remote sensing

MDPI

Article

An Integrated Detection and Multi-Object Tracking Pipeline for Satellite Video Analysis of Maritime and Aerial Objects

Zhijuan Su [1], Gang Wan [1], Wenhua Zhang [2], Ningbo Guo [1], Yitian Wu [1], Jia Liu [1], Dianwei Cong [1,*], Yutong Jia [1] and Zhanji Wei [1]

[1] School of Space Information, Space Engineering University, Beijing 101407, China; hgfgh@stu.cuz.edu.cn (Z.S.); dsddfff@stu.cuz.edu.cn (G.W.); sxguonb@163.com (N.G.); ytwu@whu.edu.cn (Y.W.); xssf@stu.cuz.edu.cn (J.L.); jiayutong@hdg.edu.cn (Y.J.); weizhanji@163.com (Z.W.)
[2] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; whzhang@njust.edu.cn
* Correspondence: congdianwei@sina.com

Abstract: Optical remote sensing videos, as a new source of remote sensing data that has emerged in recent years, have significant potential in remote sensing applications, especially national defense. In this paper, a tracking pipeline named TDNet (tracking while detecting based on a neural network) is proposed for optical remote sensing videos based on a correlation filter and deep neural networks. The pipeline is used to simultaneously track ships and planes in videos. There are many target tracking methods for general video data, but they suffer some difficulties in remote sensing videos with low resolution and those influenced by weather conditions. The tracked targets are usually misty. Therefore, in TDNet, we propose a new multi-target tracking method called MT-KCF and a detecting-assisted tracking (i.e., DAT) module to improve tracking accuracy and precision. Meanwhile, we also design a new target recognition (i.e., NTR) module to recognise newly emerged targets. In order to verify the performance of TDNet, we compare our method with several state-of-the-art tracking methods on optical video remote sensing data sets acquired from the Jilin No. 1 satellite. The experimental results demonstrate the effectiveness and the state-of-the-art performance of the proposed method. The proposed method can achieve more than 90% performance in terms of precision for single-target tracking tasks and more than 85% performance in terms of MOTA for multi-object tracking tasks.

Keywords: optical remote sensing videos; correlation filter; deep neural network; target tracking

check for updates

Citation: Su, Z.; Wan, G.; Zhang, W.; Guo, N.; Wu, Y.; Liu, J.; Cong, D.; Jia, Y.; Wei, Z. An Integrated Detection and Multi-Object Tracking Pipeline for Satellite Video Analysis of Maritime and Aerial Objects. *Remote Sens.* **2024**, *16*, 724. https://doi.org/10.3390/rs16040724

Academic Editor: Zhenming Peng

Received: 9 January 2024
Revised: 1 February 2024
Accepted: 5 February 2024
Published: 19 February 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, since the successful launch of the Jilin No. 1 optical remote sensing video satellite, the interpretation of optical remote sensing videos has become a new and important topic in the field of remote sensing. Remote sensing videos capture dynamic changes in ground targets, which provide much more sufficient information than remote sensing images. With this new type of remote sensing data, many potential applications can be conducted, including surveillance of ground facilities, the dynamic monitoring of natural disasters or environments, and especially military security [1]. Target tracking plays an important role in video analysis and surveillance [2]. It is of great significance to track and generate moving trajectories of multiple targets in remote sensing videos [3]. For example, the behaviours and intentions of targets can be analysed and predicted based on their moving trajectories; the targets can be clustered based on the moving relationship between them; and moreover, target tracking can be integrated into satellites via processors with low power and high performance for real-time monitoring of important targets. Therefore, in this paper, we intend to develop a tracking system for remote sensing videos that detects and tracks all specified targets. This has been studied by very few scholars [4] due to the absence of data.

There are quite a few studies about detecting and tracking small targets in wide-field-of-view aerial videos with similar scenes to that of remote sensing videos. For example, in [5], the target behaviour model is explored based on road structures to generate constraints for regulating matching schemes. It efficiently detects and tracks moving vehicles in low-frame-rate aerial videos. Two trackers were designed in [6], with one based on background subtraction for generating an initialised moving target and another based on target state regression for frame-to-frame tracking. In [7], a two-stage spatial–temporal convolutional neural network (CNN) was proposed to detect small objects in large scenes. However, in those methods, the targets are too small to recognise, and therefore, temporal information is important to detect the targets, which results in the limitation that only moving targets can be detected and tracked.

In this paper, we aim to track specific targets in remote sensing videos, and thus, the targets should be first detected and then tracked. Therefore, a tracking system is designed using two techniques: object detection and target tracking. Object detection methods have been well studied [8], including methods for remote sensing images [9], which are used in this paper. Meanwhile, target tracking methods for general videos have also been adequately researched [10] and are currently being explored for use in remote sensing videos. Most target tracking approaches are based on correlation filtering, with the features extracted by manual extractors [11] or deep neural networks [12].

The pioneering minimum output sum of squared error filter (MOSSE) [13], the first correlation-filter-based target tracking method, tracks a target by defining/modifying a template that maximises the output response by convolving the template with the regions of interest (RoIs) of the tracked target. It employs gray-scale features and has the fastest tracking speed but relatively a low tracking accuracy. The peak side-lobe ratio (PSR) method was also proposed in MOSSE, which could determine whether the object is obscured or tracked unsuccessfully. Aiming to solve the problem of sample redundancy caused by sparse sampling among the traditional methods [14], the circulant structure of tracking by detection with kernels (CSK) [15] introduces a cyclic sampling strategy and kernel functions to improve the performance of the tracking method. Since then, the circulant matrix and kernel functions have found success in the correlation filter-based tracking field. CSK also uses gray-scale features and has a relatively slow tracking speed but with greatly improved accuracy compared with MOSSE on the same benchmark. The kernel correlation filter (KCF) [16] is the perfect version of CSK and has two major breakthroughs. First, the KCF expands the input from gray-scale single-channel features to multi-channel features (which can be colour or HOG). Second, it defines a connection method for multi-channel features. In addition, the importance of accuracy or tracking speed can be optionally determined using different kernels, for example, a Gaussian kernel used in the KCF has the highest accuracy and a linear kernel used in the KCF has the fastest tracking speed.

In 2006, a breakthrough in deep learning was made by Hinton and Salakhutdinov [17]. Since then, deep neural networks with hierarchical layers have shown their stronger feature representation power in a wide range of computer vision applications, especially convolutional neural networks (CNNs). The applications include classification, object detection, and target tracking in not only nature image data but also remote sensing data [18]. With the powerful feature representation and abstraction of CNN for images, many advantages can be achieved by incorporating correlation filtering into a CNN. On the one hand, the learned features are data-driven instead of handcrafted and can better represent the input data. On the other hand, the features are learned to fully accomplish the task and can better adapt to the problem. Therefore, many correlation filtering methods are proposed for incorporation with deep neural networks for target tracking.

Compared to KCF, the continuous convolution operators for visual tracking (C-COT) [19] method uses VGG [20], which is a CNN prototype designed to extract features. In addition, the prototype uses cubic interpolation to interpolate feature maps with different resolutions into a continuous spatial domain. Then, the Hessian matrix is used to obtain the sub-pixel accuracy target position, as shown in [21,22]. After the interpolation equation

is determined, the continuous spatial domain training problem is also solved. In order to solve certain major problems in C-COT (i.e., slow tracking speed, overfitting, and model drift), the efficient convolution operators for tracking (ECO) [12] were proposed with the following three improvements: (1) The convolution operation was decomposed and thus, the model parameters were reduced. (2) The generative sample space model was proposed, which can simplify the generation of training data sets and ensure the diversity of samples. (3) A new model update strategy was adopted that could avoid model drift.

However, in the aforementioned target tracking methods, the position of the target in the first frame can only be given manually, and only a single object of interest can be tracked (which cannot meet the requirement of tracking in remote sensing videos). First of all, the targets should be recognised instead of manually assigned in the first frame. The aim in solving this problem is to track the ships and planes in the scene. The ships and planes should be recognised automatically. Secondly, in addition to tracking specific moving targets, it is also particularly practical to monitor the stationary targets in the videos. Again, it may make more sense to track multiple targets. In the end, it is also important to recognise and track targets that suddenly appear in the scene. Moreover, compared with nature videos, remote sensing videos are usually of low resolution and can easily suffer in quality due to weather conditions. Therefore, the targets in them are often blurry, which greatly increases the tracking difficulty.

Therefore, we propose a multi-target tracking pipeline (tracks while detecting ships and planes based on deep neural networks, called TDNet for short) that is composed of four modules, including object detection, multi-target tracking based on KCF (MT-KCF), detecting-assisted tracking (DAT), and new target recognition (NTR). The proposed tracking pipeline is processed as follows: (1) region-based fully convolutional networks (R-FCNs) [8] detect all the objects in the first frame of the video sequence and derive the coordinate values for all objects and numbers of these objects in-class; (2) the MT-KCF tracks all the detected targets; (3) the NTR recognises the newly emerged targets; (4) and the DAT helps to improve the tracking performance. TDNet was evaluated on optical remote sensing video data sets that were acquired by the Jilin No. 1 satellite. Since ships and planes are two important types of targets, we focused on these types of targets in this paper. We compared TDNet with state-of-the-art tracking methods to demonstrate the effectiveness of its pipeline and the modules within it.

The rest of the paper is organised as follows. Section 2 describes the background and preliminaries. The proposed TDNet is elaborated in Section 3. Section 4 reports and analyses the experimental results. The conclusions and future work are given in Section 5.

