proposed in [48]. It first assumes the generation of the multiscale target, uses the kernel density estimation to generate the target likelihood map, and finally, adopts an adaptive threshold algorithm to extract the motion target. In the fifth method, MCD [49], selected points from consecutive frames are tracked using the Lucas–Kanade method. The RANSAC [50] method is used to calculate the homograph matrix representing the camera motion. Then, the background model is compensated for in the current frame before updating the background model and applying background subtraction. The method proposed in paper [51] uses a region adjacency graph to represent image frames. Then, multiple frames of images are used for matching, and finally, a coloring algorithm is used to assign foreground and background labels to each region. The last approach is proposed in [52], in which bLPS-HOG features are used to train a linear SVM classifier to detect moving vehicles in airborne videos.

Table 8. Comparison of detection results on the VIVID dataset.

Method	Pr	Re	F1	FPS
Diff [40]	0.24	0.83	0.37	**32.5**
Registration diff [53]	0.77	0.85	0.81	14.8
Tian, Y. [47]	0.87	0.82	0.85	1.90
Öz, S. [48]	0.90	0.82	0.86	/
Cao, X. [52]	0.88	0.88	0.88	20.8
MCD [49]	0.90	0.88	0.89	9.28
Kalantar, B. [51]	**0.94**	0.89	0.91	1.60
MFE-MIE (our)	0.93	**0.93**	**0.93**	25.25

The best results are presented in bold.

Table 8 shows the detection performances of the methods being compared. The experimental results prove that the proposed method achieves the best performance compared with other methods. Among them, the frame difference method achieves the fastest detection rate. However, the moving background will cause many false alarms in different images, leading to the failure of the frame difference method. In addition, the methods based on image registration heavily rely on the accuracy of the registration algorithm, and the computation time of the registration algorithm is often too long to meet the real-time requirements. The methods proposed in [49,52] have a better detection effect than the previous ones but still cannot meet the real-time requirement. The method proposed in [51] adopts a new region-matching method with higher accuracy and recall. However, its time overhead is greater, and the frame rate is only 1.6 fps, making it insufficient for practical tasks. Table 8 shows that our method outperforms the classical methods in terms of the recall and F1 score. In addition, our method achieves real-time detection (25.25 fps). This is because the proposed method can obtain better motion information between images, and the combination of robust object detection information endows the network with a stronger moving object detection ability.

We also give the detection results for the frame difference method, the registration frame difference method, the method presented in [47], and MCD on the VIVID dataset in Figure 5, because they are open-source. The first column in Figure 5 shows the ground truth. The second column shows the detection results of our method, the third column shows the detection results of the frame difference method, the fourth column shows the detection results of the registration frame difference method, and the fifth column shows the detection results of the method presented in [47].

