remote sensing

MDPI

Article

# Robust Object Tracking Algorithm for Autonomous Vehicles in Complex Scenes

**Jingwei Cao [1,2], Chuanxue Song [1], Shixin Song [3,*], Feng Xiao [1], Xu Zhang [1], Zhiyang Liu [2] and Marcelo H. Ang, Jr. [2]**

[1] State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130025, China; caojw18@mails.jlu.edu.cn (J.C.); scx@jlu.edu.cn (C.S.); xiaofengjl@jlu.edu.cn (F.X.); xuz19@mails.jlu.edu.cn (X.Z.)
[2] Advanced Robotics Centre, Department of Mechanical Engineering, National University of Singapore, Singapore 117608, Singapore; e0452836@u.nus.edu (Z.L.); mpeangh@nus.edu.sg (M.H.A.J.)
[3] School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130025, China
* Correspondence: ssx@jlu.edu.cn

**Abstract:** Object tracking is an essential aspect of environmental perception technology for autonomous vehicles. The existing object tracking algorithms can only be applied well to simple scenes. When the scenes become complex, the algorithms have poor tracking performance and insufficient robustness, and the problems of tracking drift and object loss are prone to occur. Therefore, a robust object tracking algorithm for autonomous vehicles in complex scenes is proposed. Firstly, we study the Siam-FC network and related algorithms, and analyze the problems that need to be addressed in object tracking. Secondly, the construction of a double-template Siamese network model based on multi-feature fusion is described, as is the use of the improved MobileNet V2 as the feature extraction backbone network, and the attention mechanism and template online update mechanism are introduced. Finally, relevant experiments were carried out based on public datasets and actual driving videos, with the aim of fully testing the tracking performance of the proposed algorithm on different objects in a variety of complex scenes. The results showed that, compared with other algorithms, the proposed algorithm had high tracking accuracy and speed, demonstrated stronger robustness and anti-interference abilities, and could still accurately track the object in real time without the introduction of complex structures. This algorithm can be effectively applied in intelligent vehicle driving assistance, and it will help to promote the further development and improvement of computer vision technology in the field of environmental perception.

**Keywords:** environmental perception; autonomous vehicles; deep learning; Siamese network; object tracking

check for updates

**Citation:** Cao, J.; Song, C.; Song, S.; Xiao, F.; Zhang, X.; Liu, Z.; Ang, M.H., Jr. Robust Object Tracking Algorithm for Autonomous Vehicles in Complex Scenes. *Remote Sens.* **2021**, *13*, 3234. https://doi.org/10.3390/rs13163234

Academic Editor: Andrzej Stateczny

Received: 11 July 2021
Accepted: 12 August 2021
Published: 14 August 2021

## 1. Introduction

With the advancement of computer science and electronic information technology, artificial intelligence as represented by computer vision has developed rapidly. Object tracking is an essential aspect of computer vision, and it is widely used in fields such as autonomous driving, human–computer interaction, intelligent transportation, and video surveillance. Object tracking refers to automatically predicting the state of an object in subsequent frames from a given video image sequence according to the feature and position information of the object in the initial frame [1–3]. For intelligent vehicle driving assistance, an important task is to accurately and efficiently identify and track pedestrians and vehicles in the surrounding environment, which can help to greatly reduce incidences of traffic accidents and fully protect people's lives and property [4,5].

Considering that object tracking technology has important application value and practical significance, experts and scholars from various countries have conducted in-depth research on it and obtained effective findings. According to the principles of different algorithms, object tracking algorithms can generally be divided into generative tracking and discriminative tracking methods. Generative tracking methods are relatively traditional research methods. First, a model is established based on the apparent information of the object and then the most similar region is found as the object by the optimization algorithm in the region of interest, and the reliability of the similar region is guaranteed by constantly updating the model [6–8]. Ross et al. [9] proposed a robust incremental learning algorithm for object tracking. The linear space representation of principal component analysis was used to model the appearance of the object, and the model was updated based on incremental learning, which effectively improved the tracking performance. Arróspide et al. [10] proposed an object tracking method based on variable-dimension particle filtering. This method introduced a dynamically changing multidimensional space and solved the problem of the entry and disappearance of objects in the tracking process. Du et al. [11] proposed a MeanShift algorithm with adaptive block color histogram, which processed both the color statistics and spatial information of the object and solved the problem of inaccurate tracking in the original algorithm.

With the rapid development of machine learning and pattern recognition technology, discriminative tracking methods have become the focus of research in object tracking. Discriminative tracking methods usually convert object tracking into binary classification and use one or more classifiers to separate the object from the surrounding background as much as possible [12–14]. Discriminative tracking methods are mainly classified into categories based on correlation filtering and deep learning. Tracking methods based on correlation filtering adopt the signal similarity measurement in signal processing and find the image block with the highest similarity to the model in each frame as the tracking result. Danelljan et al. [15] proposed the discriminative scale space tracker for the scale change of objects. Two independent correlation filters were used to predict and estimate the position and size of the object, which showed good robustness. However, the additional correlation filter increased the computational burden and slowed down the tracking speed significantly. Henriques et al. [16] proposed a kernel correlation filter for high-speed tracking. The least squares classifier was used to optimize the mean square error between the signals, which accelerated the algorithm's computation speed while maintaining the original complexity. Liu et al. [17] used a multi-correlation filter to independently track multiple object blocks. The tracking performance of occluded objects was effectively improved through adaptive weighting, updating, and structural masking.

Due to the success of convolutional neural network (CNN) in image classification, tracking methods based on deep learning have gradually become a research hotspot in recent years. Wang et al. [18] proposed a visual tracking algorithm based on fully convolutional neural network (FCNN). This algorithm combined feature selection network with heat-map prediction network to study CNN features at different levels, which effectively alleviated the problem of object shifting. Nam et al. [19] proposed a multi-domain CNN for object tracking. The network was composed of shared layers and multiple branches of domain-specific layers and achieved relatively high tracking accuracy through online tracking. However, the real-time performance of the algorithm was poor because of the large number of candidate bounding boxes. Song et al. [20] used an adversarial learning algorithm for object tracking. The input features were discarded by randomly generating masks through the generating network, with the aim of adapting to various appearance changes of the object.

By summarizing and analyzing the existing methods, it can be seen that the main factors affecting the tracking performance of the algorithm include the object's own factors and environmental factors. Fast movement, appearance changes, scale changes, and rigid and non-rigid features are typical object factors. Lighting changes, background interference, image noise, and occlusion of irrelevant objects are common environmental factors [21–26]. Generative tracking methods normally involve complex computation and poor adaptability to the environment. When the object is in a cluttered background, tracking loss is likely to occur [27–29]. Compared with generative tracking methods, discriminative tracking methods can better deal with complex problems in practical applications, and the tracking accuracy is higher. Nevertheless, this type of algorithm involves a large amount of calculation and slow operation speeds, which makes it difficult to meet the real-time tracking requirements of intelligent vehicles [30–33]. In general, the current object tracking algorithms can only be effectively applied to simple scenes. When the scenes become complex, the algorithms have poor tracking performance and insufficient robustness and cannot obtain high precision and real-time functionalities simultaneously [34–36]. Therefore, in view of the problems and shortcomings of existing algorithms, there is an urgent need to develop a robust object tracking algorithm that can be effectively applied to complex scenes, so as to improve the technical level of intelligent vehicle driving assistance.

Addressing the gaps of current methods, this study proposes a robust object tracking algorithm for autonomous vehicles in complex scenes. The contribution of this work can be summarized in the following three items. Firstly, the Siam-FC network and related algorithms are studied, and the problems that need to be addressed in object tracking are analyzed. Secondly, we discuss the construction of a double-template Siamese network model based on multi-feature fusion with the improved MobileNet V2 used as the feature extraction backbone network, and the attention mechanism and template online update mechanism are introduced. Thirdly, we describe the relevant experiments that we carried out based on the public datasets and actual driving videos, and the tracking performance of the proposed algorithm is evaluated by using the methods of qualitative and quantitative analysis and by conducting a comprehensive comparison with state-of-the-art object tracking algorithms.

The rest of this article is organized as follows. In Section 2, the Siam-FC network and related algorithms are briefly introduced and analyzed. In Section 3, a double-template Siamese network model based on multi-feature fusion is established. In Section 4, tracking experiments are described for the proposed algorithm based on public datasets and actual driving videos. Finally, the conclusion is presented in Section 5.

## 2. Related Work

### 2.1. Siam-FC Network

The Siamese network is a supervised learning network framework for metric learning that uses two neural networks with shared weights to compare the similarity of two features through similarity measurement calculations. Different from the traditional object tracking methods, this network structure transforms the object tracking problem into the similarity learning problem, which can not only run in real time on the GPU but can also catch up with or even exceed the related filtering methods combined with depth features in terms of accuracy. Considering that there are few training datasets available in the object tracking field, the Siamese network structure can naturally increase the amount of training data and expand the limited datasets by inputting a pair of images each time, so as to achieve the goal of fully training the network [37]. The Siam-FC network is a classical network model that adopts the Siamese network framework for object tracking and was first proposed by Bertinetto in 2016 [38]. The network structure of the Siam-FC network is shown in Figure 1.