2. Preliminaries

In the detecting-assisted tracking (DAT) and new target recognition (NTR) modules, we recognised the ships and planes based on R-FCN [8], and we propose an MT-KCF approach based on KCF [16] to track them. We will first start by introducing R-FCN and KCF in this section.

2.1. R-FCN-Based Object Detection Approach

Recently, regions with CNN feature (R-CNN [23])-based object detection approaches [8] have been substantially improving upon the state-of-the-art methods in a wide range of computer vision applications. Their success is largely due to the advent of the backbone network, which is a deep convolutional network model trained on the ImageNet data set. Since R-FCN, with ResNet [24] as the backbone network, high quality features of a target can extracted; thus, we chose R-FCN with ResNet as the object detection network to complete the object detection task in DAT and NTR.

Figure 1 shows the architecture of R-FCN. $\boldsymbol{A}$ indicates the input. $\boldsymbol{B}$ is ResNet101, which is the backbone network in R-FCN (but it must be noted that $\boldsymbol{B}$ is different from the traditional ResNet101). $\boldsymbol{B}$ is constructed by removing the last full connected layer of the traditional ResNet101 and adding a full convolutional layer with a $1 \times 1 \times 1024$ size. $\boldsymbol{D}$ is

obtained after the RPN operation, which is a proposal extraction network that was proposed in Faster R-CNN [25], conducted on the last layer of ***B***. ***C*** is the position-sensitive score map, which is obtained by convolution of the last layer of ***B*** over $k * k(C+1)$ convolutional kernels with a size of $1024 \times 1 \times 1$, where C is the number of categories. ***E*** is obtained by the average pooling of regions of interest (RoIs) in ***C***. ***F*** is the result of voting by ***E***.

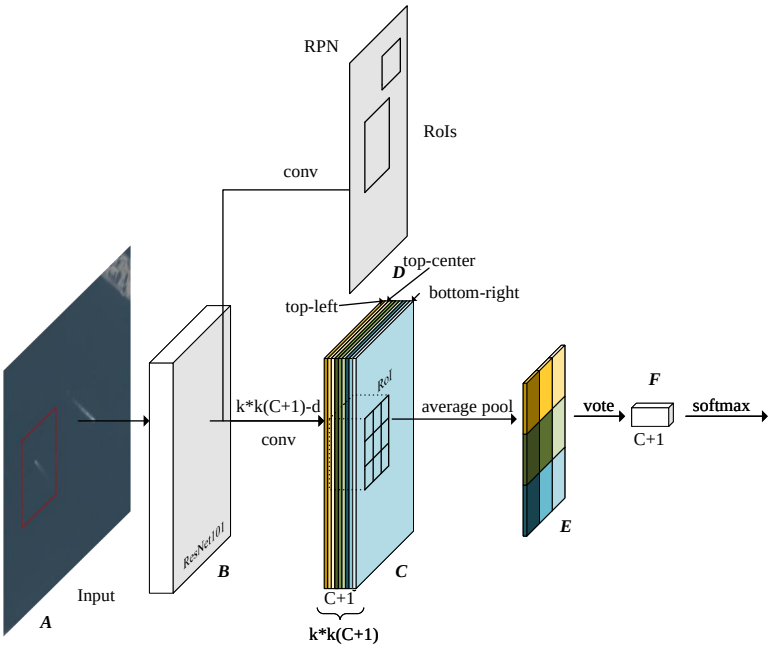


Figure 1. The architecture of R-FCN. $k * k$ represents the size of RoIs on the position-sensitive score map. C represents the number of categories. The *conv* symbol represents the convolution operation. Softmax is a multi-class classifier.

2.2. Kernel Correlation Filter Tracking

A common phenomenon in the field of target tracking is that tracking methods based on correlation filtering with traditional feature extractors are relatively fast in tracking speed; moreover, tracking methods that are based on correlation filtering incorporation with deep neural networks have relatively high tracking accuracy. Due to the special characteristics of our remote sensing video data, in which the frame per second (fps) speed was about 10 fps, we needed to control the tracking speed. KCF is a method with a relatively fast tracking speed and an acceptable tracking accuracy. In order to achieve a relatively high tracking speed, we chose to improve KCF to build a tracking method that fits our data. Therefore, in this subsection, we introduce the formulation of KCF.

2.2.1. Correlation Filter-Based Methods

The basic idea in correlation filter-based tracking methods is to design a template that maximises the output response by convolving the template with the RoIs of the tracked target. As shown in Figure 2, the idea can be described mathematically as follows:

$$g = f \otimes h, \tag{1}$$

where $\otimes$ represents a convolution operation, g represents the output response, f indicates the gray-scale image of the input image, and h represents the filter template. We only need to constantly modify the filter template to obtain the maximum output response. In these tracking methods, the output response can be calculated via a convolution operation between the filter template and the RoI of the tracked target.

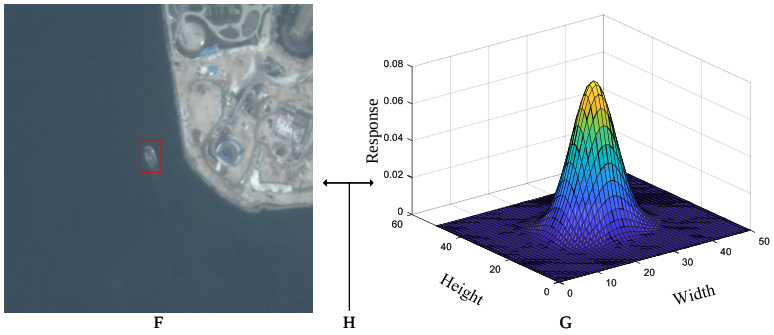


Figure 2. Schematic of a tracking method based on a correlation filter.

To increase the tracking speed, MOSSE [13] uses the fast Fourier transform (FFT), which converts the convolution operations in the real number domain to an element-wise multiplication in the Fourier domain, and this is performed to decrease the computational complexity. The specific formula for this is as follows:

$$\mathcal{F}(g) = \mathcal{F}(f \otimes h) = \mathcal{F}(f) \odot \mathcal{F}(h)^*, \tag{2}$$

With respect to the above, we defined $G = \mathcal{F}(g)$, $F = \mathcal{F}(f)$, and $H = \mathcal{F}(h)$ with $\mathcal{F}(\cdot)$, and we denoted the FFT operator, where the above formulation can be abbreviated as follows:

$$G = F \odot H^*, \tag{3}$$

where * indicates a complex conjugate and $\odot$ denotes element-wise multiplication. Thus, the filter template in the Fourier domain is obtained as follows:

$$H^* = \frac{G}{F}. \tag{4}$$

Then, the objective function can be expressed as follows:

$$\min_{H^*} = \sum_{i=1}^{m} |H^* F_i - G_i|^2 \tag{5}$$

where F_i denotes the result of the FFT operator on the training images, f_i denotes those in the Fourier domain, and G_i denotes the result of the FFT on the training outputs g_i in the Fourier domain. By minimising the objective function, we obtain the following:

$$H^* = \frac{\sum_m^{i=1} F_i \odot G_i^*}{\sum_m^{i=1} F_i \odot F_i^*}. \tag{6}$$

Then, the position where the maximum response between the template and RoI is found is the position of the target in the current frame.

2.2.2. KCF

Based on MOSSE, CSK provides a combination of samples and circulant matrices on the one hand. On the other hand, it combines the kernel function, which helps to map the original linear space problem into nonlinear space and thus solves the low-dimensional linear inseparability problem. Compared to the above methods, KCF contributes to feature selection and multi-channel feature kernel correlation, which are detailed as follows.

Assume a training sample set $(\boldsymbol{x}_i, y_i)$ with a Ridge regression function $f(\boldsymbol{x}_i) = w^T \boldsymbol{x}_i$, where $\boldsymbol{x}_i$ and y_i denote the samples and their regression targets, respectively. Moreover, $\boldsymbol{w}$ represents the weight coefficient, which can be optimised as follows:

$$\min_{\boldsymbol{w}} \sum_i (f(\boldsymbol{x}_i) - y_i) + \lambda \|\boldsymbol{w}\|^2, \tag{7}$$

where λ is used to control the importance of the two terms, with the second term being the structural complexity of the system, which is used to control overfitting. The close-form solution of Equation (15) is as follows:

$$\boldsymbol{w} = (X^T X + \lambda I)^{-1} X^T \boldsymbol{y}, \tag{8}$$

where matrix X has one sample per row $\boldsymbol{x}_i$ and each element in $\boldsymbol{y}$ is a regression target y_i. Moreover, I is an identity matrix. By converting Equation (8) into a complex version, we obtain the following:

$$\boldsymbol{w} = (X^H X + \lambda I)^{-1} X^H \boldsymbol{y}, \tag{9}$$

where X^H is the Hermitian transpose, i.e., $X^H = (X^*)^T$, and X^* is the complex conjugate of X.

In general, a large system of linear equations must be solved to compute the solution; however, this type of approach is time-consuming. To meet the standards of real-time tracking, KCF also introduces a special case of x_i. Consider an image patch with the object of interest being denoted by a vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ as a base sample; as such, KCF reconstructs matrix X by cycle shifts $\boldsymbol{x}$ as follows:

$$X = C(\boldsymbol{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}, \tag{10}$$

where X is a circulant matrix. As all circulant matrices are made diagonal by the discrete Fourier transform (DFT), we obtain the following:

$$\hat{\boldsymbol{w}} = \frac{\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{y}}}{\hat{\boldsymbol{x}}^* \odot \hat{\boldsymbol{x}} + \lambda} \tag{11}$$

where the hat ˆ denotes the DFT of a vector. The fraction denotes element-wise division.