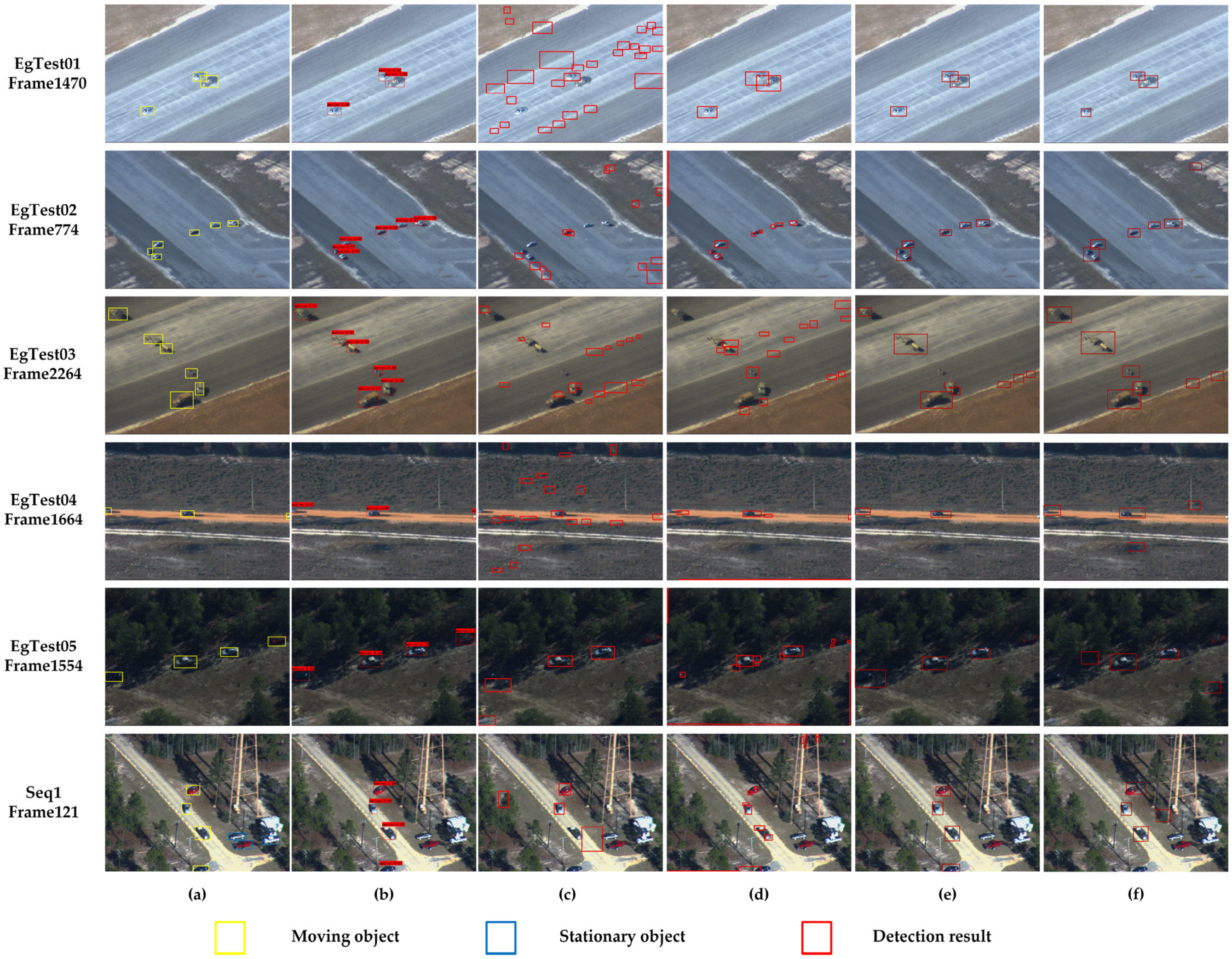

Figure 5. Comparison of the detection results on the VIVID dataset. (**a**) Ground truth, (**b**) Proposed method, (**c**) Diff, (**d**) Registration diff, (**e**) Method presented in [47], (**f**) MCD.

As shown in Figure 5, our approach adapts well to various scenes and shows the best performance. The frame difference method has the worst performance due to camera motion and hardly works in five scenes. The collinear frame difference method has a better detection performance, but it is affected by the principle of the method and shows some inaccuracies, such as the inability to eliminate the ghost, resulting in inaccurate detection frames (row 1, column 4 in Figure 5); the appearance of voids, resulting in one object being detected as two objects (row 2, column 4 in Figure 5); and the severe influence on object shadows (row 3, column 4 in Figure 5). With the addition of the edge detection algorithm, the method in [47] further improves the detection performance compared with the isotropic frame difference method but still cannot reject object shadows. Additionally, adding the edge detection algorithm brings the problem of rejecting objects that look similar to the environment (row 4, column 5 in Figure 5).