The Siam-FC network adopts a double-branch structure; one branch is a template branch and the other is a search branch. The network first performs feature extraction on the object template and search area, conducts similarity measurement calculations, and uses the similarity score response graph to determine the location of the tracking object in the search area. The Siam-FC network transforms object tracking into a template matching process and takes the full convolutional layer as the similarity measure function. The entire process can be expressed as

$$f(z, x) = \varphi(z) * \varphi(x) + b \cdot \mathrm{I}, \tag{1}$$

where $z$ and $x$ represent the template image and search image, respectively; $\varphi(z)$ and $\varphi(x)$ separately represent the feature maps of the template image and search image; $*$ is the inter-correlation operation; $b$ is the bias term; I is the identity matrix; and $f(z, x)$ is the similarity score response of the two images.

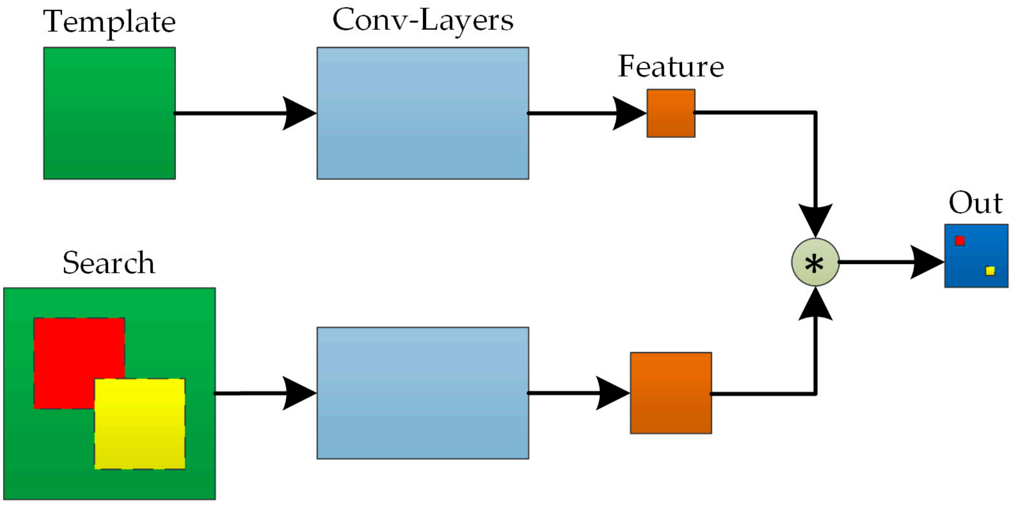


**Figure 1.** The network structure of the Siam-FC network.

Due to the fully connected network, the Siam-FC network can input images of different sizes. It does not require online learning, which greatly reduces the time for the model update, and thus it has good real-time performance. However, because the network always takes the object in the first frame as the template, the tracking stability is significantly worsened when the tracking object changes in appearance. Furthermore, the feature extraction capability of the convolutional layer of the network is insufficient, and its tracking accuracy needs to be improved.

### *2.2. Algorithms Related to Siam-FC*

As the Siam-FC network has better comprehensive performance, many related algorithms have emerged. Compared with the traditional correlation filtering methods, the CNN features extracted by Siam-FC-related algorithms are more suitable for tracking data and the entire model, the network expression ability is stronger, and competitive results have been achieved on the existing tracking datasets. A comparison of some Siam-FC-related algorithms is shown in Table 1.

**Table 1.** A comparison of some Siam-FC-related algorithms.

| | **CFNet** | **DSiam** | **SA-Siam** | **SiamRPN** |
|---|---|---|---|---|
| Release date | July 2017 | October 2017 | June 2018 | June 2018 |
| Developers | Valmadre et al. [39] | Guo et al. [40] | He et al. [41] | Li et al. [42] |
| Training datasets | ILSVRC 2015-VID dataset | ILSVRC 2015-VID dataset | Color images in the ILSVRC 2015-VID dataset | ILSVRC 2015-VID dataset and YouTube-BoundingBoxes dataset |
| Outstanding feature | The correlation filtering operation is integrated into a single network layer and embedded into the network | Addition of the object appearance change conversion layer and background suppression conversion layer in the x and z branches, respectively | Use of two branch networks to obtain semantic features and appearance features, respectively | Application of the RPN module to the tracking task and transformation of the similarity measurement calculation into classification and regression |

Valmadre et al. proposed the CFNet algorithm, based on the Siam-FC network, in which a correlation filter was added to one of the branch structures. The embedded correlation filter could be interpreted as a differentiable deep neural network and can backpropagate the loss to the convolutional layer. Therefore, CFNet combines the Siamese network using offline training with the correlation filter for online learning, maximizing their respective advantages so that it can perform end-to-end training. Guo et al. proposed DSiam using a fast conversion learning model. The model can better adapt to the appearance changes of the object and the background conversion, and the tracking performance is further improved.

He et al. proposed SA-Siam, based on the Siam-FC network, which adopts two heterogeneous Siamese networks to extract different features. One branch structure is used to extract semantic features in image classification and the other is used to extract appearance features in similarity matching. In addition, a channel attention module is added to the semantic branch structure. The tracking performance of the network is effectively improved.

Li et al. added a region proposal network (RPN) on the basis of the Siam-FC network and then proposed SiamRPN. The first half of the network is a Siamese network structure that is used to extract the totality of the semantic features. The second half of the network is an RPN composed of classification and regression branches. The classification branch is used to distinguish the tracking object from irrelevant background, and the regression branch is used to generate the bounding boxes that match the actual object size. SiamRPN transforms the similarity measurement calculation into classification and regression, which markedly improves the tracking accuracy and speeds up the search for multi-scale objects.

### *2.3. Limitations of Existing Algorithms*

So far, the existing Siam-FC-related algorithms have made considerable progress compared with the original network, but some urgent problems remain. The limitations of the existing algorithms can be summarized as follows:

(1) The adopted feature extraction backbone network has insufficient ability to extract deep features, cannot identify the exact location of the tracking object in the search area, and has poor tracking accuracy;

(2) The network model only performs object recognition based on deep features, ignoring the detailed information of shallow features. When the scale of the tracking object changes greatly, the object can be lost easily;

(3) The balance of tracking speed and tracking accuracy cannot be guaranteed. There are too many training parameters and redundant features are prone to occur, and the tracking efficiency is relatively low.

## 3. Network Model

### *3.1. Double-Template Siamese Network Model Based on Multi-Feature Fusion*

Traditional Siamese network models usually perform offline training on large-scale samples first, conduct similarity matching in the search image based on the template in the first frame, and finally achieve the goal of object tracking. However, in the tracking process, the appearance and scale of the object change greatly over time. If the network model cannot be updated online in time, then the tracking errors generated will gradually accumulate, making the tracking effect worse and worse. If the template is updated at regular intervals with the latest tracking results, then the desired tracking effect cannot be achieved because the template itself is not accurate, and the feature representation error of the object becomes larger over time. Therefore, reasonable adjustments to the structure of the Siamese network are needed in response to the aforementioned problems.

The Siam-FC network uses the unfilled AlexNet as the basic backbone network. Due to the shallow depth of the AlexNet network, the feature extraction ability for deep features of the object is insufficient. When the tracking object moves rapidly, motion blur is easily produced, and the input video frame is low-resolution and blurry. The network model cannot extract the deep semantic features of the object, resulting in a significant decline in the discriminative ability of the tracker. Consequently, a deeper basic backbone network is needed to extract deep features.

Aimed at the problems of poor tracking performance and insufficient robustness in Siam-FC-related algorithms in complicated scenarios, this study makes a considerable improvement on the basis of the existing network and establishes a double-template Siamese network model based on multi-feature fusion for object tracking. Figure 2 shows the building blocks of the double-template Siamese network model based on multi-feature fusion. Figure 3 presents the basic structure of the double-template Siamese network model based on multi-feature fusion.

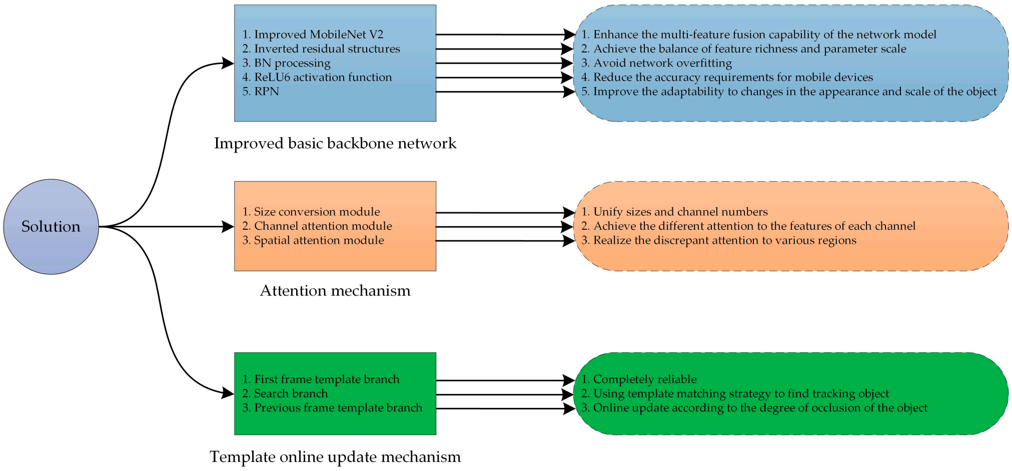

Solution
1. Improved MobileNet V2
2. Inverted residual structures
3. BN processing
4. ReLU6 activation function
5. RPN
Improved basic backbone network
1. Enhance the multi-feature fusion capability of the network model
2. Achieve the balance of feature richness and parameter scale
3. Avoid network overfitting
4. Reduce the accuracy requirements for mobile devices
5. Improve the adaptability to changes in the appearance and scale of the object
1. Size conversion module
2. Channel attention module
3. Spatial attention module
Attention mechanism
1. Unify sizes and channel numbers
2. Achieve the different attention to the features of each channel
3. Realize the discrepant attention to various regions
1. First frame template branch
2. Search branch
3. Previous frame template branch
Template online update mechanism
1. Completely reliable
2. Using template matching strategy to find tracking object
3. Online update according to the degree of occlusion of the object