In addition, by projecting the inputs of a linear problem to a non-linear feature-space $\varphi(\boldsymbol{x})$ with the kernel tricks, i.e., $\boldsymbol{w} = \sum_i \alpha_i \varphi(\boldsymbol{x}_i)$, the variables under optimization are $\boldsymbol{\alpha}$ instead of $\boldsymbol{w}$. By introducing a kernel function, the solution to the kernelised version of the Ridge regression in [15] is as follows:

$$\boldsymbol{\alpha} = (K + \lambda I)^{-1} \boldsymbol{y}, \tag{12}$$

where K is kernel matrix and $\boldsymbol{\alpha}$ is the vector of coefficient α_i. In KCF, K represents the circulant matrix for the data sets of cyclic shifts. As such, by diagonalising Equation (12), we obtain the following:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\boldsymbol{y}}}{\hat{\boldsymbol{k}}^{\boldsymbol{xx}} + \lambda}, \tag{13}$$

where $\boldsymbol{k}^{\boldsymbol{xx}}$ denotes the first row of the kernel matrix $K = C(\boldsymbol{k}^{\boldsymbol{xx}})$.

In the next frame, we can compute the regression function for all candidate patches with the following:

$$\hat{f}(z) = \hat{\boldsymbol{k}}^{\boldsymbol{xz}} \odot \hat{\boldsymbol{\alpha}}. \tag{14}$$

As for the details of the kernel values $\boldsymbol{k}^{\boldsymbol{xx}}$ and $\boldsymbol{k}^{\boldsymbol{xz}}$, readers can refer to [16] for more details. Intuitively, evaluating $\hat{f}(z)$ at all locations can be seen as a spatial filtering operation over the kernel values $\boldsymbol{k}^{\boldsymbol{xz}}$. Each $\hat{f}(z)$ is a linear combination of the neighboring kernel values from $\boldsymbol{k}^{\boldsymbol{xz}}$, and they are weighted by the learned coefficients $\boldsymbol{\alpha}$. Since this is a filtering operation, it can be formulated more efficiently in the Fourier domain. Through the inverse Fourier transform, the final corresponding response in the real domain will be obtained.

3. The Proposed TDNet

Figure 3 shows the proposed pipeline for multi-target tracking. Figure 4 shows the flowchart of the proposed method. The whole architecture consists of four modules: object detection; multi-target tracking (MT-KCF); detecting-assisted tracking (DAT); and new emerging target recognition (NTR). The four modules in TDNet are processed as follows: (1) The pre-trained R-FCN is used to detect the ships and planes in the first frame of a video so as to obtain the coordinate information and category information of all the targets to be tracked. Then, all the detected targets are numbered in-class as the basis for MT-KCF. (2) MT-KCF is used to accomplish multi-target tracking. (3) DAT is used to assist the tracking process, such as, for example, tracking lost targets. (4) NTR is used to realise the recognition of new targets.

In this section, we mainly describe the four modules in TDNet that were mentioned above: object detection, MT-KCF, DAT, and NTR.

3.1. Object Detection

We use pre-trained R-FCN, whose backbone network is ResNet101, to complete the object detection task in TDNet. Here, we introduce R-FCN from two aspects: data set composition and sample selection, as well as parameter setting and optimization.

3.1.1. Data Set Composition and Sample Selection

R-FCN should be first trained with labeled samples; therefore, we split all the videos in the data set by frame. Since the resolution of the Jilin No. 1 optical remote sensing video satellite is relatively low and the target's motion speed in the video is relatively slow, we selected an image every 20 frames as the training image. Experiments have shown that selecting a sample every 20 frames can satisfy the demand well. We labeled the object proposals via the intersection over union (IoU) parameter, which is the area overlap ratio between the proposal and ground truth. If the IoU is in the range of [0.5, 1], the corresponding proposal is defined as a positive sample; otherwise, the proposal is defined as a negative sample.

3.1.2. Parameter Setting and Optimisation

The loss function defined on each region of interest (RoI) was composed of two parts: the classification part and the regression part (i.e., $L(c, z_{x,y,w,h}) = L_{cls}(c_{c^*}) + \mu[c^* > 0]L_{reg}(z, z^*)$). Among them, c^* is the label of an RoI (where $c = 0$ represents the RoI as a background), L_{cls} is the cross-entropy loss for classification, L_{reg} is the bounding boxes loss for regression, and z^* is the coordinates of the ground-truth box. In order to obtain better detection results, we used an online hard example mining (OHEM) strategy to accomplish training. In the training process, we set the balance weight μ to 1 and the learning rate for the first 20k mini-batches to 0.001; in addition, the learning rate was reduced 10 times for each additional 10k mini-batches and a total of 40 k mini-batches were trained.

We used the pre-trained R-FCN to detect the ships and planes in the first frame of the video; furthermore, we also obtained the coordinate information $(x_{1i}, y_{1i}, x_{2i}, y_{2i})$ and category information c_i of all the objects to be tracked. Then, the coordinate information was converted into the top left coordinate (x_{1i}, y_{1i}), as well as into the height and width coordinate (h_{1i}, w_{1i}), where $i \in m$ and m represents the number of all detected objects. Next, we performed in-class numbering for all m targets and used the numbered information as the basis for multi-object tracking. Figure 5 shows the object detection results and the in-class numbering results.

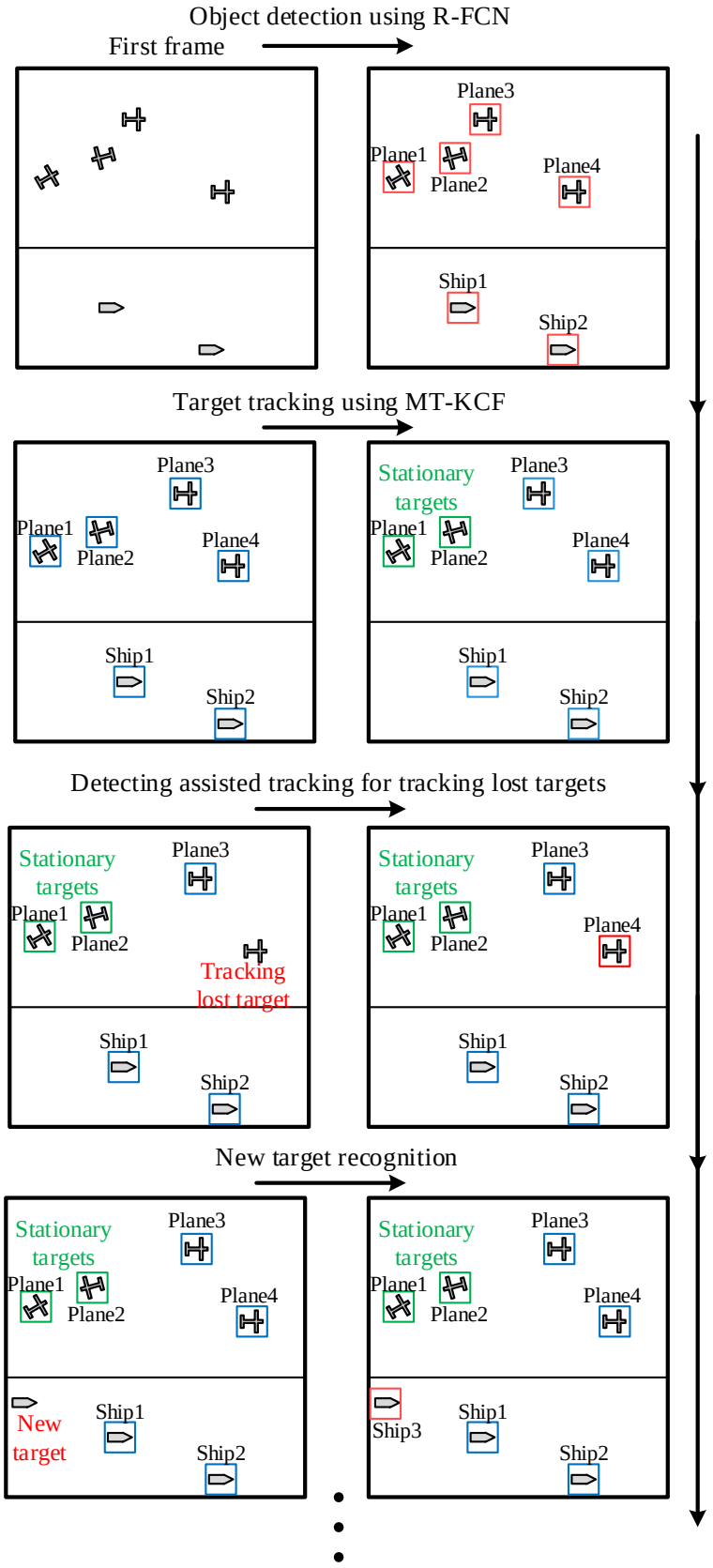


Figure 3. The proposed pipeline.

