remote sensing

MDPI

Article

OCSCNet-Tracker: Hyperspectral Video Tracker Based on Octave Convolution and Spatial–Spectral Capsule Network

Dong Zhao [1,2], Mengyuan Wang [1,2], Kunpeng Huang [1], Weixiang Zhong [1,2], Pattathal V. Arun [3], Yunpeng Li [1,2], Yuta Asano [4], Li Wu [1,2] and Huixin Zhou [5,*]

[1] Jiangsu Province Engineering Research Center of Photonic Devices and System Integration for Communication Sensing Convergence, Wuxi University, Wuxi 214105, China; dzhao@cwxu.edu.cn (D.Z.); mywang@nuist.edu.cn (M.W.); hernandez@stu.cwxu.edu.cn (K.H.); wxzhong@nuist.edu.cn (W.Z.); ypli@cwxu.edu.cn (Y.L.); wuli@cwxu.edu.cn (L.W.)
[2] School of Electronics & Information Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China
[3] School of Computer Science & Engineering Group, Indian Institute of Information Technology, Sri City 441108, India; arun.pv@iiits.in
[4] Digital Content & Media Sciences Research Division, National Institute of Informatics, Tokyo 101-8430, Japan; asanoy@nii.ac.jp
[5] School of Physics, Xidian University, Xi'an 710071, China
* Correspondence: hxzhou@mail.xidian.edu.cn; Tel.: +86-029-8820-2553

Abstract: In the field of hyperspectral video tracking (HVT), occclusion poses a challenging issue without a satisfactory solution. To address this challenge, the current study explores the application of capsule networks in HVT and proposes an approach based on octave convolution and a spatial–spectral capsule network (OCSCNet). Specifically, the spatial–spectral octave convolution module is designed to learn features from hyperspectral images by integrating spatial and spectral information. Hence, unlike traditional convolution, which is limited to learning spatial features, the proposed strategy also focuses on learning and modeling the spectral features. The proposed spatial–spectral capsule network integrates spectral information to distinguish among underlying capsule categories based on their spectral similarity. The approach enhances separability and establishes relationships between different components and targets at various scales. Finally, a confidence threshold judgment module utilizes the information from the initial and adjacent frames for relocating the lost target. Experiments conducted on the HOT2023 dataset illustrate that the proposed model outperforms state-of-the-art methods, achieving a success rate of 65.2% and a precision of 89.3%. In addition, extensive experimental results and visualizations further demonstrate the effectiveness and interpretability of the proposed OCSCNet.

Keywords: hyperspectral video tracker; capsule network; spatial–spectral feature extraction

check for updates

Academic Editor: Guoming Gao

Received: 7 January 2025
Revised: 10 February 2025
Accepted: 14 February 2025
Published: 18 February 2025

Citation: Zhao, D.; Wang, M.; Huang, K.; Zhong, W.; Arun, P.V.; Li, Y.; Asano, Y.; Wu, L.; Zhou, H. OCSCNet-Tracker: Hyperspectral Video Tracker Based on Octave Convolution and Spatial–Spectral Capsule Network. *Remote Sens.* **2025**, *17*, 693. https://doi.org/10.3390/rs17040693

Copyright: © 2025 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

The advancements in spectral imaging technology have facilitated quick and viable capturing of hyperspectral images(HSIs),which contain rich spatial and spectral information. In comparison to traditional RGB data, HSI data [1–3] contains hundreds of bands [4,5], providing rich spectral information [6–8]. The richness in spectral information facilitates accurate identification and characterization of the target. The hyperspectral video target tracking [9] has many advantages, such as high accuracy, strong anti-interference ability, and strong adaptability [10,11]. Hyperspectral video tracking algorithms have numerous applications in industries, including aerospace engineering [12], agricultural [13,14] and

ecological environment monitoring [15,16], intelligent monitoring [17,18], and defense applications [19].

State-of-the-art tracking algorithms typically employ a CNN [20,21] for feature extraction. However, the huge number of spectral bands of HSI data limits the direct use of a CNN for feature extraction [22,23]. For instance, there are challenges in dealing with the high spectral correlation of the HSI data. The pooling layers in the CNN cause a loss of valuable information and also ignore the relative spatial relationships between encoded features. Motivated by the need for a more robust and efficient algorithm capable of leveraging the full potential of HSI data, we explore the use of capsule networks. In capsule networks [24], capsules capture and model crucial information about the features. The vector representation of the features in capsules facilitates effective modeling of the relative spatial context. Hence, the representation capability of the model and its ability to recognize complex patterns are improved. In this regard, more and more algorithms are turning to capsule networks. Paoletti et al. [25] applied capsule networks in hyperspectral target classification. Mei et al. [26] designed a capsule residual structure. We explore capsule networks in the field of hyperspectral video object tracking.