**Figure 2.** The building blocks of the double-template Siamese network model based on multi-feature fusion.

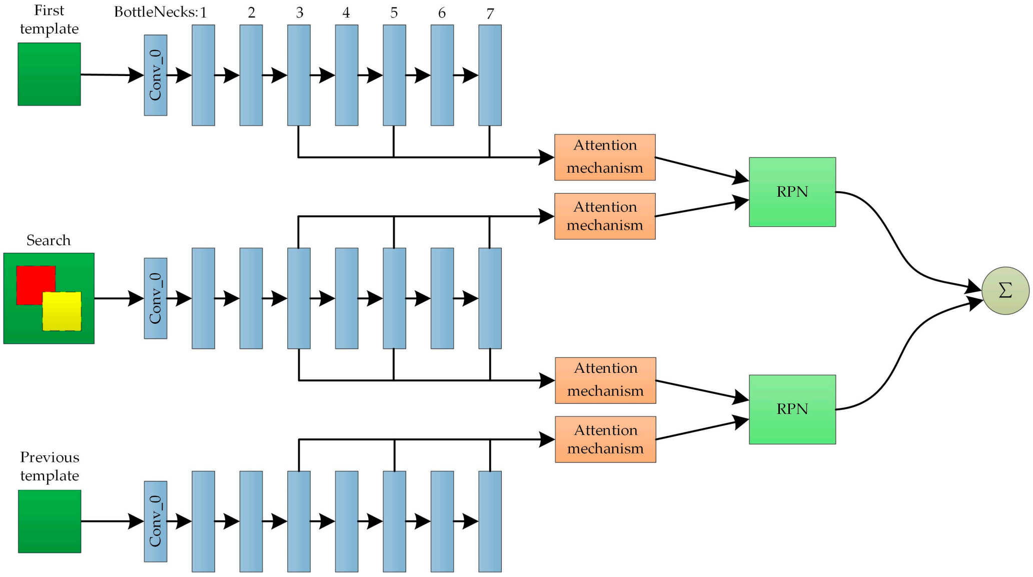


**Figure 3.** The basic structure of the double-template Siamese network model based on multi-feature fusion.

As shown in Figure 3, the proposed network model adopts a three-branch Siamese network structure. In addition to the template branch based on the first frame and search branch, a template branch based on the previous frame is added. The advantage of this structure is that it overcomes the disadvantage of utilizing only the first frame template or the latest template, and it can still track the latest object in real time without introducing complex structures. The feature extraction network is redesigned and the improved MobileNet V2 is used as the basic backbone network. MobileNet V2 is a lightweight CNN released by Google [43]. Its original model is composed of a conventional convolutional layer, seven BottleNecks, and a pooling layer. Compared with AlexNet, MobileNet V2 has a deeper network, smaller volume, and higher accuracy and involves less computation. To realize multi-feature fusion and reduce irrelevant calculations, MobileNet V2 is optimized properly and two convolutional layers of the seventh BottleNeck and pooling layer are removed. Furthermore, to fully utilize the deep and shallow features, multiple attention mechanisms are introduced after BottleNecks 3, 5, and 7 to achieve a significant expression of the object features. The attention-adjusted feature maps are input into the region proposal network branches, and a weighted summation on the classification and regression results of the two template branches is performed to achieve the accurate object tracking. The double-template Siamese network model based on multi-feature fusion improves and optimizes the traditional template structure of the Siamese network, which is conducive to achieving a balance of feature richness and parameter scale.

In MobileNet V2, each BottleNeck generally contains one or several inverted residual structures, and the unit of the inverted residual structure is shown in Figure 4. Each inverted residual structure usually consists of 1 × 1 point-to-point convolutional layers, 3 × 3 deep separable convolutional layers, and 1 × 1 point-to-point linear convolutional layers. Compared with the traditional residual structure, the dimensions of the inverted residual structure are first expanded and then compressed to increase the number of channels and obtain more feature information. Different from the traditional convolutional layer, the combined use of point-to-point convolutional layers and deep separable convolutional layers greatly enriches the training features, avoids the destruction of the original features, and helps to reduce the number of parameter calculations and improve the efficiency of convolution operations. In addition, batch normalization processing is added after each convolutional layer, and the ReLU6 activation function is added after the first point-to-point convolutional layer and deep separable convolutional layer. The expression of the ReLU6 activation function is as follows:

$$\mathrm{ReLU6}(x) = \begin{cases} 0, & \text{if} \quad x < 0 \\ x, & \text{if} \quad 0 \le x \le 6 \\ 6, & \text{if} \quad x > 6 \end{cases} . \tag{2}$$

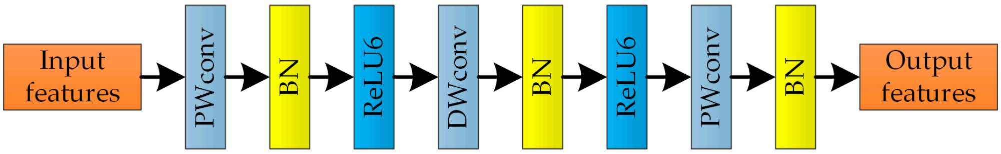


**Figure 4.** The unit of the inverted residual structure.

As the name suggests, the role of the ReLU6 activation function is to limit the maximum output to 6. This condition aims to avoid unnecessary numerical loss resulting from the accuracy limitation of the mobile device so that it can still have good numerical resolution at low accuracy. Through repeated tests, the best experimental results can be obtained by setting the maximum output to 6. Furthermore, to avoid the information loss of elements with values less than 0 after feature extraction, the ReLU6 activation function is no longer added after the last point-to-point convolutional layer.

### 3.2. *Attention Mechanism*

In the Siam-FC network, the image features used are the deep features of the last layer of the feature extraction network, while the shallow features are ignored. The properties of deep and shallow features are completely different. Deep features contain rich semantic feature information but the resolution is insufficient; shallow features contain enough detailed feature information but the semantic features are deficient. When the appearance and scale of the object change greatly in the cluttered background, tracking loss can easily occur; the introduction of attention mechanism is an effective measure to solve this problem. In other words, attention mechanism is a kind of signal processing mechanism. First, a quick scan of the global information is performed to obtain the salient objects or focus areas and then the attention weight of the global information is reasonably redistributed in order to focus on the key details and reduce the interference of irrelevant background information. The attention mechanism in this study is mainly realized through the size conversion, channel attention, and spatial attention modules.

### 3.2.1. Size Conversion Module

As the output features of BottleNecks 3, 5, and 7 are different in terms of size and channel number, a size conversion module is required to unify their sizes and channel numbers. For different sizes, this study adopts up-sampling operation to make the sizes consistent; for different numbers of channels, this study uses 1 × 1 convolution kernels to convert them to the same number of channels.

### 3.2.2. Channel Attention Module

The channel attention module mainly improves the weight of feature channels related to the tracking object and reduces the weight of feature channels unrelated to the tracking object in order to achieve the differential attention to the features of each channel. This approach is beneficial for the elimination of interference noise and redundant features and can improve the expression accuracy of key features. Figure 5 shows the structure of the channel attention module.

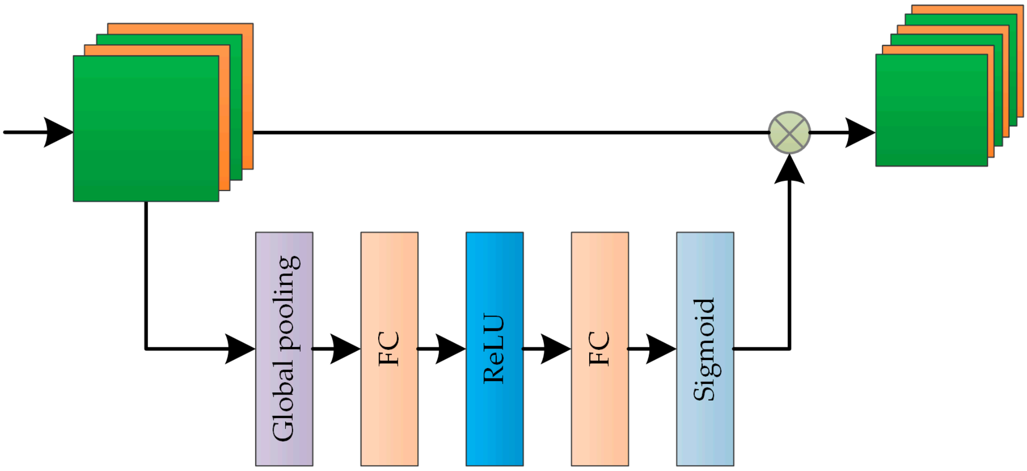


**Figure 5.** The structure of the channel attention module.

The structure of the channel attention module utilizes the SE-block in the SENet algorithm for image classification [44]. In this structure, we first define the channel aggregate of the input feature map:

$$A = [a_1, a_2, a_3, \ldots, a_n], \tag{3}$$

where $a_k \in R^{H \times W}, k = 1, 2, 3, \ldots, n$.