3.4.2. Experiments on the MDR105 Dataset

We also compared several advanced CNN-based algorithms on the MDR105 dataset. Among them, Deep-Sort [54] was trained using the same settings as in the paper [38]. YOLOv5 series and YOLOv7 were fine-tuned on the dataset using their published weights. The results are shown in Table 8. ECO [55] is an excellent single-target tracking algorithm, but the presence of multiple moving targets in the dataset at the same time leads to poor performance. Deep-Sort is a multitarget tracking algorithm that has achieved excellent results in multitarget tracking tasks. The algorithm first extracts all targets from each image frame by the target detection network and then predicts the motion trajectory of each target by Kalman filtering and uses a weighted Hungarian algorithm for matching. It can detect and track all targets in the image simultaneously but cannot distinguish whether the targets are moving or not. YOLOv5 is a target detection algorithm that can quickly detect all objects in an image but cannot distinguish whether they are moving or stationary. The method proposed in [38] (named MDR in Table 9) is consistent with the idea of our method, which uses the classical alignment algorithm combined with the frame difference method as the motion detection branch and uses the simplified YOLOv3 algorithm as the target detection branch. It integrates the detection results of the two branches to obtain the final motion target. It combines the classical frame difference method and the CNN-based target detection method to achieve a good detection effect. However, it fails to achieve end-to-end detection and meet the real-time detection requirements.

Table 9. Comparison of the detection results on the MDR105 dataset.

METHOD	Pr	Re	F1
ECO	0.86	0.73	0.79
Deep-sort	0.82	0.87	0.85
YOLOv5s	0.86	0.74	0.80
YOLOv5m	0.84	0.82	0.83
YOLOv5l	0.89	0.76	0.83
YOLOv7	0.80	0.83	0.82
MDR	**0.96**	0.85	0.90
MFE-MIE (our)	0.92	**0.91**	**0.92**

The best results are presented in bold.

As Table 9 shows, ECO is a single-target tracking algorithm, and its inability to track multiple targets simultaneously results in it having the worst performance. YOLOv5 series, YOLOv7, and Deep-sort can only detect the objects in the image and cannot determine whether they are moving, so their recall rates are relatively low. In addition, from the experimental results of the YOLOv5 series and YOLOv7, we can see that increasing the target detection capability of the target detection algorithm cannot achieve better motion target detection results, which is consistent with our intuition. MDR has the advantage of combining motion and object detection, and its detection effect is second only to our method. However, its motion detection branch has a low ability to detect motion, resulting in a low recall rate. Our method achieves a state-of-the-art performance, achieving the highest F1 scores while balancing the accuracy and recall rates.

Figure 6 presents a visualization of the partial detection results of the above five algorithms on the MDR105 dataset. The selected sequences are very representative. Among them, the Car05 sequence has only one moving target, but the target is similar to the background. In addition, the target is tiny. There are three vehicles in the Car08 sequence; two of them are moving. One of the vehicles has a relatively obvious appearance, and the other has a tiny target, which makes detection more difficult. In the sequence of Car21, there are three moving vehicles on the road and many stationary vehicles on the side of the road. They share the same semantic features as moving vehicles. The Car22 sequence contains one moving vehicle and one stationary vehicle. In the Person34 sequence, only one person is moving, while the vehicles on the road are all stationary.