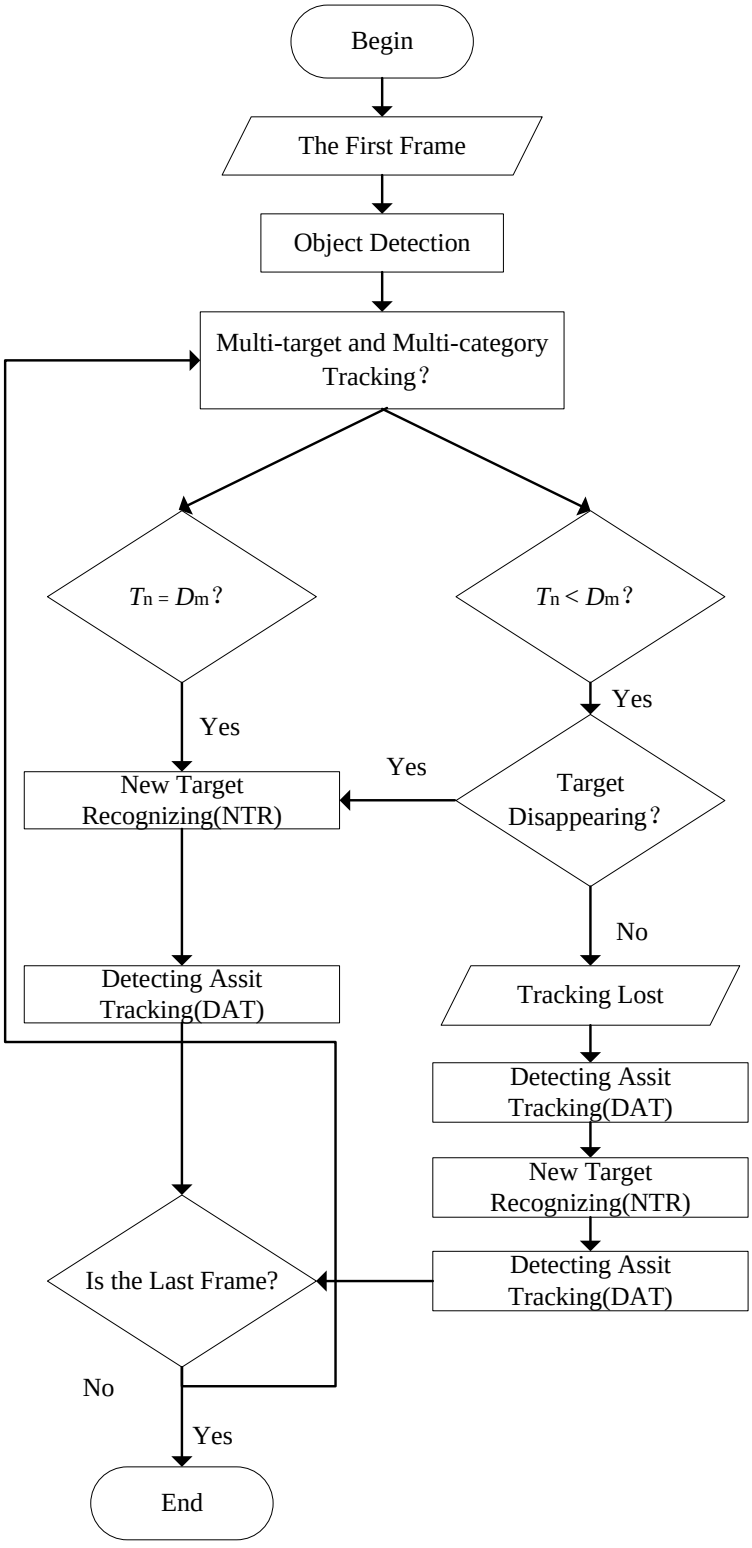


Figure 4. The flowchart of the proposed pipeline.

(**a**) (**b**) (**c**) (**d**)

Figure 5. (**a**,**c**) The detection results of R-FCN for planes and ships. (**b**,**d**) The in-class numbering results according to the detection results.

3.2. Multi-Target Tracking (MT-KCF)

Since KCF [16] is an excellent single-target tracking method, we proposed a multi-target tracking method named MT-KCF (which is based on KCF) to solve the problem where there are usually multiple targets. The structure of MT-KCF is shown in Figure 6.

Let us take a video as an example. We assumed that a total of m objects $(p_1, p_2, ..., p_q, s_{q+1}, s_{q+2}, ..., s_m)$ were detected in the first frame of the video, where q is the number of planes, namely, trackers $(trp_1, trp_2, ..., trp_q, trs_{q+1}, trs_{q+2}, ..., trs_m)$. Taking a plane target t as an example, the tracker training sample set is $(\boldsymbol{p}_t, \boldsymbol{y}_t)$, where $\boldsymbol{p}_t$ denotes the training sample set of the t-th target (plane) and $\boldsymbol{y}_t$ is their regression set. Then, the following KCF, where the optimal $\boldsymbol{w}_t$ of the t-th target is obtained by solving the optimization problem, is obtained:

$$\min_{\boldsymbol{w}_t} \sum_i (f(\boldsymbol{p}_t) - \boldsymbol{y}_t) + \lambda \|\boldsymbol{w}_t\|^2. \tag{15}$$

As in KCF, $\boldsymbol{w}_t$ can be optimised by solving $\boldsymbol{\alpha}$ in the nonlinear space $\varphi(\boldsymbol{p}_t)$, which is achieved via following Equations (8)–(13). There are m targets and, in order to track them simultaneously, m filters $(\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_m)$ corresponding to each target should be learned.

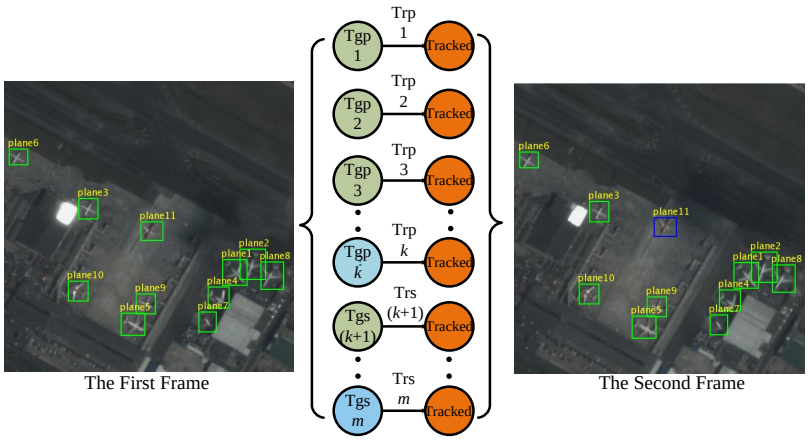


Figure 6. The basic structure of MT-KCF. The green box indicates the stationary targets, while the blue box indicates the moving targets.

After $\boldsymbol{w}_t$ is trained, it is used to predict the location of the t-th target in the next frame. Suppose $\boldsymbol{z}$ is a test batch of the target in the next frame. Then, we obtain the response of $\boldsymbol{z}$ as follows:

$$\boldsymbol{f}_t(\boldsymbol{z}) = \hat{\boldsymbol{k}}^{\boldsymbol{p}_t \boldsymbol{z}} \odot \hat{\boldsymbol{\alpha}} = (\hat{\boldsymbol{k}}^{\boldsymbol{p}_t \boldsymbol{z}} \odot \frac{\hat{\boldsymbol{y}}_t}{\hat{\boldsymbol{k}}^{\boldsymbol{p}_t \boldsymbol{p}_t} + \lambda}), \tag{16}$$

$$\text{Response}(\boldsymbol{z}) = \text{real}(F^{-1}(\boldsymbol{f}(\boldsymbol{z}))), \tag{17}$$

where F^{-1} represents the inverse Fourier transform; $\hat{\boldsymbol{k}}^{\boldsymbol{p}_t\boldsymbol{z}}$ and $\hat{\boldsymbol{k}}^{\boldsymbol{p}_t\boldsymbol{p}_t}$ represent the kernel values satisfying the relevant conditions as defined in KCF; and $\odot$ indicates the Hadamad product of the vector. For multi-target tracking, m targets will response simultaneously $(\boldsymbol{f}_1(\boldsymbol{z}), \boldsymbol{f}_2(\boldsymbol{z}), \ldots, \boldsymbol{f}_m(\boldsymbol{z}))$. The position where the maximum $\boldsymbol{f}_t(\boldsymbol{z_i})$ is found is the position of the target t in the current frame, where $i \in N$ and N is the number of samples selected in the current frame for a special target. The location of the m maximum responses are the positions of the m-tracked targets in the current frame. MT-KCF is adaptive to changes at the target scale. Moreover, the width w and the height h of the targets change if the next DAT operation is performed. Through this step, we obtained the position information (x_j, y_j, h_j, w_j) of all the targets, where (x_j, y_j) represent the center coordinates of the j-th target and $j \in n$. In general, $n = m$.

In order to better accomplish the multi-target tracking tasks, we should not only track the moving targets, but also detect the stationary targets. Due to the fact that the box surrounding the stationary objects should stay the same in different frames, we used the same box when an object is recognised as a stationary object to avoid gradual mistracking. In order to achieve this function, we used indicators d_o and s_o to determine the stationary objects. In detail, if d_o is over 0 and s_o is less than 0.95, the target is considered a moving target; otherwise, the target is determined to be a stationary target. The indicators are formulated as follows:

$$d_o = \sqrt[2]{(x_f - x_{f-1})^2 + (y_f - y_{f-1})^2}, \tag{18}$$

$$s_o = \frac{S_f \cap S_{f-1}}{S_f \cup S_{f-1}}, \tag{19}$$

where the indicator d_o is formulated as the Euclidean distance between the center coordinate (x_f, y_f) of the target in the current frame and the center coordinate (x_{f-1}, y_{f-1}) of the target in the previous frame. S_{f-1} denotes the area of the target in the $(f-1)$-th frame and S_f denotes the area of the target in the f-th frame.