After global pooling, the feature vector obtained is

$$b = [b_1, b_2, b_3, \ldots, b_n], \tag{4}$$

where $b_k \in R^{H \times W}, k = 1, 2, 3, \ldots, n$.

After the feature vector $b$ passes through the first fully connected layer (FC), a ReLU activation function is added to obtain the nonlinear result. Then, the sigmoid function is added after passing through the second FC, and the resulting feature vector is

$$\alpha = [\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n], \tag{5}$$

where $\alpha_k \in R^{H\times W}, k = 1,2,3,\ldots,n$.

The feature vector $\alpha$ is superimposed on the original feature map $A$, and the feature channel is rescaled. At this time, the channel aggregate of the channel attention feature map is

$$\overline{A} = \alpha \cdot A = [\overline{a}_1, \overline{a}_2, \overline{a}_3, \ldots, \overline{a}_n], \tag{6}$$

where $\overline{a}_k \in R^{H\times W}, k = 1,2,3,\ldots,n$.

### 3.2.3. Spatial Attention Module

The spatial attention module primarily assigns different weights to various spatial positions on the feature map in order to realize the discrepant attention to various regions of the image, which is helpful in further strengthening the relevance of the features in the spatial position. Figure 6 presents the structure of the spatial attention module.

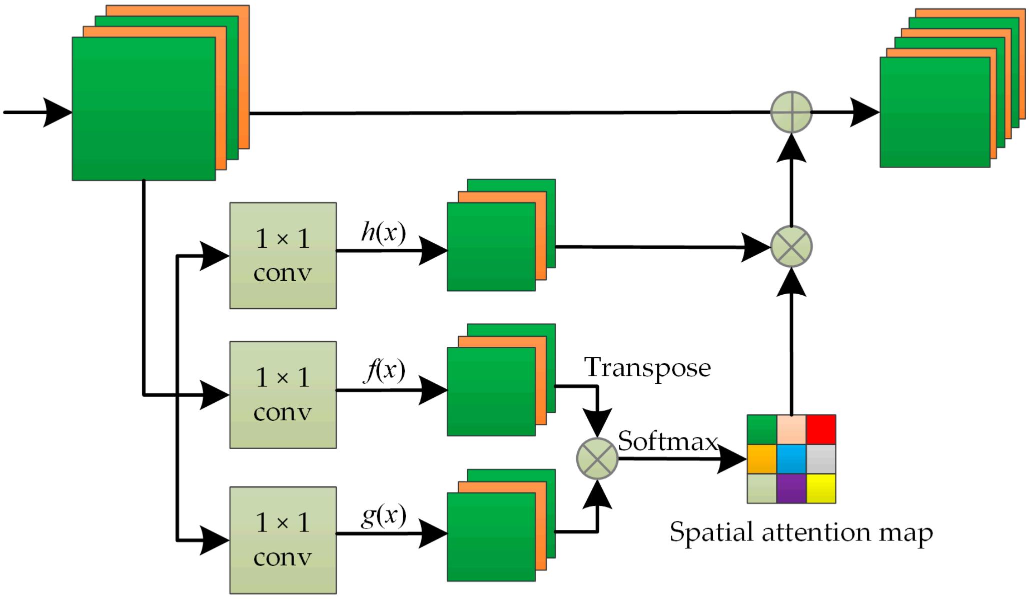


**Figure 6.** The structure of the spatial attention module.

The structure of the spatial attention module utilizes a non-local model for image recognition [45]. In this structure, the convolution kernels with the size of 1 × 1 are first used to convolve the input feature map, and three convolution results can be obtained. Then, three different conversion functions are used to convert the convolution results separately. The conversion functions $f(x)$, $g(x)$, and $h(x)$ are as follows:

$$\begin{array}{l} f(x) = W_1 \times x \\ g(x) = W_2 \times x \\ h(x) = W_3 \times x \end{array}, \tag{7}$$

where $W_1$ is the weight of function $f(x)$, $W_2$ is the weight of function $g(x)$, and $W_3$ is the weight of function $h(x)$.

The output result of function $f(x)$ is transposed, and then matrix multiplication is performed with the output result of function $g(x)$. After the calculation of the Softmax function, the spatial attention map can be obtained, and the calculation formula is

$$Y_{b,a} = \frac{e^{f(x_a)^T \times g(x_b)}}{\sum_{k=1}^{W\ H} e^{f(x_a)^T \times g(x_b)}}, \tag{8}$$

where $a$ represents the $a$-th position in the input image; $b$ represents the $b$-th position in the input image; $f(\cdot)$ is the output result of function $f(x)$; and $g(\cdot)$ is the output result of function $g(x)$.

The spatial attention map and output result of function $h(x)$ perform the matrix multiplication, and then the result is added to the original feature map $x$. The feature map adjusted by the spatial attention module can thus be obtained. The relevant calculation formula is

$$O_b = x_b + \beta \times (\textstyle\sum_{a=1}^{W\ H} Y_{b,a} \times h(x_a)), \tag{9}$$

where $\beta$ is the weight parameter and $h(\cdot)$ is the output result of function $h(x)$.

### 3.3. Template Online Update

As the features and states of the object in the two adjacent frames change minimally during the tracking process, the candidate object that is more similar to the object in the previous frame should be selected as the tracking object, rather than the selection being simply based on the highest response map. Therefore, a new scoring vector needs to be established according to the size, aspect ratio, and position of the object.

Based on the assumptions that the size of the tracking boxes of the previous frame is $(w, h)$ and the length and width vector group of candidate objects in the new frame is $\{w_i, h_i\}_{i=1}^{n}$, $i$ is the identifier of candidate objects and $n$ is the number of candidate objects. When the size, aspect ratio, and position of the object in two adjacent frames are different, the reliability of the tracking results is low. According to these three characteristics, the reliability weight is defined as follows:

$$O_b = x_b + \beta \times (\textstyle\sum_{a=1}^{W\ H} Y_{b,a} \times h(x_a)), \tag{10}$$

where $s = \max(\frac{\sqrt{w_i h_i}}{\sqrt{wh}}, \frac{\sqrt{wh}}{\sqrt{w_i h_i}})$, $r = \max(\frac{w/h}{w_i/h_i}, \frac{w_i/h_i}{w/h})$, and $p = \sqrt{\frac{(w_i-w)^2+(h_i-h)^2}{2L^2}}$(L is the side length of the search area).

In the object tracking process, the degree of occlusion can be divided into full occlusion, partial occlusion, and no occlusion. When the object is occluded, the influence of position is ignored, and the reliability weight is redefined as follows:

$$W_i = \exp(-(s \times r - 1)). \tag{11}$$

The tracking result score of each candidate object is

$$S_i = W_i score_i, \tag{12}$$

where $score_i$ is the initial score of the candidate object.

As the state of the object changes randomly, the tracking results of subsequent frames cannot be guaranteed to be completely reliable, except for the first frame. Therefore, the best tracking result score for a candidate object should not be regarded as the final tracking result of the frame but should be combined with the cumulative result of the previous frame as follows:

$$S_t = (1 - \lambda)S_{t-1} + \lambda S_t^*, \tag{13}$$

where $S_t^*$ is the best tracking result score of the candidate object and $\lambda$ is the update rate, which is 0 when the object is completely occluded.

In addition, when the object is under incomplete occlusion, to reduce the influence of interference and noise, the update rate is set to be proportional to the best tracking result score of the candidate objects as follows:

$$\lambda = \lambda_{init} S_t^*, \tag{14}$$

where $\lambda_{init}$ is the initial update rate, which is set to 0.4 in this study.

Before object tracking, the object image of the first frame is cropped, and it is set as the template of the first and previous frames. During the tracking process, the template of the first frame remains unchanged, and the template of the previous frame is updated online according to the degree of occlusion. When the best tracking result score of the candidate object is higher than the preset threshold, it indicates that the reliability of the tracking result is relatively high, the tracking object is not obviously occluded, and the features and state of the object have changed minimally; the template of the previous frame is then updated immediately. Otherwise, the template of the previous frame remains the same.

## 4. Experiments

### *4.1. Experimental Environment*

The software environment consists of an Ubuntu 16.04 64-bit operating system, with the PyTorch deep learning framework, CUDA 9.1, cuDNN 7.1, and Python 3.8.0.