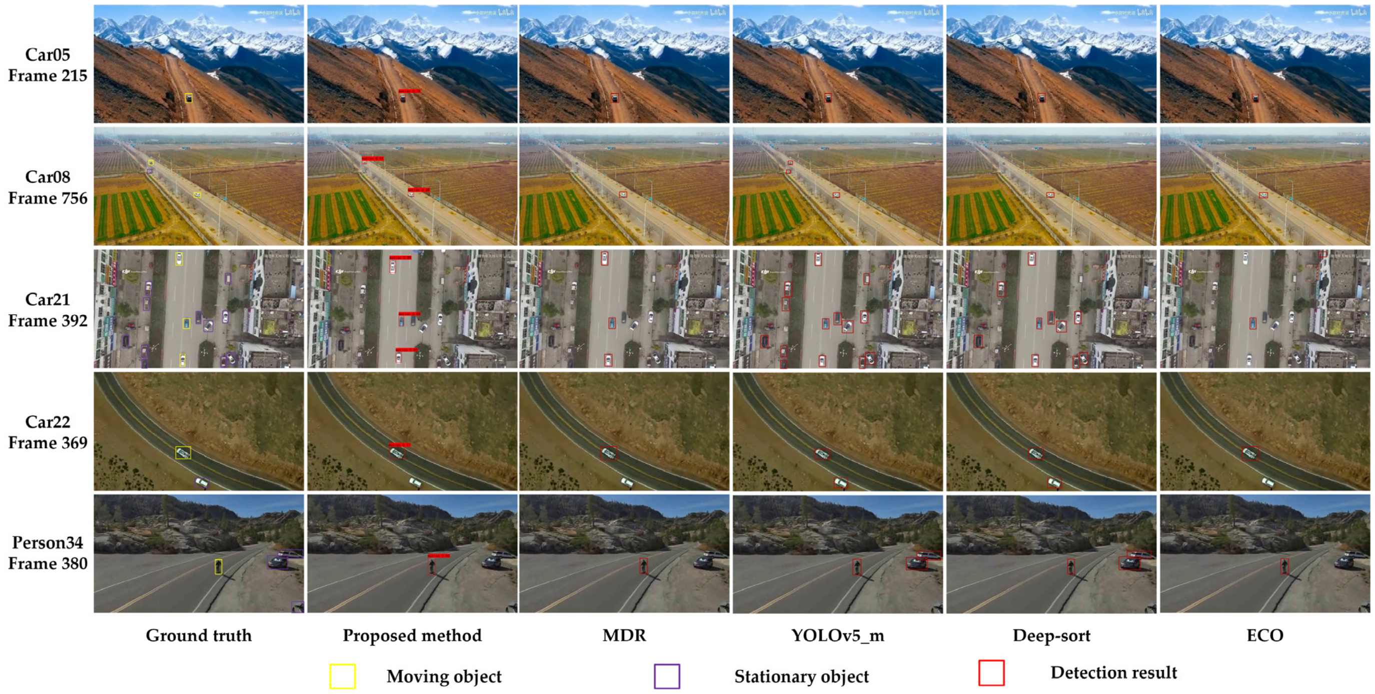

Figure 6. Comparison of the detection results on the MDR105 dataset.

In Figure 6, column 1 shows the ground truth. Column 2 shows the detection results of our proposed method, column 3 shows the detection results of MDR, column 4 shows the detection results of YOLOv5, column 5 shows the detection results of Deep-sort, and column 6 shows the detection results of ECO. YOLOv5 and Deep-sort can detect all objects in the figure. However, they cannot distinguish whether the target is moving (Figure 6, row 2, column 4, column 5). MDR combines classical motion detection algorithms with CNN-based target detection algorithms to obtain good results, but its weak motion detection capability causes a missed detection (Figure 6, row 2, column 4). Compared with the other four algorithms, our proposed method achieves the best moving target detection effect. The reason for this is that our MFE module and the MIE network can extract the motion information between frames well and then fuse the motion information at three levels with the semantic information extracted from the target detection branch, which is suitable for the detection of tiny targets and objects with small displacements.

Finally, the partial detection results of the proposed method on the VIVID datasets and MDR105 datasets are shown in Figure 7.