Figure 7 shows an example of multi-target tracking where static targets are surrounded by green boxes and moving targets are marked by blue boxes with azure lines as their trajectories. The boxes of statistic targets should stay as statistics. However, due to the slight variance in the learned filter, there may be certain errors that lead to changed boxes. Therefore, we recognise the statistic targets via the location of their center and fix the changed boxes. The behaviour of the moving targets can be analysed by their trajectories. For example, the trajectory of Plane 5 was found to be longer than others (which meant that Plane 5 was taking off).

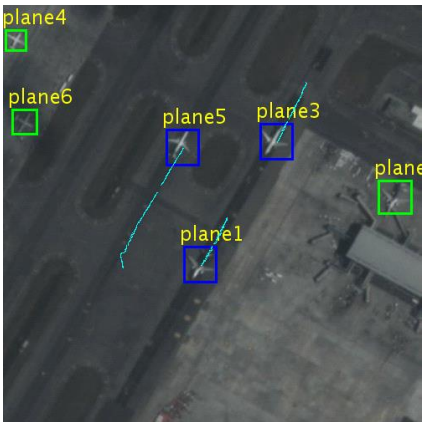


Figure 7. An example of multi-target tracking, where the azure lines denote the trajectories of the moving targets.

3.3. Detecting-Assisted Tracking (DAT)

The detecting-assisted tracking (DAT) module was introduced due to two motivations: (1) to improve the target tracking accuracy and precision of TDNet, as by introducing DAT to continuously correct the coordinates of the tracked targets, the tracking lag occurrence can be prevented, and thus, the performance will be significantly increased; (2) to retrieve the "Tracking Lost" targets. By introducing DAT to detect certain frames of the video, the lost targets can thus be retrieved in time.

Because DAT was used to constantly correct the tracked targets' position, it was important to match the position information and the category information of the detected objects with the position information and IDs obtained by MT-KCF in the same frame. As such, we propose a strategy called DAT-MS to match these two kinds of information. The DAT process was performed every 10 frames unless a "Tracking Lost" target was found.

Assume that the f-th frame needs to perform the DAT operation. Suppose that the position information and IDs of all tracked targets in the fth frame are $(x_v, y_v, h_v, w_v, c_v)$, where n indicates the number of tracked targets, c_v is the ID of the vth tracked target, and $v \in [1, n]$. After the object detection operation on the f-th frame, the position information and the category information for all detected objects are obtained as $(x_r, y_r, h_r, w_r, c_r)$, where m represents the number of objects detected by R-FCN, c_r is the category of the r-th detected object, and $r \in [1, m]$. In order to establish a one-to-one correspondence between the location information and IDs obtained from MT-KCF, as well as between the location information and category information obtained from R-FCN, we define DAT-MS (d_v and d_r) as follows:

$$d_v = \min_r \| x_v - x_r \|_2^2 + \| y_v - y_r \|_2^2, v = 1, 2, ..., n, \tag{20}$$

which helps us to find the object among all the m detected objects that best matches v from the distance. If the category in c_r is the same, with the category in c_v and d_v being less than a certain threshold γ, we determine that the target r and the object v have a one-to-one correspondence. Then, the target's location information is updated based on the object detection result. If a v cannot find the corresponding r, this target maintains the original information.

$$d_r = \min_v \| x_r - x_v \|_2^2 + \| y_r - y_v \|_2^2, r = 1, 2, ..., m. \tag{21}$$

In addition to the successfully matched targets that are obtained through Equation (20), if an r cannot find the corresponding v through Equation (21), we determine this target as a "Tracking Lost" target. The "Tracking Lost" targets are then recovered by the information from previous frames. However, if we still have not found the information of the detected target after going back 10 frames, we consider the detected target as a "New Target" (we will introduce new target recognition in the next subsection). Moreover, if the existing target disappears, the IDs of the other tracked targets remained unchanged. Figure 8 shows the basic structure of DAT. In addition to disappearing targets, there are also emergent targets, which should be recognised during the tracking process.

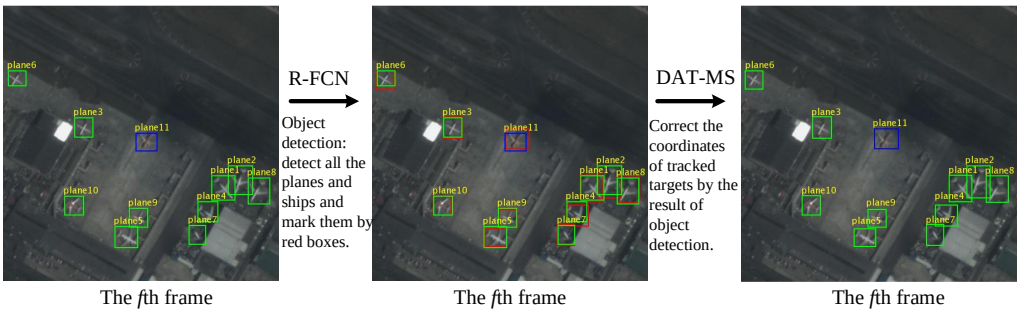


Figure 8. The basic structure of DAT. The green boxes indicate the stationary target and the blue boxes indicate the moving targets. The red boxes indicate the R-FCN-based detection results.

3.4. New Target Recognition (NTR)

We also need to consider the case where new targets appear in a current frame. The module of new target recognition (NTR) was used to recognise the newly emerged targets, and it was implemented every five frames in order not to harm the tracking speed. Existing new target recognition methods usually recognise the newly emerged targets frame-by-frame [26]. However, in remote sensing videos, the targets are blurry but move relatively slowly. Therefore, the recognition should have a high accuracy, but the time it takes to achieve this is extensive. For the Kalman-based multi-target tracking method, if the target is not detected in this frame, it will be treated as a new target if it is detected in the next frame. Moreover, existing methods fail to handle stationary targets well.

Due to the low resolution of remote sensing videos, the speed of the targets in the video will not be too fast, and there is no need to recognise the newly emerged target frame-by-frame. R-FCN is applied every five frames, which decreases the computational time. Then, a matching strategy similar to Equation (21) is used to find the targets that are newly appearing in the scene. As described above, the "New Target" is defined according to the information of the previous frames.

If a new target appears, then the interclass number for a certain category increases by 1, and the new target will be labeled and tracked. We call this process NTR-MS. Figure 9 shows the basic structure of NTR.

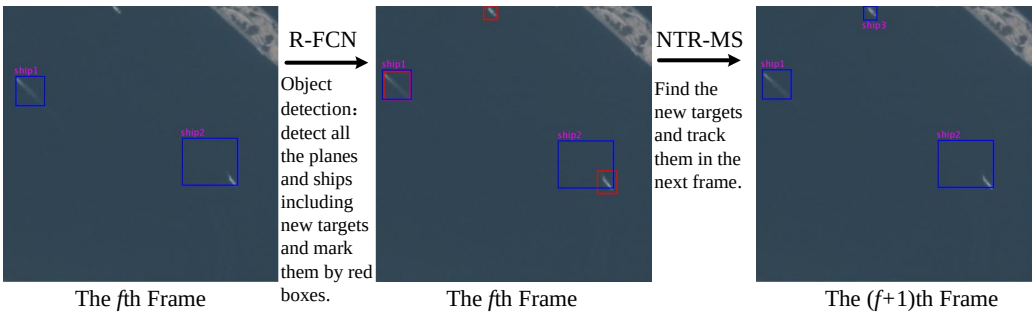


Figure 9. The basic structure of NTR. The blue boxes indicate the moving targets. The red boxes indicate the R-FCN-based detection results.

4. Experimental Results

In this section, we mainly introduce the following aspects: data sets, evaluation metrics, and performance comparison.

4.1. Data Sets

We evaluated the performance of the proposed TDNet-based target tracking methods on data sets that were obtained from optical remote sensing video of the Jilin No. 1 commercial remote sensing satellite group, where the ground pixel resolution was 1.12 m and the duration was about 30 s. Jilin No. 1 commercial satellites are China's first self-developed commercial remote sensing satellite group. It was developed by the Changchun Institute of Optics, Fine Mechanics and Physics of the Chinese Academy of Sciences, and it was launched by a Long March No. 2 carrier rocket at Jiuquan Satellite Launch Center at 12:13 on 7 October 2015. This satellite group operates in sun-synchronous orbits with an average orbit height of 650 km.

As the ground pixel resolution of the video satellite is only 1.12 m, it is challenging to achieve target tracking in these data sets. In our work, we chose three cities, Santiago, Bogota, and Hong Kong, as well as two categories of plane and ship for the experiments. Since the size of each of the original videos was relatively large, and as there were fewer areas where the target was densely distributed, we deducted densely targeted size $500 \times 500 \times 3$ areas as our data sets. We selected a total of six data sets (Video 1, Video 2, Video 3, Video 4, Video 5, and Video 6), of which three data sets (Video 1, Video 2, and Video 3) were used for single-moving-target tracking experiments and four data sets

(Video 3, Video 4, Video 5, and Video 6) were used for multi-moving-target tracking experiments. The multi-moving-target tracking experiments were divided into three types: a single-category and multi-moving-target tracking (SC_MT) experiment on Video 3 and Video 4; a multi-category and multi-moving-target tracking (MC_MT) experiment on Video 5; and a new emerging target recognition (NTR) and tracking experiment on Video 6. Figure 10 shows several of the images in each data set. With the exception of Video 6, the other images are the first frame of the corresponding data set. Images in the first, tenth, and twenty-second frames in Video 6 are shown in Figure 10. Table 1 shows the detailed information of each data set. For the target on the boundary line, we determined whether the target was included in the statistical scope according to the position of the target's center of gravity.