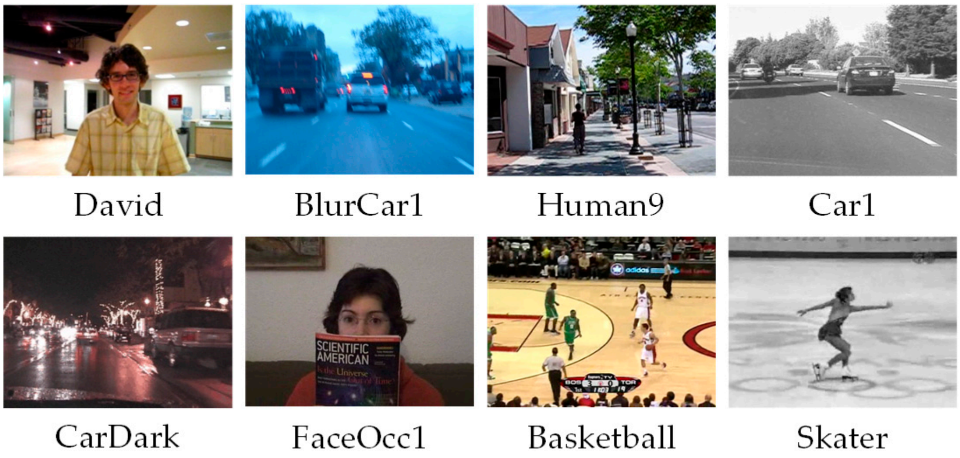
The hardware environment consists of an Intel Core i7-7700 CPU @ 3.60 GHz processor, with 32 GB memory and an NVIDIA GeForce GTX 1080Ti GPU, 11 GB.

### *4.2. Object Tracking Experiment Based on Public Datasets*

#### 4.2.1. Public Datasets

Different public datasets are respectively used as training and testing sets to obtain better experimental results. This study uses the YouTube-BoundingBoxes dataset as the training set, which is a large-scale dataset of video URLs released by Google in 2017 [46]. This dataset is composed of 380,000 15–20 s video clips and contains 23 categories including 5 million single-object bounding boxes with manual annotations. The internal images of the dataset are obtained through intensive sampling with high image quality, and the classification accuracy of the bounding boxes can reach 95%, which makes it an extremely effective training dataset for object detection and object tracking.

This study uses the OTB2015 dataset as the testing set, which was expanded by Wu from the OTB2013 dataset [47]. This dataset has become one of the benchmarks for evaluating object tracking algorithms. It consists of 100 fully annotated video sequences, a quarter of which use grayscale data. All the sequences cover 11 challenging aspects: occlusion (OCC), illumination variation (IV), scale variation (SV), motion blur (MB), background clutter (BC), deformation (DEF), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), and low resolution (LR). The OTB2015 dataset fully takes into account the various challenging aspects that may appear in complex scenes, which is conducive to the comprehensive testing of the robustness of object tracking algorithms in practical application scenarios. Representative images of some video sequences from the OTB2015 dataset are shown in Figure 7.



David BlurCar1 Human9 Car1

CarDark FaceOcc1 Basketball Skater

**Figure 7.** Representative images of some video sequences from the OTB2015 dataset.

### 4.2.2. Training Settings and Evaluation Indicators

The end-to-end collaborative training is conducted based on the YouTube-Bounding Boxes dataset. During training, the sizes of the template images of the first and previous frames are each set to 127 × 127, whilst that of search image is set to 255 × 255, and the size of the output response map is set to 17 × 17. When the tracking object is not obviously occluded, the anchor step of the RPN is always set to 8, and the aspect ratio includes 1/3, 1/2, 1, 2, and 3.

The stochastic gradient descent method is used for network optimization, the momentum coefficient is set to 0.9, and the weight attenuation value is set to 0.0001. The training process is divided into 50 rounds, and the learning rate decreases logarithmically from 0.03 to 0.0005 with the number of training rounds.

The object tracking algorithms were usually quantified based on the evaluation indicators to accurately evaluate their performance. The evaluation indicators used in this study mainly included the overlap rate (OR) and center location error (CLE).

The OR is the final overlap ratio of the predicted and real boxes; that is, the ratio of the intersection area of the predicted and real boxes to the union area of the two boxes. The maximum value is 1 and the minimum value is 0. This value is used to reflect the closeness between the tracking result and the real object, also known as the success rate, which can be expressed by the following formula:

$$OR = \frac{Area(R_T \cap R_G)}{Area(R_T \cup R_G)}, \tag{15}$$

where $R_T$ represents the region of the predicted box, $R_G$ represents the region of the real box, and $Area(\cdot)$ represents the number of pixels in the area.

The CLE refers to the center position error between the final predicted box and the real box; that is, the Euclidean distance between the center coordinates of the predicted and real boxes. This indicator is used to reflect the tracking accuracy of the algorithms and can be expressed by the following formula:

$$CLE = \sqrt{(x_T - x_G)^2 + (y_T - y_G)^2}, \tag{16}$$

where $(x_T, y_T)$ is the center coordinate of the predicted box and $(x_G, y_G)$ is the center coordinate of the real box.

### 4.2.3. Analysis and Discussion of Testing Results

To fully test the tracking effect of the proposed object tracking algorithm in complex scenes, six typical video sequences from the OTB2015 dataset are selected as testing sequences. The selected video sequences cover all challenging aspects in order to better simulate the possible interference factors that may appear in actual road scenes. The testing sequences and their challenging aspects are listed in Table 2. Furthermore, we select six state-of-the-art object tracking algorithms for experiments: CFNet, SA-Siam and SiamRPN, based on the Siamese network framework; and SRDCF [48], MCCT [49], and SACF [50], with the correlation filtering model. The above algorithms are consistent with the system operating environment of the proposed algorithm, and related tests are performed based on the same tracking dataset to comprehensively evaluate the tracking performance of the proposed network model.

1. Qualitative analysis and discussion

We observe the tracking results of the testing sequences in a variety of complex scenes, and analyze and discuss the performance of the algorithms in image description. Figure 8 shows the tracking results of different object tracking algorithms for the testing sequences.

**Table 2.** Testing sequences and their challenging aspects.

| Video Sequence | Challenging Aspects |
|---|---|
| CarDark | IV, BC |
| Human4 | IV, SV, OCC, DEF |
| BlurCar1 | MB, FM |
| MotorRolling | IV, SV, MB, FM, IPR, BC, LR |
| Suv | OCC, IPR, OV |
| Biker | SV, OCC, MB, FM, OPR, OV, LR |

The above figure shows the tracking performances of the seven object tracking algorithms, including the proposed algorithm, for different testing sequences, and the tracking results are represented by bounding boxes of different colors.

In the CarDark video sequence, the main challenging aspects are IV and BC. In the initial frames, all tracking algorithms can closely follow the tracking object. However, at the 289th frame, due to the dark light and the presence of an interfering vehicle with similar characteristics to the tracking vehicle in the background, CFNet exhibits tracking drift. At the 309th frame, SRDCF also shows tracking drift due to a similar interference vehicle. At the 393rd frame, all the tracking algorithms except for CFNet and SRDCF, which completely lost the object, perform well without losing the object.

In the Human4 video sequence, the main challenging aspects are DEF, OCC, SV, and IV. The scale of the tracking person in this video sequence is small, the features contained in the object are limited, and interference occurs from unrelated surrounding objects. However, the tracking is easy in general because of the simple background. At the 195th frame, only CFNet has a positioning error and the tracking box deviates from the object person. At the 255th frame, CFNet repositions accurately and moves with the tracking person, as the other tracking algorithms.

In the BlurCar1 video sequence, the main challenging aspects are MB and FM. In the initial frames without motion blur, all algorithms can track the object vehicle accurately. However, at the 256th frame, the video image begins to become blurred due to camera shaking. At the 257th frame, all the algorithms except for the proposed algorithm have different degrees of tracking drift. As the video image gradually recovers its clarity, all algorithms can track the object vehicle stably at the 516th frame, indicating that the algorithms have memory functions and can save the key features of the object vehicle. With the rapid movement of the vehicle, the background of the video image is further blurred. At the 768th frame, CFNet, SRDCF, and MCCT completely lose the tracking object, thereby resulting in tracking failure.

In the MotorRolling video sequence, the main challenging aspects are IPR, LR, BC, SV, IV, MB, and FM. At the 68th frame, the video image has low resolution and motion blur is produced due to the fast motion of the motorcycle. Except for the proposed algorithm and SiamRPN, all other algorithms show slight tracking drift. At the 76th frame, the video image is still blurred and the motorcycle rotates in the air under a complex and dim background. The tracking boxes of all the algorithms except for the proposed algorithm deviate from the tracking object.

In the Suv video sequence, the main challenging aspects are OCC, IPR, and OV. In the initial frame, all algorithms perform well in tracking the object vehicle. Starting from the 509th frame, the object vehicle begins to be partially obscured. At the 536th frame, the object vehicle is completely obscured by roadside trees. All the algorithms except for the proposed algorithm, SiamRPN, and SA-Siam show obvious tracking drift. At the 576th frame, the object vehicle is no longer occluded by the surrounding environment, but some tracking algorithms are not accurate enough for object positioning, indicating that long-term occlusion has a certain effect on the tracking performance of the algorithms.