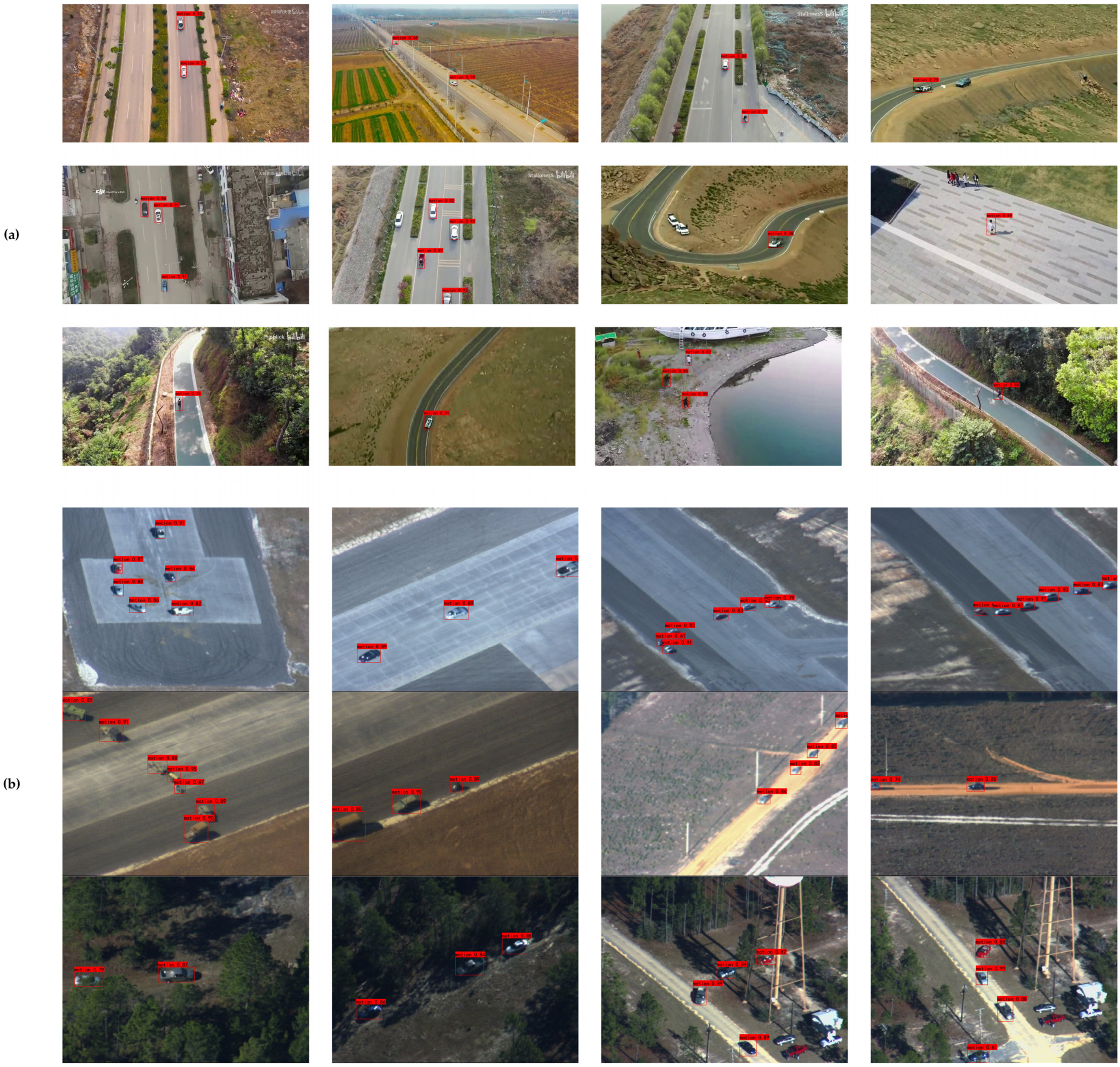

Figure 7. Results of the proposed method on the different datasets. (**a**) Detection results on the MDR105 dataset; (**b**) Detection results on the VIVID dataset.

4. Discussion

In this work, inspired by the excellent performance of two-stream networks combining spatial and temporal information for motion recognition, we proposed an end-to-end moving object detection model for remote sensing image sequences that operates in real-time. The experimental results show that our proposed MFE-MIE model achieves a real-time (25.25 fps) detection speed with the highest F1 scores (0.93). In addition, we performed a quantitative evaluation by ablation experiments to demonstrate that the proposed MFE module enhances motion feature extraction. We also proposed an MIE network based on improved 3D convolution and experimentally demonstrated that it improves accuracy and reduces false alarm rates.

In Section 3, we analyzed which method is more suitable for motion information retention when fusing the temporal dimensional information of motion features. The experiments show that the maximum pooling method provides more retention than the average pooling method, and the accuracy, recall, and F1 scores increased by 1.45%, 5.78%, and 0.04, respectively.

Additionally, we used the convolutional layer for background compensation at different positions and determined the influence on the detection results. The experimental results show that using the weight R_k to enhance the features without background compensation can achieve better detection results. This is because multiplying the compensated features F'_{k-n} using the weights R_k artificially weakens the motion information between the subsequent F_k and F'_{k-n}. This is not conducive to the subsequent further extraction of motion information by the MIE network. During the process of motion feature enhancement, the original features should be preserved and enhanced as much as possible instead of artificially reducing the interframe information, which is harmful to the subsequent motion information extraction.