The motivation for proposing a hyperspectral video tracker based on octave convolution and a spatial–spectral capsule network (OCSCNet-Tracker) is to gain insight into the potential of capsule networks in integrating spatial and spectral features of hyperspectral data to improve the performance of hyperspectral video tracking (HVT) systems. Specifically, even if two objects are very similar in color, the unique spectral information contained in HSIs can be used as a key basis for identifying the target, which can significantly improve the detection capability of the tracker. At the same time, the neural network is also able to extract rich semantic and detailed information, which is crucial for accurate target classification and position regression. In addition, since the tracking task is inherently a dynamic process that is constantly changing, effective exploitation of the movement trajectory and appearance features of the target plays an important role in accurately predicting its future position. Therefore, the OCSCNet-Tracker is initially designed to exploit the synergistic effect of these multidimensional cues to maximize tracking accuracy.

In this study, we adopt the capsule network as the base framework and propose the OCSCNet-Tracker. The proposed framework comprises four modules, namely, the spatial–spectral octave convolution module, feature extraction module, classification–regression capsule network module, and confidence judgment module. The spatial–spectral octave convolution module learns HSI information by integrating spatial and spectral information. The feature extraction module utilizes the spatial–spectral capsule network to model the spectral information and establish the relationships between different capsules. The proposed approach improves the separability of the target and facilitates relationships between different components and targets at various scales. The classification and regression capsule networks serve to classify candidate samples and predict the bounding box coordinates, contributing to the overall tracking process. The confidence judgment module uses the information of historical frames to make threshold judgments. It also updates the octave convolution fusion module to prevent target loss. The key scientific contributions of this paper can be summarized in detail as follows:

1. We propose a novel capsule network-based tracking framework, OCSCNet-Tracker, which is the first application of capsule networks in hyperspectral target tracking. Its core innovation is centered on modeling the similarity of spatial and spectral information among capsule categories through a spatial–spectral capsule network. By introducing a weight-adaptive allocation mechanism between capsules, the model dynamically adjusts the contribution of different capsules to the target region. This enhances target reconstruction capability and improves robustness against occlusion.

2. We propose a spatial–spectral octave convolution module to fuse spatial and spectral information. The advantage of proposed spectral octave convolution with spectral dimension is to utilize the adaptive weight learning mechanism to process spatial–spectral information. The information fusion is significantly enhanced by cross-information learning and utilizing an internal update mechanism with an external cross-learning strategy.
3. We propose a confidence judgement module with the advantage of optimizing the current frame uncertainty region by using historical frame information. The template update strategy is adaptively adjusted by calculating the intersection over union (IoU) score and classification confidence. When the matching score is low, a weighted average fusion is used to update the template to reduce target drift and increase the robustness.

The article is organized as follows: Section 2 provides a review of the relevant literature; Section 3 introduces the framework and background principles of the hyperspectral video tracking algorithm, particularly the OCSCNet-Tracker; and Section 4 presents the tracking results of the proposed and baseline algorithms on hyperspectral sequences. Finally, conclusions are summarized in Section 5.

2. Related Works

2.1. Visible Video Tracker

In the research related to visible light target tracking, detection-based tracking frameworks have been a major mainstream method of investigation. The main idea of a detection-based target tracking algorithm is to integrate the object detection [27] and tracking tasks. It involves initializing the tracker using the target position [28] and then utilizing the tracking algorithm to consistently follow the target in the video sequence. Voigtlaender et al. [29] proposed a detection-based R-CNN to assess whether the proposed region aligns with the template region. The template objects at any location are redetected based on the similarity between the predicted and template regions, followed by bounding box regression. Li et al. [30] incorporated a region proposal network(RPN) into a Siamese structure, showcasing robust tracking performance. Danellian et al. introduced ATOM [31], which utilizes an IoUNet for offline training on large-scale datasets. Specifically, it employs an online updated convolutional network for modeling target appearance to design a robust target classification module. Overall, detection-based tracking algorithms have demonstrated effectiveness in tracking tasks. To tackle the issue of discontinuity between tracking frames, Wei et al. [32] proposed ARTrack, which introduces spatial–temporal prompts that capture the sequential evolution of trajectories and propagate motion cues, thereby achieving more accurate tracking results. Tumanyan et al. [33] proposed DINO-Tracker, which combines test-time training with DINO-ViT features for long-term dense tracking, outperforming self-supervised methods and excelling in long-term occlusions. However, the application of these algorithms to hyperspectral video target tracking is limited, owing to the huge dimensionality and complexity of the candidate frames. As the octave convolution can effectively extract multiscale features in HSIs, it is optimal for target identification and localization. Meanwhile, the capsule network improves the robustness of tracking by capturing the pose and deformation information of the target. Hence, in this research, we combine the detection-based tracking, octave convolution, and capsule network techniques to develop an integrated hyperspectral video target tracking algorithm. The proposed strategy solves some of the challenges in hyperspectral video target tracking, such as scale variation, illumination variation, and occlusion.