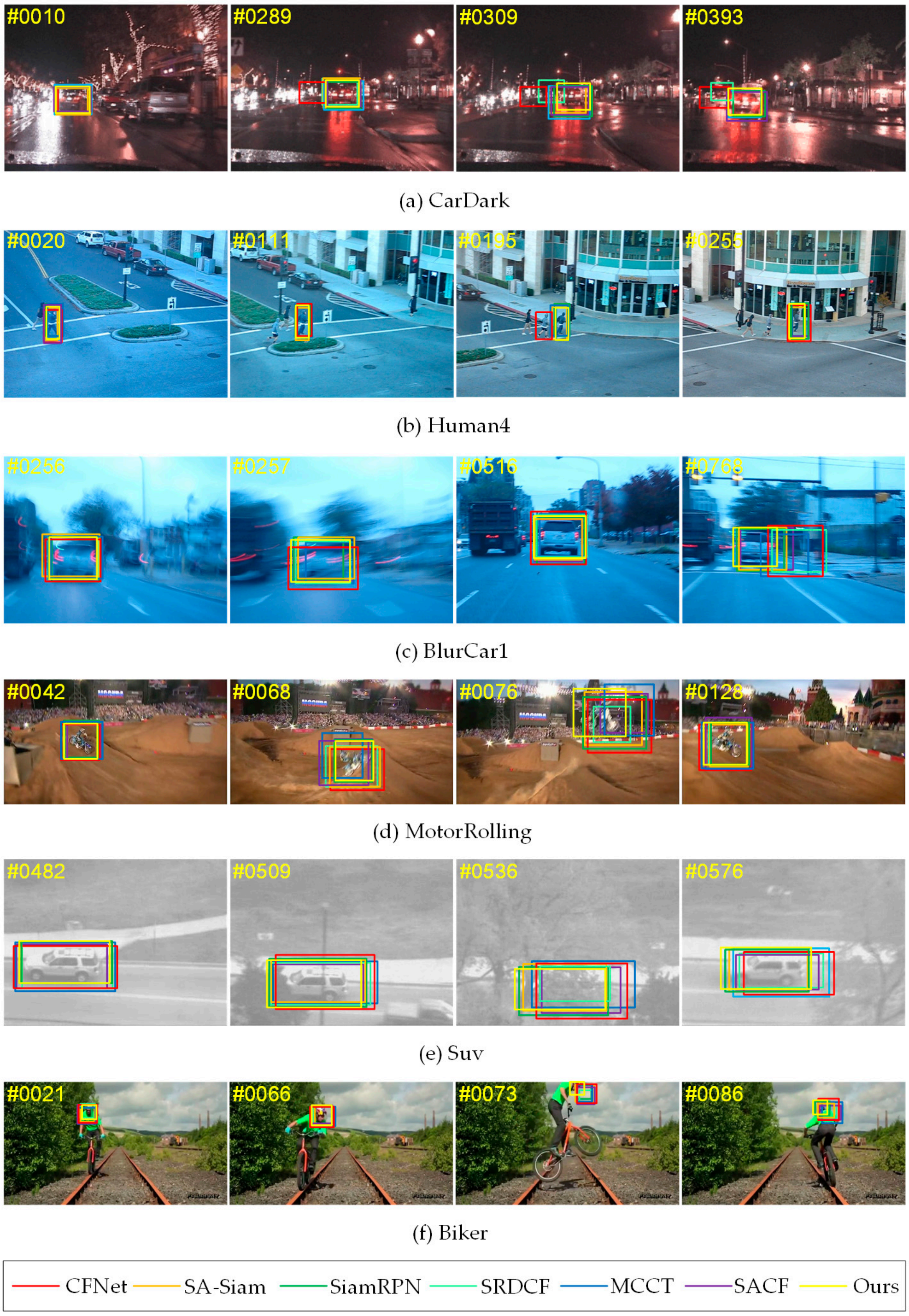


**Figure 8.** The tracking results of different object tracking algorithms for the testing sequences.

In the Biker video sequence, the main challenging aspects are OV, SV, OPR, MB, FM, LR, and OCC. At the 21st frame, the biker rides forward, and all algorithms can achieve accurate tracking of his face. At the 73rd frame, due to the out-of-plane rotation of the bicycle, the biker's face is sideways and beyond the edge of the image, and only the proposed algorithm can track it stably. At the 86th frame, the biker rides backward. Owing to the limited object features extracted, all the algorithms except for the proposed algorithm and SiamRPN have a small range of tracking drift.

From the qualitative analysis of the test results, it can be seen that other object tracking algorithms have limitations, and most of them can only be applied to a single simple scenario. When the environment changes dramatically or the appearance of the object changes significantly, it is easy for tracking drift to occur or even for the object to be lost. The proposed algorithm can still achieve effective tracking of different objects in complex scenes, and the bounding boxes are accurate and of the appropriate size. It shows good robustness with regard to the changes in appearance of the object and various degrees of occlusion, as well as excellent environmental adaptability with regard to the illumination variation and differing background.

2. Quantitative analysis and discussion

The tracking results of the different object tracking algorithms in the testing sequences are quantified, and the performances of the algorithms are analyzed and discussed in the form of numerical comparison. The average OR and average CLE of the various object tracking algorithms in the testing sequences are shown in Tables 3 and 4, respectively.

**Table 3.** The average OR (%) of the various object tracking algorithms in the testing sequences.

| Video Sequence | SRDCF | MCCT | SACF | CFNet | SA-Siam | SiamRPN | Ours |
|---|---|---|---|---|---|---|---|
| CarDark | 58.9 | 68.5 | 69.2 | 64.3 | 74.1 | 78.4 | 82.5 |
| Human4 | 79.5 | 80.6 | 87.1 | 68.4 | 82.5 | 84.6 | 85.8 |
| BlurCar1 | 58.2 | 59.0 | 65.4 | 55.8 | 72.3 | 76.5 | 80.6 |
| MotorRolling | 57.9 | 58.8 | 62.7 | 54.7 | 72.5 | 77.4 | 80.1 |
| Suv | 58.6 | 57.7 | 63.5 | 52.3 | 76.1 | 77.8 | 79.4 |
| Biker | 80.1 | 83.4 | 85.2 | 75.7 | 82.8 | 85.6 | 86.5 |
| Average | 65.5 | 68.0 | 72.2 | 61.9 | 76.7 | 80.1 | 82.5 |

**Table 4.** The average CLE (pixels) of the various object tracking algorithms in the testing sequences.

| Video Sequence | SRDCF | MCCT | SACF | CFNet | SA-Siam | SiamRPN | Ours |
|---|---|---|---|---|---|---|---|
| CarDark | 16.3 | 14.4 | 13.6 | 21.1 | 12.5 | 11.3 | 9.1 |
| Human4 | 6.4 | 6.2 | 5.5 | 10.6 | 6.0 | 5.9 | 5.7 |
| BlurCar1 | 21.7 | 19.5 | 16.6 | 23.8 | 14.5 | 12.8 | 10.4 |
| MotorRolling | 16.6 | 16.3 | 15.9 | 24.2 | 14.8 | 11.6 | 10.5 |
| Suv | 16.9 | 18.2 | 15.1 | 21.4 | 12.8 | 12.2 | 11.2 |
| Biker | 6.2 | 5.0 | 4.5 | 10.1 | 5.3 | 4.2 | 3.8 |
| Average | 14.0 | 13.3 | 11.9 | 18.5 | 11.0 | 9.7 | 8.5 |

From the quantitative analysis of testing results, it can be seen that, compared with other object tracking algorithms, the proposed algorithm shows outstanding performance in all testing sequences. The mean value of the average OR is the highest, reaching 82.5%, and the mean value of the average CLE is the lowest, only 8.5 pixels. The proposed algorithm obtains the best performance parameters in all the video sequences except for the Human4 video sequence. Among the correlation filtering algorithms, SACF is the best of its kind, but there is still a big gap compared with the proposed algorithm. Such algorithms can be better applied for object tracking in simple scenes, and the shortcomings of having insufficient robustness are easily exposed in complex backgrounds. Although the linear interpolation update in the correlation filtering model can lead to the tracker gradually adapting to the current features of the object without losing the initial features, gradual accumulation occurs due to tracking errors in the long-term tracking process. When the tracking object is occluded for a long time, this kind of algorithm encounters difficulty in accurately locating the latest position of the object. Compared with the correlation filtering algorithms, the other algorithms based on the Siamese network framework exhibit better tracking performances, except for CFNet. Among them, the proposed algorithm performs better than SiamRPN, largely due to the effective fusion of deep and shallow features and the online updating mechanism of dynamic templates. The robustness of the network model is significantly enhanced by dynamically adjusting parameters and updating templates. Although SA-Siam also deploys a channel attention module, due to the lack of a tracking result detection mechanism, some useless background information is learned by mistake, making it prone to exhibiting tracking drift with cluttered backgrounds. In other words, the proposed algorithm can approach the real object to the greatest extent in a variety of complex scenes, with high tracking accuracy and powerful anti-jamming ability for different interference factors, indicating that the double-template Siamese network model based on multi-feature fusion can greatly enhance the stability of the algorithm while effectively improving its tracking performance.

For autonomous vehicles, one crucial feature is that the tracking speed of the algorithm must meet the real-time requirements. Table 5 shows the tracking speeds of the different object tracking algorithms in the testing sequences.