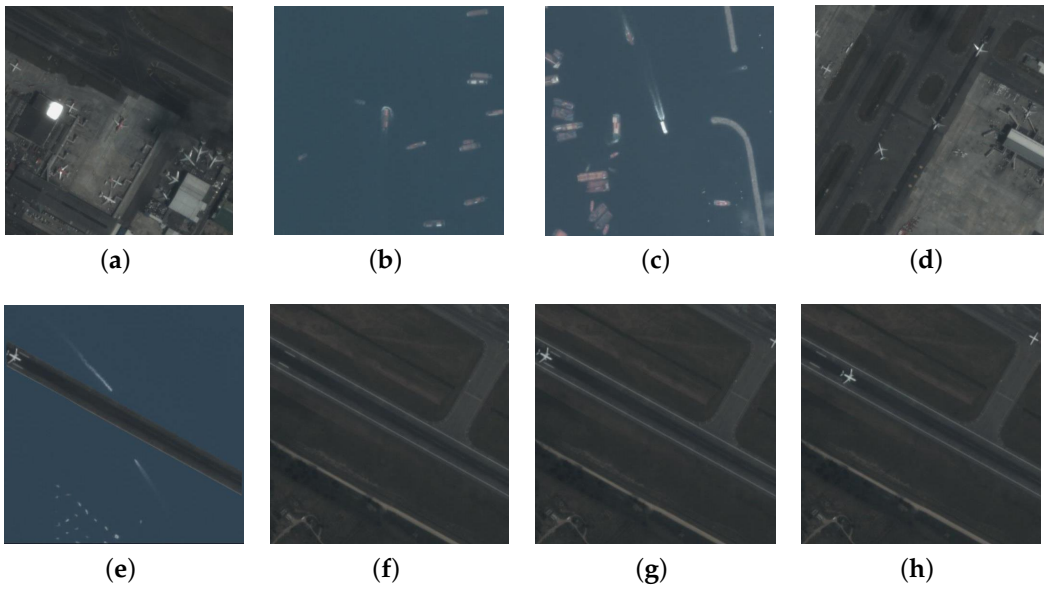


Figure 10. The total six data sets. (**a**) The first frame of Video 1. (**b**) The first frame of Video 2. (**c**) The first frame of Video 3. (**d**) The first frame of Video 4. (**e**) The first frame of Video 5. (**f**–**h**) The first, tenth, and twenty-second frames of Video 6.

It is worth noting that Video 5 is an artificial data set. When we were researching the target tracking method for remote sensing videos, the airport was usually not near the harbor. There was no scene in the data set where the planes and the ships could be separated into the same video data set. In order to conduct the multi-target tracking experiments, we created a data set of such images by splicing together airports and harbors.

As introduced in Section 3, R-FCN should be first trained on annotated data in order to accurately recognise and locate the corresponding targets. In this paper, we followed the basic training process of R-FCN [8], where it was first pre-trained on the ImageNet data set [27], which is a large-scale image data set. Then, the layers of D, E, and F, as shown in Figure 1, were fine-tuned with labeled frames that were extracted from a training video captured on a Jilin No. 1 commercial remote sensing satellite. A training frame is exhibited in Figure 11, where many patches were extracted to train the proposal network, which was used to generate the proposal boxes that contain targets. The samples include patches containing targets and no targets. The classifier was trained on image patches with a single object, and these were used to distinguish different targets, as well as the background. As introduced above, a frame was selected in the consecutive 20 frames, which was found to be enough to adequately train R-FCN.

Table 1. Data set details.

The Six Data Sets						
	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6
Category	plane	ship	ship	plane	ship and plane	plane
Moving targets	1	1	3	3	3	2
Stationary targets	10	11	16	3	0	0
Total	11	12	19	6	3	2

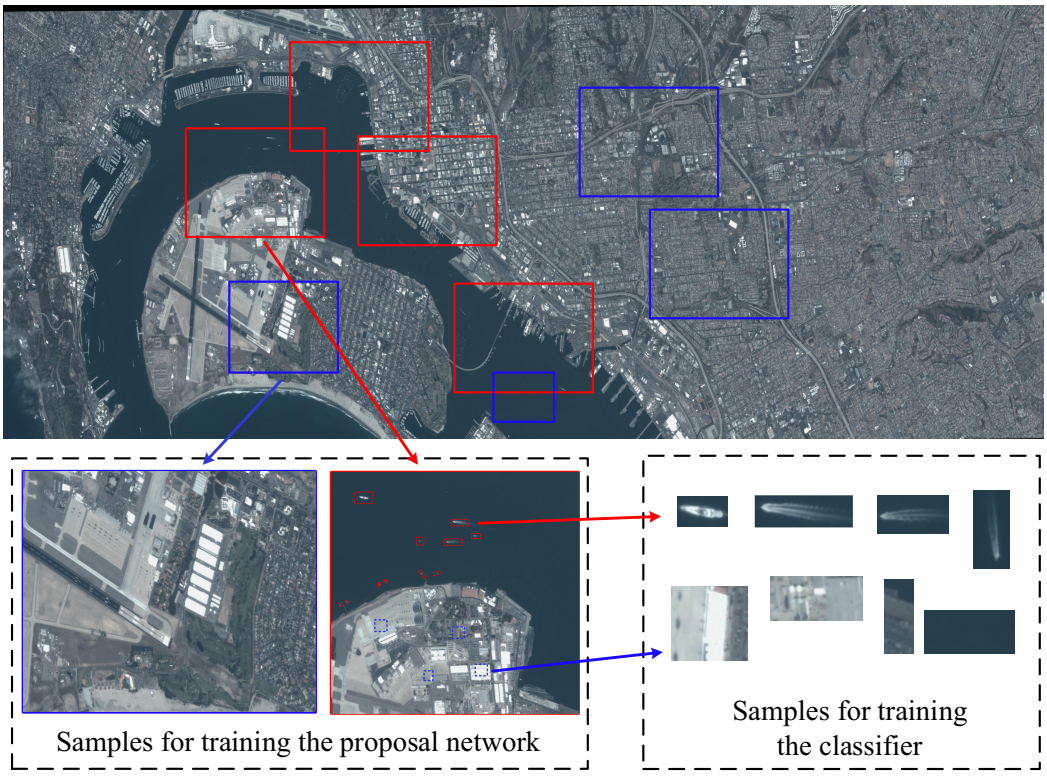


Figure 11. Illustration of a training frame for R-FCN.

4.2. Evaluation Metrics

Regarding the single-moving-target tracking tasks, we used the evaluation metrics of mean precision and mean frames per second (fps), which were also used in [16]. These metrics were used as the basis to compare the performance of our method with that of the compared methods. Since the performance of our tracking method depends on the performance of the object detection method to a certain extent, we used the index mean average precision (mAP) metric (which is frequently used in object detection methods [8,23,28]) to evaluate the performance of the object detection method when used in TDNet. The mAP metric represents the area under the precision vs. recall curve (PRC) of the object detection method [29]. The definitions of precision (P) and recall (R) in ref. [29] are as follows:

$$P = \frac{TP}{TP + FP}, \tag{22}$$

$$R = \frac{TP}{TP + FN}, \tag{23}$$

where TP, FP, and FN denote the number of true positives, the number of false positives, and the number of false negatives, respectively.

Regarding the multi-moving-target tracking tasks, we adopted the CLEAR MOT metrics, which are widely used to evaluate the performance of multi-moving-target tracking methods [30–32]. Furthermore, they were also utilised to quantitatively evaluate the performance of the multi-moving-target tracking methods used in our work. The CLEAR MOT metrics include multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), mostly tracked (MT—i.e., the percentage of ground truth objects whose trajectories are covered by a tracking output this is at least 80%) targets, mostly lost (ML—i.e., the percentage of ground truth objects whose trajectories are covered by a tracking output that is less than 20%) targets, the total number of false positives (FP), the total number of false negatives (FN), the total number of ID switches (IDS), and the number of frameworks processed in one second (Hz). MOTA and MOTP can be formulated as follows:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \tag{24}$$

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}, \tag{25}$$

where t represents the current frame, m_t is the missing numbers in the t-th frame, fp_t is the misjudged numbers in the t-th frame, $mmme_t$ is the mismatched numbers in the t-th frame, g_t is the number of tracked targets in the t-th frame, c_t is the number of matched targets in the t-th frame, and d_i^t is the calculated matching errors for each pair of matches in the t-th frame.

4.3. Performance Comparison of the Single-Moving-Target Tracking Experiments

We compared our methods (TDNet(DAT) with DAT and TDNet(NO DAT) without DAT) with seven other state-of-the-art single-target-tracking methods (KCF [16], C-COT [19], CASAMF [33], CAMOSSE [33], CASTAPLE [33], ECO-HC [12], and ECO [12]) in terms of mean precision (Mean P) and mean fps (fps) on Video 1, Video 2, and Video 3 for plane11, ship6, and ship4, respectively. TDNet(NO DAT) was the method (TDNet) without a DAT module (Table 2).