2.2. Hyperspectral Video Target Tracker

In the related research of hyperspectral target tracking, correlation filter-based methods and deep learning-based methods are two main streams. The core idea of the correlation filter-based target tracking method is to use the correlation between the target and the background to quickly locate the target position by frequency domain operation. The method has the advantages of high computational efficiency and easy implementation. For example, ref. [34] proposed a convolutional feature-based kernel correlation filter (KCF) method that uses a 3D hyperspectral cube normalized within the target region as a convolutional kernel, a multidimensional feature representation extracted by a two-layer convolutional network, and further input into the KCF framework for target tracking. Ref. [35] used a 3D gradient spectral spatial histogram to describe the target structure and combined it with a background-aware correlation filter to achieve tracking. To further improve the feature resolution and scale adaptivity of the tracker, ref. [36] proposed an improved HOG feature extractor and a 2D adaptive scale search strategy. However, the correlation filter method suffers from the problem of insufficiently robust manual features. On the other hand, deep learning-based target tracking methods are able to characterize the target more efficiently and improve the robustness and stability of tracking by automatically learning high-level features. The use of end-to-end training reduces the reliance on manual features. In addition, the deep learning method is able to fuse multimodal information to improve tracking performance in complex scenes. To feed the extracted deep convolutional features into the KCF tracker instead of the traditional manual features, ref. [37] proposed a deep hyperspectral kernel correlation filter-based tracking method (DeepHKCF), which combines the manual features and deep convolutional neural network features for airborne target tracking in the hyperspectral domain. Based on the deep features extracted by the deep network, ref. [38] used a histogram of gradients (HOG) and VGG-19 networks to generate feature maps that were able to effectively track targets that were difficult to track in visible video by combining spatial, spectral, and temporal information. However, deep networks face training difficulties with insufficient hyperspectral data. Ref. [39] proposed a deep hyperspectral tracker (SiamHT) based on the Siamese network to estimate the target state by designing a heterogeneous encoder and spectral semantic representation module to extract spatial and spectral semantic feature fusion, respectively. Ref. [40] proposed a two-depth Siamese network framework (SiamHYPER) with the help of a pre-trained RGB tracker to solve the problem of insufficient training due to the lack of hyperspectral data. To address the issue of insufficiently trained models in depth, ref. [8] introduced a Siamese framework called Siamese framework based on band attention grouping (SiamBAG). This SiamBAG framework converts hyperspectral images into pseudo-color images using a band attention module. The pseudo-color images are then fed into a deep network to generate classification, regression, and scale tuning branches, ultimately fine-tuning the target position to enhance tracking performance. To capture motion cues, Chen et al. [41] proposed SENSE, a novel tracker that combines material and motion cues for robust tracking. To tackle scale variation in hyperspectral video tracking, Zhao et al. [42] proposed SASU-Net, a tracker that uses spectral adaptive aggregation and template updating to improve tracking performance.

2.3. Octave Convolution

Researchers have explored various convolution operations to efficiently handle spectral information in hyperspectral data. Among these, the octave convolution, performing convolution on multiple scales, offers a novel perspective for hyperspectral target tracking. Chen et al. [43] introduced octave convolution to process input image signals at multiple scales, drawing inspiration from the frequency analysis of human visual system.

Xu et al. [44] proposed a method for HSI classification by integrating octave convolution with a 3D CNN model. Research has shown that octave convolutions are effective in capturing features across different spectral ranges. Subsequently, Xu et al. [45] further fused octave convolution with capsule networks, enhancing the accuracy and speed of HSI classification. Kan et al. [46] introduced the attention-based octave network for HSI denoising, which successfully preserved spatial and spectral features of HSIs. Tang et al. [47] combined 3D octave convolution with the spatial–spectral attention network, further advancing the performance of HSI classification. In summary, octave convolution has significantly improved HSI classification results [48], offering new perspectives in HSI processing. However, its application in hyperspectral video target tracking has not been explored. Hence, we propose an octave convolution module with a capsule network to enhance the performance of hyperspectral video target tracking. The proposed algorithm integrates the multiscale characteristics of octave convolution with target localization and recognition capabilities, introducing new possibilities in hyperspectral video target tracking tasks.