**Table 5.** The tracking speeds of the different object tracking algorithms in the testing sequences.

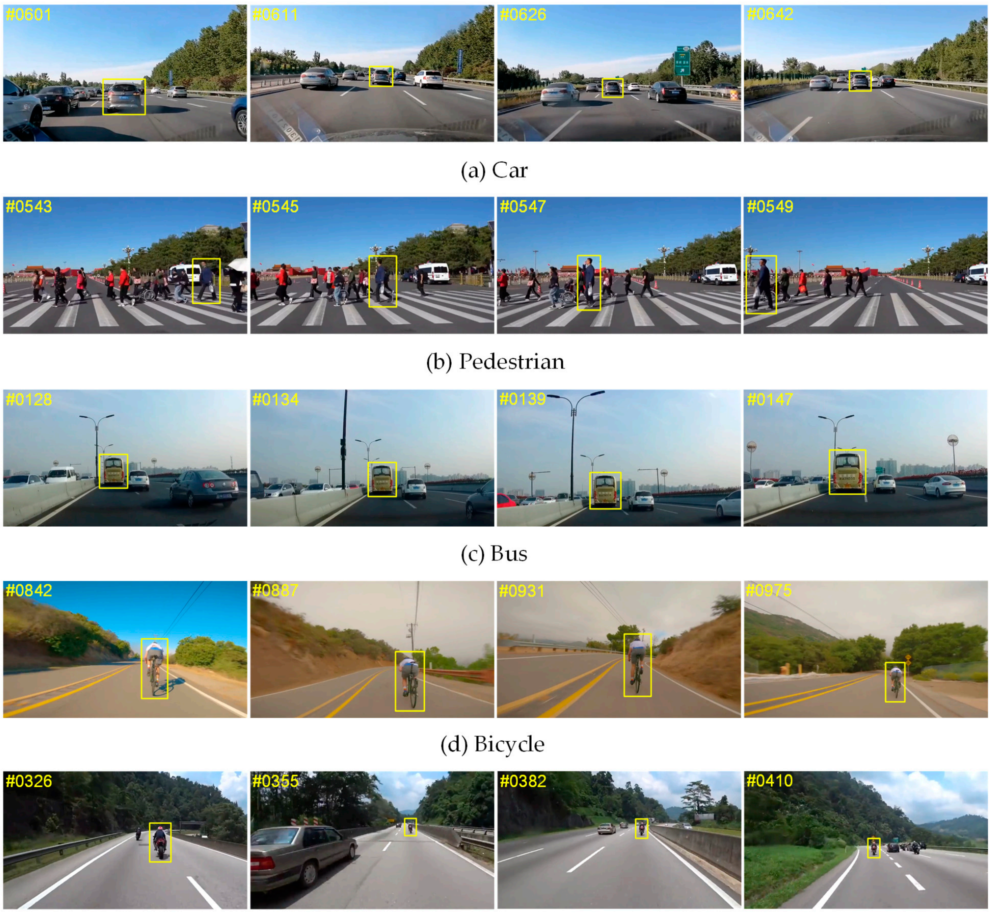
| Algorithm | SRDCF | MCCT | SACF | CFNet | SA-Siam | SiamRPN | Ours |
|---|---|---|---|---|---|---|---|
| FPS | 24 | 40 | 42 | 67 | 52 | 58 | 56 |
| Real time | N | Y | Y | Y | Y | Y | Y |

In this article, the algorithm with tracking speeds above 30 FPS is considered to meet the requirements for real-time tracking. It can be seen intuitively from the above table that, among the correlation filtering algorithms, SACF and MCCT have higher tracking speeds, while SRDCF does not meet the requirements for real-time tracking. The algorithms based on the Siamese network framework all meet the requirements for real-time tracking. Among them, CFNet has the highest tracking speed, reaching 67 FPS, and the proposed algorithm ranks third, reaching 56 FPS, slightly behind CFNet and SiamRPN. The results show that, although the attention mechanism and template online update mechanism are introduced into the network model, it does not have a great impact on the tracking speed, but it further improves the tracking accuracy while ensuring good real-time performance.

### *4.3. Object Tracking Experiment Based on Actual Driving Videos*

To fully test the object tracking performance of the proposed algorithm, in addition to using public datasets for experiments, in this study we also conduct related experiments based on actual driving videos. The actual driving videos used were recorded in urban and rural public roads to better simulate the actual driving environment. The videos are divided into five sequences according to the different tracking objects (car, pedestrian, bus, bicycle, and motorbike) in order to cover more of the objects encountered in the actual driving process. Each video sequence contains a different number of frames. Among the video sequences, that of the car is the largest, with 2849 frames, and the video sequence of the motorbike is the smallest, with 1564 frames. The average number of frames is 2232. The LabelImg tool is used to manually label the tracking objects in the actual driving video sequences, ignoring the label box errors caused by the naked eye, and the default label values are the true values. Figure 9 presents representative images of the tracking results of the proposed algorithm for different actual driving video sequences. Table 6 shows the tracking results for various types of objects in the actual driving video sequences.

The table shows that the testing results are closely related to the types of tracking objects, and the various tracking objects correspond to different testing results. The best testing result is obtained for the pedestrian video sequence: the average OR is 88.1%, the average CLE is 4.8 pixels, and the tracking speed is 62 FPS. The worst testing result was obtained for the bus video sequence, for which the average OR is 80.7%, the average CLE is 10.2 pixels, and the tracking speed is 53 FPS. To sum up, the average OR of the proposed algorithm for the actual driving video sequences is 84.7%, the average CLE is 6.9 pixels, and the tracking speed is 58 FPS, thereby meeting the real-time tracking requirements. The testing results show that the proposed algorithm demonstrates admirable tracking performance for different objects, and the network model has good generalization ability and can realize accurate and real-time tracking of various objects in the actual driving environment.



(a) Car

(b) Pedestrian

(c) Bus

(d) Bicycle

(e) Motorbike

**Figure 9.** Representative images of the tracking results of the proposed algorithm for different actual driving video sequences.

**Table 6.** The tracking results for various types of objects in the actual driving video sequences.

| Sequence Number | Object Type | Average OR (%) | Average CLE (Pixels) | FPS | Frame |
|---|---|---|---|---|---|
| 1 | Car | 85.8 | 6.5 | 59 | 2849 |
| 2 | Pedestrian | 88.1 | 4.8 | 62 | 2670 |
| 3 | Bus | 80.7 | 10.2 | 53 | 2282 |
| 4 | Bicycle | 85.5 | 5.7 | 60 | 1795 |
| 5 | Motorbike | 83.4 | 7.1 | 56 | 1564 |
| Mean | - | 84.7 | 6.9 | 58 | 2232 |

In other words, whether using public datasets or actual driving videos, the double-template Siamese network model based on multi-feature fusion shows good accuracy and real-time performance in experiments, which can be effectively applied to real-time object tracking of autonomous vehicles, and is conducive to the further development and improvement of computer vision technology for intelligent vehicle driving assistance.

## 5. Conclusions

This study proposed a robust object tracking algorithm for autonomous vehicles in complex scenes. We improved and optimized the traditional template structure of the Siamese network and constructed a double-template Siamese network model based on multi-feature fusion. In addition to the first frame-template branch and search branch, the previous frame-template branch was also added. The improved lightweight network MobileNet V2 was used as the backbone network to improve the ability to extract deep and shallow features of objects, and the attention mechanism and template online update mechanism were introduced. Finally, related experiments were carried out based on public datasets and actual driving videos. The testing results showed that the proposed algorithm had high tracking accuracy and speed and good tracking performance for different objects in a variety of complex scenes.

Compared with existing object tracking algorithms, the proposed algorithm exhibits stronger robustness and better anti-interference abilities, and it can still accurately track objects in real time without introducing complex structures. This algorithm can be effectively applied in intelligent vehicle driving assistance, and it will help to promote the further development and improvement of computer vision technology in the field of environmental perception. By efficiently identifying and tracking pedestrians and cars in the surrounding environment, it could have the benefit of greatly alleviating traffic congestion and effectively guaranteeing road traffic safety. Considering the variety of objects in actual road scenes, we plan to conduct in-depth research on a multi-object tracking algorithm and a trajectory prediction algorithm in the future, and carry out hardware implementation and practical application based on FPGA, so as to better meet the real needs of society.

**Author Contributions:** Conceptualization, S.S. and F.X.; methodology, J.C., C.S. and M.H.A.J.; software, J.C., X.Z. and F.X.; validation, J.C., X.Z. and Z.L.; formal analysis, J.C., Z.L. and M.H.A.J.; investigation, J.C., S.S. and Z.L.; resources, C.S. and F.X.; data curation, J.C., X.Z. and Z.L.; writing—original draft preparation, J.C. and C.S.; writing—review and editing, J.C., S.S. and M.H.A.J.; visualization, J.C., X.Z. and F.X.; supervision, C.S. and M.H.A.J.; project administration, S.S. and F.X.; funding acquisition, C.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Science and Technology Development Plan Program of Jilin Province under Grant 20200401112GX, in part by the Industry Independent Innovation Ability Special Fund Project of Jilin Province under Grant 2020C021-3, and in part by the China Scholarship Council under Grant 202006170148.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to express their gratitude to the editors and the anonymous reviewers for their insightful and constructive comments and suggestions, which have led to this improved version of the paper. The authors also appreciate that this work was supported by the National Research Foundation, Prime Minister's Office, Singapore, under its CREATE program, Singapore-MIT Alliance for Research and Technology (SMART) Future Urban Mobility (FM) IRG.