At the same time, we studied the impacts of different ways of fusing motion and semantic information on the moving object detection task. The experimental results show that the accuracy rate of concatenation along the channel was 1.89% lower than the element-wise summation for the same feature dimension. Nonetheless, this precision sacrifice is meaningful, with 0.07 and 14.5% gains in terms of the F1 and recall, respectively. Element-wise summation is more flexible than concatenation along channels. The element-wise summation method requires motion features and semantic features to have the same spatial dimension and number of channels. However, the concatenation along channels method only requires them to have the same spatial dimension. When we set the number of semantic feature channels to 768 and the number of motion feature channels to 1024, we were able to achieve the same F1 (0.93) score as when the number of the two branch channels was 1024. In addition, this setup reduced the overall detection time.

The experimental results on the VIVID dataset show that the proposed method can better localize moving objects and achieve real-time detection compared with the classical moving object detection method. In contrast, the detection effect of the classical algorithm is affected by the registration method and cannot achieve real-time detection. In addition, the detection effect depends on postprocessing. The proposed algorithm can obtain the best detection effect in different image sequences. The idea of combining motion information with semantic information was proven to be effective for moving object detection tasks.

The detection results on the MDR105 dataset show that our method focuses on the moving object detection task and achieves the best detection results compared with other deep-learning-based methods. Unlike the algorithms presented in [38], our method is all implemented by CNN. The proposed method only needs two RGB images as the input and no other data are required as additional input to output the detection result in an end-to-end manner. Both object detection algorithms and multiobject tracking algorithms can detect all objects in the image. However, they cannot tell whether the object is in motion.

The detection results for the VIVID and MDR105 moving object datasets show that, in most cases, the proposed method can accurately locate the positions of the moving objects, even though the target within the VIVID is smaller. Objectively speaking, there are some misdetections and missed detections in our method. When the displacement of the target is small, the motion detection branch fails to detect the motion information, so the moving target is treated as stationary by our method, resulting in a missed detection. This is due to the lack of motion detection capability. Meanwhile, our method of fusing motion information with semantic information is relatively simple.

5. Conclusions

Detecting moving objects in remote sensing image sequences is a challenging problem due to the small sizes of moving objects, the small amplitudes of motion, and the complex image background. In this work, we proposed a moving object detection model that provides a feasible solution for moving object detection in remote sensing image sequences. We use a dedicated branch to extract motion information between frames and design a motion feature enhancement module before computing motion information. The extracted motion information is combined with the powerful target detection capability of deep learning to achieve an end-to-end solution to the motion target detection task. The experimental results prove that the proposed method performs well, allowing the rapid detection of moving objects. At the same time, it was proven that combining semantic information based on a single-frame image and motion information based on multiple frames is helpful for moving object detection.

Our future work will focus on two main aspects. First, we will further strengthen the extraction of motion features and study how to make the network better retain interframe motion information without adding additional information. For example, we might try to approximate the optical flow representation using features from two images. Second, we will explore an approach that can better combine object and motion detection branches to make them more suitable for the moving object detection task. For instance, feeding two frames of images directly into a network branch allows the network to learn how to fuse motion information with semantic information.

Author Contributions: Conceptualization, B.W. and S.Z.; methodology, B.W.; validation, B.W. and S.Z.; investigation, B.W., F.X. and J.L.; resources, J.L. and C.L.; writing—original draft preparation, B.W.; writing—review and editing, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61905240.

Data Availability Statement: Not applicable.