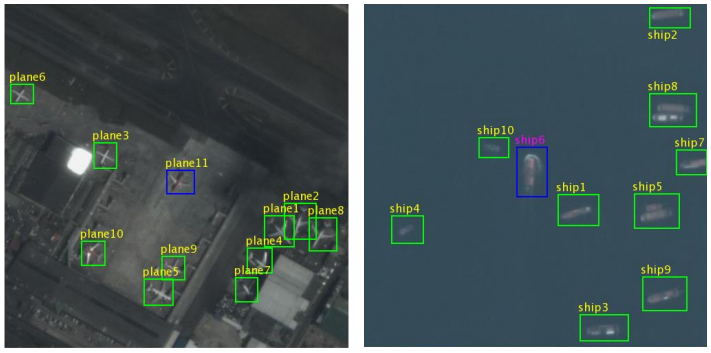
Table 2. Single-moving-target tracking comparisons.

Video **1**									
	CASAMF [33]	CAMOSSE [33]	KCF [16]	CASTAPLE [33]	C-COT [19]	ECO-HC [12]	ECO [12]	TDNet(NO DAT)	TDNet(DAT)
Mean P	74.71%	77.01%	89.82%	92.29%	92.76%	93.08%	93.81%	89.82%	**94.65%**
Fps	6.08	89.99	93.59	14.32	0.20	15.89	1.06	**93.59**	82.84
mAP(**od**)	-	-	-	-	-	-	-	100%	100%
Video **2**									
	CASAMF [33]	CAMOSSE [33]	KCF [16]	CASTAPLE [33]	C-COT [19]	ECO-HC [12]	ECO [12]	TDNet(NO DAT)	TDNet(DAT)
Mean P	92.06%	93.44%	93.89%	93.94%	93.97%	94.16%	94.69%	93.89%	**97.25%**
Fps	6.02	90.49	90.60	14.85	0.15	16.89	1.05	**90.60**	79.34
mAP(**od**)	-	-	-	-	-	-	-	90.91%	90.91%
Video **3**									
	CASAMF [33]	CAMOSSE [33]	KCF [16]	CASTAPLE [33]	C-COT [19]	ECO-HC [12]	ECO [12]	TDNet(NO DAT)	TDNet(DAT)
Mean P	87.91%	92.51%	92.75%	93.78%	93.80%	93.85%	95.39%	92.75%	**97.69%**
Fps	8.64	86.43	125	15.00	0.19	18.02	1.05	**125**	97.59
mAP(**od**)	-	-	-	-	-	-	-	94.74%	94.74%

There was no "Tracking Lost" phenomenon observed on these data sets for all of the methods. TDNet(DAT) performed better than the compared methods in terms of mean P (94.65% on Video 1, 97.69% on Video 2, and 97.25% on Video 3). In terms of tracking speed, our tracking methods were not the fastest, but they did demonstrate an acceptable level

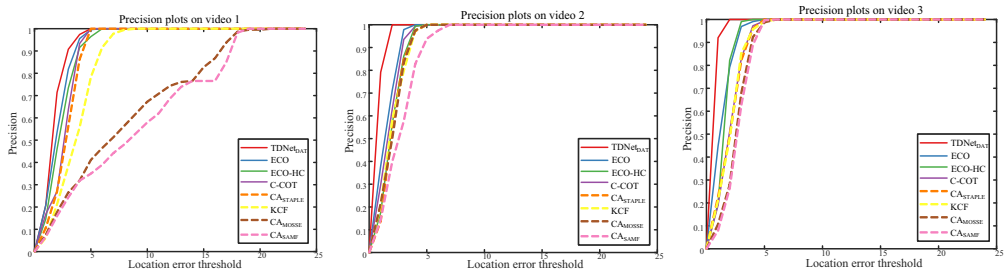
with a mean fps of 82.84. In addition, due to the scale among the targets being essentially uniform (in addition to several ships), the targets could easily be detected by the object detection method. Moreover, the mAP of R-FCN in Video 1 was 1.

Figure 12 shows the tracking results of the TDNet$_{(DAT)}$-based method on Video 1 and Video 2. There were eleven planes in Video 1, in which there was one moving target and ten stationary targets. TDNet$_{(DAT)}$ detected a total of eleven planes, in which there was one moving target and ten stationary targets. There were twelve ships in Video 2, which contained one moving target and eleven stationary targets. TDNet$_{(DAT)}$ detected a total of ten targets, with one moving, nine stationary, and two missed targets. Table 3 shows the state statistics for TDNet$_{(DAT)}$ when used on Video 1, Video 2, and Video 3. By comparing TDNet$_{(NO\ DAT)}$ with TDNet$_{(DAT)}$, it could be found that the introduction of DAT was able to improve the tracking performance. Figure 13 shows the precision plots for TDNet$_{(DAT)}$ and the compared methods. It is worth noting that TDNet$_{(NO\ DAT)}$ was the same as KCF for single-moving-target tracking performance. Therefore, the precision plots no longer drew the method TDNet$_{NO\ DAT}$. Due to the relatively large targets and simple backgrounds in Videos 1, 2, and 3, it was found to be less challenging for the methods to track them correctly. When the pixel constraints between the center points were slowly relaxed, the accuracy was close to one. This experiment proved the effectiveness of DAT in TDNet in a single-moving-target tracking experiment.



(**a**) (**b**)

Figure 12. Visualization of the SC_ST tracking results as captured via TDNet$_{(DAT)}$ on Video 1 and Video 2.



(**a**) (**b**) (**c**)

Figure 13. Precision plots of the moving target as captured by TDNet$_{DAT}$ and the other seven compared methods on (**a**) Video **1**, (**b**) Video **2**, and (**c**) Video **3**.

Table 3. State Statistics.

State statistics						
	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6
Category	plane	ship	ship	plane	plane and ship	plane
$TDNet_{DAT}$ (moving targets)	1	1	3	3	3	2
$TDNet_{DAT}$ (stationary targets)	10	**9**	**15**	3	0	0

4.4. Performance Comparison of the Multi-Moving-Target Tracking Experiments

TDNet can not only achieve single-moving-target tracking tasks, but also achieve single- and multi-category and multi-moving-target tracking tasks and monitor the newly emerging targets in a scene. Therefore, in this part of the paper, we designed three kinds of experiments as follows: single-category and multi-moving-target tracking (SC_MT tracking); multi-category and multi-moving-target tracking (MC_MT tracking); and new target recognition and tracking (NTR tracking). For these tasks, we compared our methods with the Markov decision process (MDP) [34]. $TDNet_{(NO\ DAT)}$ was the method (TDNet) that was used without a DAT module.

4.4.1. SC_MT Tracking

To verify the TDNet-based SC_MT tracking performance, we compared our method with the MDP on Video 3 and Video 4.

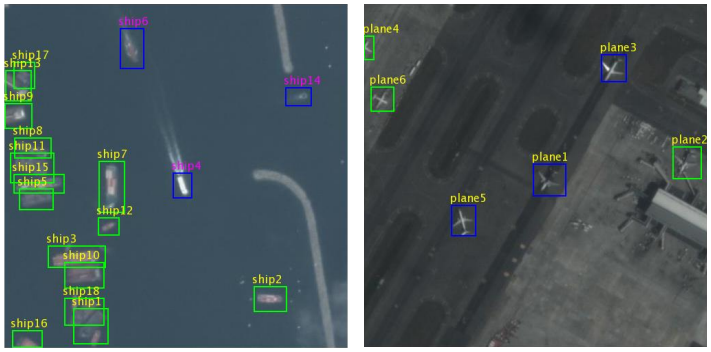
Table 4 shows the SC_MT tracking performance comparison results. The values in the table are the average values of Video 3 and Video 4 for every metric.

Table 4. Single-category and multi-target tracking comparisons.

SC_MT Tracking on Video 4 and Video 3								
Methods	**MOTA**	**MOTP**	**MT**	**ML**	**FP**	**FN**	**IDS**	**Hz**
MDP [34]	80.21%	87.31%	90.37%	3.21%	0	421	3	8.52
CenterTrack [10]	82.10%	87.62%	91.25%	2.43%	0	411	2	21.80
$TDNet_{(NO\ DAT)}$	84.62%	89.41%	91.56%	1.06%	0	398	0	**11.36**
$TDNet_{(DAT)}$	**85.31%**	**91.38%**	**93.41%**	**0.83%**	0	**347**	0	10.21

The TDNet-based method ($TDNet_{(DAT)}$) had the highest MOTA and MOTP compared to $TDNet_{(NO\ DAT)}$ and MDP. In terms of tracking speed, $TDNet_{(NO\ DAT)}$ ranked first, but it was found that it can also maintain it within ten frames.

Figure 14 shows the tracking visualization of the $TDNet_{(DAT)}$-based SC_MT tracking method when used on Video 3 and Video 4. There were nineteen targets in Video 3, which contained three moving targets and sixteen stationary targets. $TDNet_{(DAT)}$ detected a total of eighteen targets, with three moving and fifteen stationary targets. There were six targets (five of which were detected and one that was not) in Video 4, with three moving targets and three stationary targets. $TDNet_{(DAT)}$ detected a total of six targets, with three moving and three stationary targets. Table 3 shows the state statistics for $TDNet_{(DAT)}$ when used on Video 3 and Video 4. By comparing $TDNet_{(DAT)}$ with $TDNet_{NO\ DAT}$, it could be found that the introduction of DAT improves the tracking performance. This experiment proved the effectiveness of TDNet in tracking multiple targets.