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

## References

1. Zheng, F.; Shao, L.; Han, J. Robust and long-term object tracking with an application to vehicles. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 3387–3399. [CrossRef]
2. Wang, Y. Moving vehicle detection and tracking based on video sequences. *Trait. Signal* **2020**, *37*, 325–331. [CrossRef]
3. Chavda, H.K.; Dhamecha, M. Moving object tracking using PTZ camera in video surveillance system. In Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 1–2 August 2017; pp. 263–266.
4. Granstrom, K.; Renter, S.; Fatemi, M.; Svensson, L. Pedestrian tracking using Velodyne data-Stochastic optimization for extended object tracking. In Proceedings of the 28th IEEE Intelligent Vehicles Symposium (IV), Redondo Beach, CA, USA, 11–14 June 2017; pp. 39–46.
5. Kim, J.; Choi, Y.; Park, M.; Lee, S.; Kim, S. Multi-sensor-based detection and tracking of moving objects for relative position estimation in autonomous driving conditions. *J. Supercomput.* **2020**, *76*, 8225–8247. [CrossRef]
6. Ren, S.; Li, Y. An improved moving object tracking method based on meanshift algorithm. *ICIC Express Lett. Part B Appl.* **2016**, *7*, 1291–1297.
7. Fang, Y.; Wang, C.; Yao, W.; Zhao, X.; Zhao, H.; Zha, H. On-road vehicle tracking using part-based particle filter. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 4538–4552. [CrossRef]

8. Li, H.; Huang, L.; Zhang, R.; Lv, L.; Wang, D.; Li, J. Object tracking in video sequence based on kalman filter. In Proceedings of the 2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC), Chongqing, China, 6–8 November 2020; pp. 106–110.
9. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [CrossRef]
10. Arróspide, J.; Salgado, L.; Nieto, M. Multiple object tracking using an automatic variable-dimension particle filter. In Proceedings of the 2010 IEEE International Conference on Image Processing (ICIP), Hong Kong, China, 26–29 September 2010; pp. 49–52.
11. Du, K.; Ju, Y.; Jin, Y.; Li, G.; Qian, S.; Li, Y. MeanShift tracking algorithm with adaptive block color histogram. In Proceedings of the 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Three Gorges, Yichang, China, 21–23 April 2012; pp. 2692–2695.
12. Nishimura, H.; Nagai, Y.; Tasaka, K.; Yanagihara, H. Object tracking by branched correlation filters and particle filter. In Proceedings of the 4th Asian Conference on Pattern Recognition (ACPR), Nanjing, China, 26–29 November 2017; pp. 85–90.
13. Li, C.; Liu, X.; Su, X.; Zhang, B. Robust kernelized correlation filter with scale adaption for real-time single object tracking. *J. Real-Time Image Process.* **2018**, *15*, 583–596. [CrossRef]
14. Yuan, D.; Zhang, X.; Liu, J.; Li, D. A multiple feature fused model for visual object tracking via correlation filters. *J. Multimedia Tools Appl.* **2019**, *78*, 27271–27290. [CrossRef]
15. Danelljan, M.; Häger, G.; Fahad Shahbaz, K.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the 25th British Machine Vision Conference (BMVC), Nottingham, UK, 1–5 September 2014; pp. 1–11.
16. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
17. Liu, T.; Wang, G.; Yang, Q. Real-time part-based visual tracking via adaptive correlation filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2014; pp. 4902–4912.
18. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Visual tracking with fully convolutional networks. In Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 3119–3127.
19. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4293–4302.
20. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.W.H.; Yang, M.H. Vital: Visual tracking via adversarial learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8990–8999.
21. Amamra, A.; Aouf, N. Real-time multiview data fusion for object tracking with RGBD sensors. *Robotica* **2016**, *34*, 1855–1879. [CrossRef]
22. Cashbaugh, J.; Kitts, C. Vision-based object tracking using an optimally positioned cluster of mobile tracking stations. *IEEE Syst. J.* **2018**, *12*, 1423–1434. [CrossRef]
23. Soleimanitaleb, Z.; Keyvanrad, M.A.; Jafari, A. Object tracking methods: A review. In Proceedings of the 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 24–25 October 2019; pp. 282–288.
24. Dewangan, D.K.; Sahu, S.P. Real time object tracking for intelligent vehicle. In Proceedings of the 1st International Conference on Power, Control and Computing Technologies (ICPC2T), Chhattisgarh, India, 3–5 January 2020; pp. 134–138.
25. Ravindran, R.; Santora, M.J.; Jamali, M.M. Multi-object detection and tracking, based on DNN, for autonomous vehicles: A review. *IEEE Sens. J.* **2021**, *21*, 5668–5677. [CrossRef]
26. Avola, D.; Cinque, L.; Diko, A.; Fagioli, A.; Foresti, G.L.; Mecca, A.; Pannone, D.; Piciarelli, C. MS-Faster R-CNN: Multi-stream backbone for improved Faster R-CNN object detection and aerial tracking from UAV images. *Remote Sens.* **2021**, *13*, 1670. [CrossRef]
27. Liu, Y.; Wang, Z.L.; Cai, B.G. An intelligent vehicle tracking technology based on SURF feature and Mean-shift algorithm. In Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (IEEE ROBIO), Bali, Indonesia, 5–10 December 2014; pp. 1224–1228.
28. Matsushita, Y.; Yamaguchi, T.; Harada, H. Object tracking using virtual particles driven by optical flow and Kalman filter. In Proceedings of the 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, 15–18 October 2019; pp. 1064–1069.
29. Panda, J.; Nanda, P.K. Video object-tracking using particle filtering and feature fusion. In Proceedings of the International Conference on Advances in Electrical Control and Signal Systems (AECSS), Bhubaneswar, India, 8–9 November 2019; pp. 945–957.
30. Judy, M.; Poore, N.C.; Liu, P.; Yang, T.; Britton, C.; Bolme, D.S.; Mikkilineni, A.K.; Holleman, J. A digitally interfaced analog correlation filter system for object tracking applications. *IEEE Trans. Circuits Syst. Regul. Pap.* **2018**, *65*, 2764–2773. [CrossRef]
31. Han, D.; Lee, J.; Lee, J.; Yoo, H.J. A low-power deep neural network online learning processor for real-time object tracking application. *IEEE Trans. Circuits Syst. Regul. Pap.* **2019**, *66*, 1794–1804. [CrossRef]
32. Huang, Y.; Zhao, Z.; Wu, B.; Mei, Z.; Cui, Z.; Gao, G. Visual object tracking with discriminative correlation filtering and hybrid color feature. *Multimed. Tools Appl.* **2019**, *78*, 34725–34744. [CrossRef]
33. Dong, E.; Deng, M.; Tong, J.; Jia, C.; Du, S. Moving vehicle tracking based on improved tracking-learning-detection algorithm. *IET Comput. Vis.* **2019**, *13*, 730–741. [CrossRef]

34. Yang, T.; Li, D.; Bai, Y.; Zhang, F.; Li, S.; Wang, M.; Zhang, Z.; Li, J. Multiple-object-tracking algorithm based on dense trajectory voting in aerial videos. *Remote Sens.* **2019**, *11*, 2278. [CrossRef]
35. Yu, Q.; Wang, B.; Su, Y. Object detection-tracking algorithm for unmanned surface vehicles based on a radar-photoelectric system. *IEEE Access* **2021**, *9*, 57529–57541.
36. Chen, L.; Zhao, Y.; Yao, J.; Chen, J.; Li, N.; Chan, J.C.-W.; Kong, S.G. Object tracking in hyperspectral-oriented video with fast spatial-spectral features. *Remote Sens.* **2021**, *13*, 1922. [CrossRef]
37. Zhu, K.; Zhang, X.; Chen, G.; Tan, X.; Liao, P.; Wu, H.; Cui, X.; Zuo, Y.; Lv, Z. Single object tracking in satellite videos: Deep Siamese network incorporating an interframe difference centroid inertia motion model. *Remote Sens.* **2021**, *13*, 1298. [CrossRef]
38. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional siamese networks for object tracking. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
39. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-end representation learning for Correlation Filter based tracking. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.
40. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning dynamic siamese network for visual object tracking. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1781–1789.
41. He, A.; Luo, C.; Tian, X.; Zeng, W. A twofold siamese network for real-time object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4834–4843.
42. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
43. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. Available online: http://202.113.61.185/xc-ftp/Paper2/Deep_Learning/mobilenetv2.pdf (accessed on 16 January 2018).
44. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
45. 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, 18–22 June 2018; pp. 7794–7803.
46. Real, E.; Shlens, J.; Mazzocchi, S.; Pan, X.; Vanhoucke, V. YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7464–7473.
47. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
48. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 4310–4318.
49. Wang, N.; Zhou, W.; Tian, Q.; Hong, R.; Wang, M.; Li, H. Multi-cue correlation filters for robust visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4844–4853.
50. Zhang, M.; Wang, Q.; Xing, J.; Gao, J.; Peng, P.; Hu, W.; Maybank, S. Visual tracking via spatially aligned correlation filters network. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 484–500.