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

References

1. Wu, S.; Oreifej, O.; Shah, M. Action Recognition in Videos Acquired by a Moving Camera Using Motion Decomposition of Lagrangian Particle Trajectories. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 1419–1426.
2. Zhang, B.W.; Wang, L.M.; Wang, Z.; Qiao, Y.; Wang, H.L. Real-Time Action Recognition with Deeply Transferred Motion Vector CNNs. *IEEE Trans. Image Process.* **2018**, *27*, 2326–2339. [CrossRef] [PubMed]
3. Su, Y.; Liu, J.; Xu, F.; Zhang, X.; Zuo, Y. A Novel Anti-Drift Visual Object Tracking Algorithm Based on Sparse Response and Adaptive Spatial-Temporal Context-Aware. *Remote. Sens.* **2021**, *13*, 4672. [CrossRef]
4. Ma, C.; Yang, X.; Chongyang, Z.; Yang, M.H. Long-Term Correlation Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5388–5396.
5. Cao, Y.; Wang, G.; Yan, D.; Zhao, Z. Two Algorithms for the Detection and Tracking of Moving Vehicle Targets in Aerial Infrared Image Sequences. *Remote Sens.* **2015**, *8*, 28. [CrossRef]
6. Kim, B.; Neville, C. Accuracy and feasibility of a novel fine hand motor skill assessment using computer vision object tracking. *Sci. Rep.* **2023**, *13*, 1813. [CrossRef] [PubMed]
7. Li, W.; Ma, Q.; Liu, C.; Zhang, Y.; Wu, X.; Wang, J.; Gao, S.; Qiu, T.; Liu, T.; Xiao, Q.; et al. Intelligent Metasurface System for Automatic Tracking of Moving Targets and Wireless Communications Based on Computer Vision. *Nat. Commun.* **2023**, *14*, 989. [CrossRef]
8. Moeslund, T.B.; Granum, E. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.* **2001**, *81*, 231–268. [CrossRef]
9. Yi, S.; Li, H.; Wang, X. Pedestrian Behavior Modeling from Stationary Crowds with Applications to Intelligent Surveillance. *IEEE Trans. Image Process.* **2016**, *25*, 4354–4368. [CrossRef]
10. Rahmaniar, W.; Wang, W.-J.; Chen, H.-C. Real-Time Detection and Recognition of Multiple Moving Objects for Aerial Surveillance. *Electronics* **2019**, *8*, 1373. [CrossRef]
11. Yazdi, M.; Bouwmans, T. New trends on moving object detection in video images captured by a moving camera: A survey. *Comput. Sci. Rev.* **2018**, *28*, 157–177. [CrossRef]
12. Roy, S.D.; Bhowmik, M.K. A Comprehensive Survey on Computer Vision Based Approaches for Moving Object Detection. In Proceedings of the IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020; pp. 1531–1534.
13. Yu, Y.; Kurnianggoro, L.; Jo, K.-H. Moving Object Detection for a Moving Camera Based on Global Motion Compensation and Adaptive Background Model. *Int. J. Control Autom. Syst.* **2019**, *17*, 1866–1874. [CrossRef]
14. Zhao, X.; Wang, G.; He, Z.; Jiang, H. A survey of moving object detection methods: A practical perspective. *Neurocomputing* **2022**, *503*, 28–48. [CrossRef]