(**a**) (**b**)

Figure 14. Visualisation of the TDNet-based SC_MT tracking results of Video 3 (**a**) and Video 4 (**b**).

4.4.2. MC_MT Tracking

To prove the superiority of the TDNet-based MC_MT tracking performance, we compared our methods with the MDP on Video 5.

Table 5 shows the SC_MT tracking performance comparison results. The TDNet-based method (TDNet$_{(DAT)}$) had the highest MOTA and MOTP when compared to the other methods. In terms of tracking speed, TDNet$_{(NO\ DAT)}$ ranked first. Although a category was added, the performance was not found to be lower than that of the SC_MT tracking approach. Figure 15 shows a visualization of the TDNet$_{(DAT)}$-based MC_MT tracking method's tracking performance on Video 5, as well as a visualization of the TDNet$_{(DAT)}$-based NTR tracking method's tracking performance on Video 6. There were three targets in Video 5, which contained two moving ships and one moving plane. As shown in Figure 15a, TDNet$_{(DAT)}$ detected a total of three targets, with two moving ships and one moving plane. Table 3 shows the state statistics for TDNet$_{(DAT)}$ on Video 5. Although the resolution of the data sets was relatively low, our method could still detect and track all the targets in Video 5.

(**a**) (**b**) (**c**)

Figure 15. Visualisation of the tracking results of the TDNet$_{(DAT)}$-based MC_MT tracking method on Video 5, as well as a visualisation of tracking results of the TDNet$_{(DAT)}$-based NTR tracking method on Video 6. (**a**) Video 5. (**b**,**c**) Video 6.

Table 5. Multi-category and multi-target tracking comparisons.

MC_MT Tracking on Video 5								
Methods	**MOTA**	**MOTP**	**MT**	**ML**	**FP**	**FN**	**IDS**	**Hz**
MDP [34]	80.21%	79.31%	100%	0%	0	0	2	8.52
CenterTrack [10]	83.50%	81.25%	100%	0%	0	0	0	21.80
TDNet$_{(NO\ DAT)}$	84.62%	83.51%	100%	0%	0	0	0	**11.36**
TDNet$_{(DAT)}$	**89.31%**	**85.62%**	100%	0%	0	0	0	10.21

4.4.3. NTR Tracking

TDNet can not only achieve SC_ST tracking, SC_MT tracking, and MC_MT tracking, but it can also recognise and monitor targets that newly appear in a scene. Moreover, we also compared our methods with the MDP on Video 6.

Initially, there are no targets in Video 6. The first target appears in the tenth frame, and the second target appears in the twenty-second frame. Both targets are moving. As shown in Figure 15, TDNet$_{(DAT)}$ recognised and tracked the two targets in turn.

Table 6 shows the NTR tracking performance comparison results. As with the NTR tracking approach, the TDNet$_{(DAT)}$-based method obtained the highest MOTA and MOTP values when compared to other methods. In terms of tracking speed, TDNet$_{(NO\ DAT)}$ ranked first and it was also within ten frames. Although a new target emerged, the performance was not harmed. In addition, the experiment verified the effectiveness of NTR and DAT in TDNet.

Table 6. NTR tracking comparisons.

NTR Tracking on Video 6								
Methods	**MOTA**	**MOTP**	**MT**	**ML**	**FP**	**FN**	**IDS**	**Hz**
MDP [34]	93.95%	89.38%	100%	0%	0	0	0	15.52
CenterTrack [10]	94.30%	91.03%	100%	0%	0	0	0	21.80
TDNet$_{(NO\ DAT)}$	95.97%	92.04%	100%	0%	0	0	0	**76.26**
TDNet$_{(DAT)}$	**97.98%**	**93.10%**	100%	0%	0	0	0	72.27

5. Concluding Remarks

This paper presented a tracking pipeline called TDNet for detecting and tracking ships and planes in optical remote sensing videos. As it is a newly emergent technology in the field of remote sensing data, remote sensing videos have only been researched by a few scholars, yet the initial results are promising. The tracking of ships and planes is of great significance in military applications. The proposed TDNet is composed of four modules: target recognition via R-FCN, multi-target tracking via the proposed MT-KCF, detecting-assisted tracking (DAT), and new target recognition (NTR). The ships and planes were first recognised in the first frame of the video, and they were then tracked via MT-KCF in the following frames. DAT was used to improve the tracking performance by recovering the tracking-lost targets and preventing the tracking lag occurrence. For newly emerged targets, NTR was used to recognise them and to track them during the tracking process. The quantitative comparison results on the six- and two-category optical remote sensing data sets demonstrated a huge performance gain when using the proposed method. The experimental results demonstrate the effectiveness and the state-of-the-art performance of the proposed method. The proposed method can achieve more than 90% performance in terms of precision on single-target tracking tasks and more than 85% performance when using MOTA on multi-object tracking tasks. However, as we know, our newly added DAT will introduce an additional computational cost. Hence, in our future work, we will focus on the TDNet model and look to integrate learning methods that can reduce the computational burden without compromising the performance.

Author Contributions: Z.S.: methodology, software, writing—original draft; G.W., W.Z., N.G. and Y.W.: supervision; J.L., D.C., Y.J. and Z.W.: validation and investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant numbers: 62302219 and 62276133), Natural Science Foundation of Jiangsu Province (Grant number: BK20220948), Internal Parenting Program (Grant number: 145AXL250004000X), and Research on Autonomous Navigation Strategy and Key Technologies of Earth Moon Space Spacecraft (Grant number:SKLGIE2022-ZZ2-08).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are unavailable due to privacy concerns.

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

References

1. Yuan, Y.; He, G.; Jiang, W.; Wang, G. Application of earth observation system of video satellite. *Remote Sens. Land Resours* **2018**, *30*, 1–8.
2. Du, Y.; Song, Y.; Yang, B.; Zhao, Y. StrongSORT: Make DeepSORT Great Again. *IEEE Trans. Multimed.* **2022**, *25*, 8725–8737. [CrossRef]
3. He, Q.; Sun, X.; Yan, Z.; Li, B.; Fu, K. Multi-Object Tracking in Satellite Videos with Graph-Based Multitask Modeling. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5619513. [CrossRef]
4. Zhang, J.; Zhang, X.; Huang, Z.; Cheng, X.; Feng, J.; Jiao, L. Bidirectional Multiple Object Tracking Based on Trajectory Criteria in Satellite Videos. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5603714. [CrossRef]
5. Xiao, J.; Cheng, H.; Sawhney, H.S.; Han, F. Vehicle detection and tracking in wide field-of-view aerial video. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 679–684.
6. Prokaj, J.; Medioni, G. Persistent Tracking for Wide Area Aerial Surveillance. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
7. LaLonde, R.; Zhang, D.; Shah, M. ClusterNet: Detecting Small Objects in Large Scenes by Exploiting Spatio-Temporal Information. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 23 June 2018; pp. 4003–4012.
8. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
9. Han, J.; Zhang, D.; Cheng, G.; Guo, L.; Ren, J. Object Detection in Optical Remote Sensing Images Based on Weakly Supervised Learning and High-Level Feature Learning. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3325–3337. [CrossRef]
10. Zhou, X.; Koltun, V.; Krähenbühl, P. Tracking Objects as Points. In Proceedings of the Computer Vision- ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part IV; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12349, pp. 474–490.
11. Lukezic, A.; Vojír, T.; Cehovin, L.; Matas, J.; Kristan, M. Discriminative Correlation Filter with Channel and Spatial Reliability. *arXiv* **2016**, arXiv:1611.08461.
12. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939. [CrossRef]
13. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
14. Babenko, B.; Yang, M.H.; Belongie, S. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632. [CrossRef]
15. Rui, C.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the ECCV 2012-12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 702–715.
16. Henriques, J.F.; Rui, C.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef] [PubMed]
17. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
18. Jiao, L.; Liu, F. Wishart Deep Stacking Network for Fast POLSAR Image Classification. *IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc.* **2016**, *25*, 3273–3286. [CrossRef]
19. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the ECCV 2016-14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 472–488.
20. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
21. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features. *Comput. Vis. Image Underst.* **2008**, *110*, 404–417. [CrossRef]
22. Lindeberg, T. Scale invariant feature transform. *Scholarpedia* **2012**, *7*, 2012–2021. [CrossRef]
23. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recigniztion. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

25. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the NIPS, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
26. Choi, W. Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor. In Proceedings of the ICCV, Santiago, Chile, 7–13 December 2015; pp. 3029–3037.
27. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
28. Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. *arXiv* **2016**, arXiv:1612.08242.
29. Cheng, G.; 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. [CrossRef]
30. Bernardin, K.; Stiefelhagen, R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *Eurasip J. Image Video Process.* **2008**, 246309. . [CrossRef]
31. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. In Proceedings of the ECCV, Amsterdam, The Netherlands, 11–14 October 2016; pp. 17–35.
32. Li, Y.; Huang, C.; Nevatia, R. Learning to associate: HybridBoosted multi-target tracker for crowded scene. In Proceedings of the CVPR, Miami, FL, USA; 20–25 June 2009; pp. 2953–2960.
33. Mueller, M.; Smith, N.; Ghanem, B. Context-Aware Correlation Filter Tracking. In Proceedings of the CVPR, Honolulu, HI, USA, 26 July 2017; pp. 1387–1395.
34. Xiang, Y.; Alahi, A.; Savarese, S. Learning to Track: Online Multi-object Tracking by Decision Making. In Proceedings of the ICCV, Santiago, Chile, 7–13 December 2015; pp. 4705–4713.

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.