15. Chapel, M.-N.; Bouwmans, T. Moving objects detection with a moving camera: A comprehensive review. *Comput. Sci. Rev.* **2020**, *38*, 100310. [CrossRef]
16. Collins, R.; Lipton, A.; Kanade, T.; Fujiyoshi, H.; Duggins, D.; Tsin, Y.; Tolliver, D.; Enomoto, N.; Hasegawa, O.; Burt, P. A System for Video Surveillance and Monitoring. *Robot. Inst.* **2000**, *5*, 1–68.
17. Haritaoglu, I.; Harwood, D.; Davis, L.S. W/sup 4/: Real-time surveillance of people and their activities. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 809–830. [CrossRef]
18. Maier, J.; Humenberger, M. Movement Detection Based on Dense Optical Flow for Unmanned Aerial Vehicles. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 146. [CrossRef]
19. Minaeian, S.; Liu, J.; Son, Y.-J. Effective and Efficient Detection of Moving Targets from a UAV's Camera. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 497–506. [CrossRef]
20. Wu, Y.; He, X.; Nguyen, T.Q. Moving Object Detection with a Freely Moving Camera via Background Motion Subtraction. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 236–248. [CrossRef]
21. Elgammal, A.; Duraiswami, R.; Davis, L.S. Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1499–1504. [CrossRef]
22. Stauffer, C.; Grimson, W.E.L. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 747–757. [CrossRef]
23. Zuo, C.; Qian, J.; Feng, S.; Yin, W.; Li, Y.; Fan, P.; Han, J.; Qian, K.; Chen, Q. Deep learning in optical metrology: A review. *Light Sci. Appl.* **2022**, *11*, 39. [CrossRef]
24. Li, J.; Jiang, S.; Song, L.; Peng, P.; Mu, F.; Li, H.; Jiang, P.; Xu, T. Automated optical inspection of FAST's reflector surface using drones and computer vision. *Light Sci. Appl.* **2023**, *4*, 1–11. [CrossRef]
25. Huang, L.; Luo, R.; Liu, X.; Hao, X. Spectral imaging with deep learning. *Light Sci. Appl.* **2022**, *11*, 61. [CrossRef] [PubMed]
26. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
28. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
29. 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.
30. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:abs/1804.02767.
31. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
32. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 779–788.
33. 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:abs/2207.02696.
34. 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:abs/2209.02976.
35. Lateef, F.; Kas, M.; Ruichek, Y. Temporal Semantics Auto-Encoding based Moving Objects Detection in Urban Driving Scenario. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 1352–1358.
36. Xiao, C.; Yin, Q.; Ying, X.; Li, R.; Wu, S.; Li, M.; Liu, L.; An, W.; Chen, Z. DSFNet: Dynamic and Static Fusion Network for Moving Object Detection in Satellite Videos. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
37. Zhu, H.; Yan, X.; Tang, H.; Chang, Y.; Li, B.; Yuan, X. Moving Object Detection with Deep CNNs. *IEEE Access* **2020**, *8*, 29729–29741. [CrossRef]
38. Zhu, J.; Wang, Z.; Wang, S.; Chen, S. Moving Object Detection Based on Background Compensation and Deep Learning. *Symmetry* **2020**, *12*, 1965. [CrossRef]
39. Li, D.; Mo, B.; Zhou, J. Boost Infrared Moving Aircraft Detection Performance by Using Fast Homography Estimation and Dual Input Object Detection Network. *Infrared Phys. Technol.* **2022**, *123*, 104182. [CrossRef]
40. Jain, R.; Nagel, H.H. On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *1*, 206–214. [CrossRef] [PubMed]
41. Feichtenhofer, C.; Fan, H.Q.; Malik, J.; He, K.M. SlowFast Networks for Video Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6201–6210.
42. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
43. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 1933–1941.
44. Collins, R.; Zhou, X.; Teh, S. An Open Source Tracking Testbed and Evaluation Web Site. In Proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Clearwater, FL, USA, 15–17 January 2005; pp. 1–8.

45. Song, S.; Chaudhuri, K.; Sarwate, A.D. Stochastic gradient descent with differentially private updates. In Proceedings of the IEEE Global Conference on Signal and Information Processing, Austin, TX, USA, 3–5 December 2013; pp. 245–248.
46. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J.; IEEE. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 1026–1034.
47. Tian, Y.; Peng, C.; Wang, D.; Wan, B. High confidence detection for moving target in aerial video. *IET Image Process.* **2019**, *13*, 2724–2734. [CrossRef]
48. Alkanat, T.; Tunali, E.; Öz, S. A Real-time, Automatic Target Detection and Tracking Method for Variable Number of Targets in Airborne Imagery. In Proceedings of the 10th International Conference on Computer Vision Theory and Applications (VISAPP), Berlin, Germany, 11–14 March 2015; pp. 61–69.
49. Yi, K.M.; Yun, K.; Kim, S.W.; Chang, H.J.; Choi, J.Y. Detection of Moving Objects with Non-stationary Cameras in 5.8ms: Bringing Motion Detection to Your Mobile Device. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 27–34.
50. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
51. Kalantar, B.; Mansor, S.B.; Abdul Halin, A.; Shafri, H.Z.M.; Zand, M. Multiple Moving Object Detection from UAV Videos Using Trajectories of Matched Regional Adjacency Graphs. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5198–5213. [CrossRef]
52. Cao, X.; Wu, C.; Lan, J.; Yan, P.; Li, X. Vehicle Detection and Motion Analysis in Low-Altitude Airborne Video Under Urban Environment. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1522–1533. [CrossRef]
53. Shastry, A.C.; Schowengerdt, R.A. Airborne Video Registration and Traffic-Flow Parameter Estimation. *IEEE Trans. Intell. Transp. Syst.* **2005**, *6*, 391–405. [CrossRef]
54. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
55. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.

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.