2.4. Attention Mechanism in Target Tracking

Among the RGB-based tracking algorithms, SE-Net [49] introduced the squeeze-and-excitation (SE) module, which estimates channel weights based on their inter-dependencies. Woo et al. [50] proposed the convolutional block attention module (CBAM), which computes the attention in both spatial and channel dimensions. RASNet [51] combines residual attention, channel attention, and general attention to address the issue of overfitting. Siamattention [52] introduced deformable self-attention and cross-attention mechanisms. Hyperspectral target tracking approaches leverage spectral information for stable tracking. DSP-Net [53] incorporates spectral information into the nonlocal structure. Siamhyper [40] designed a spatial–spectral cross-attention module to achieve efficient and stable tracking. The modeling of spectral and spatial attention has provided crucial information for improving the stability and accuracy of the tracker. Meanwhile, the effective use of spectral information is also key to achieving stable tracking. Therefore, it is significant to employ spectral information within capsule networks in hyperspectral target tracking tasks.

3. Proposed Methods

Figure 1 summarizes the framework of the proposed algorithm. In this section, we introduce the specific details of the proposed algorithm. First, we use the spatial–spectral octave convolution module to extract spatial high-frequency and low-frequency information, as well as main and secondary spectral information. By fusing this information with the weights learned during the training process, we obtain the fusion results. Subsequently, the fusion results obtained and the combined scores obtained from the spectral information are fed into a spatial–spectral capsule network to extract relevant features and establish relationships between the target and multiple scale components. In particular, the fusion results of Frame $t-1$ and Frame t are subjected to a pixel-wise convolution operation after which they are input to a spatial–spectral capsule network to extract features to ensure the correlation between the features of the two frames.

Finally, the outputs of the module are directed to classification and regression capsules, where the former handles the classification of candidate samples, and the latter manages bounding box regression. Additionally, we incorporate a confidence judgment module that estimates the confidence of the predicted target position and decides whether to update the spectral branch of the octave convolution module.

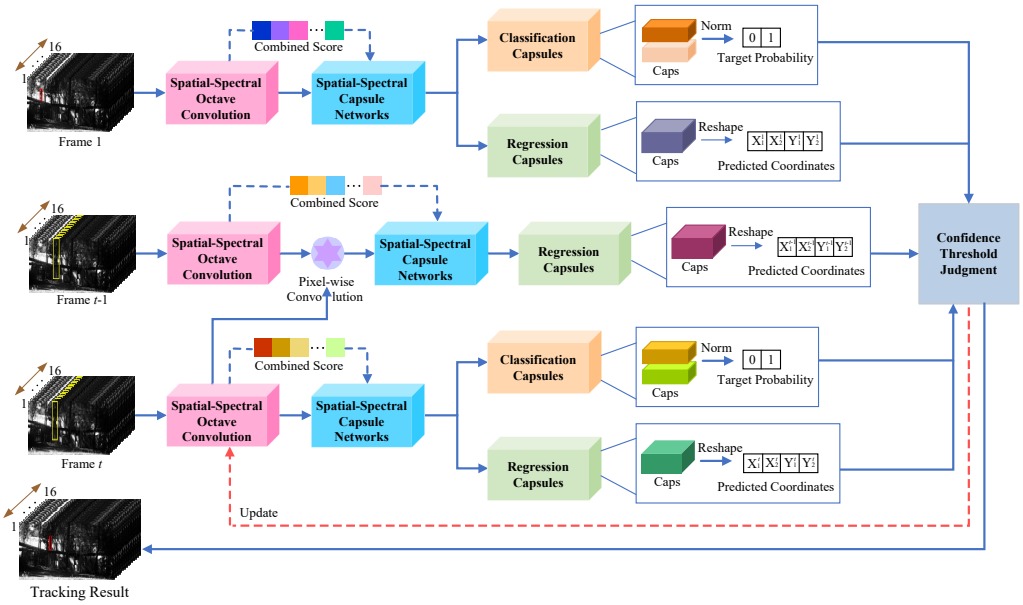

Figure 1. The overall framework of the proposed algorithm.

3.1. Spatial–Spectral Octave Convolution Capsule Model

In this section, we were motivated and theorized that during the convolution process, the convolution kernel slides through the input data several times to extract the information. This configuration increases both memory and computational footprints, posing challenges to both the performance and efficiency of the algorithm. To address the issues related to memory and computational cost, we introduced octave convolutional neural networks. Specifically, the original octave convolutional network treats spectral features as a mixture of different frequency information, which limits its ability to accurately model the spectral and spatial features to a certain extent. To overcome this limitation, we designed a spatial–spectral octave convolution capsule module, as shown in Figure 2.

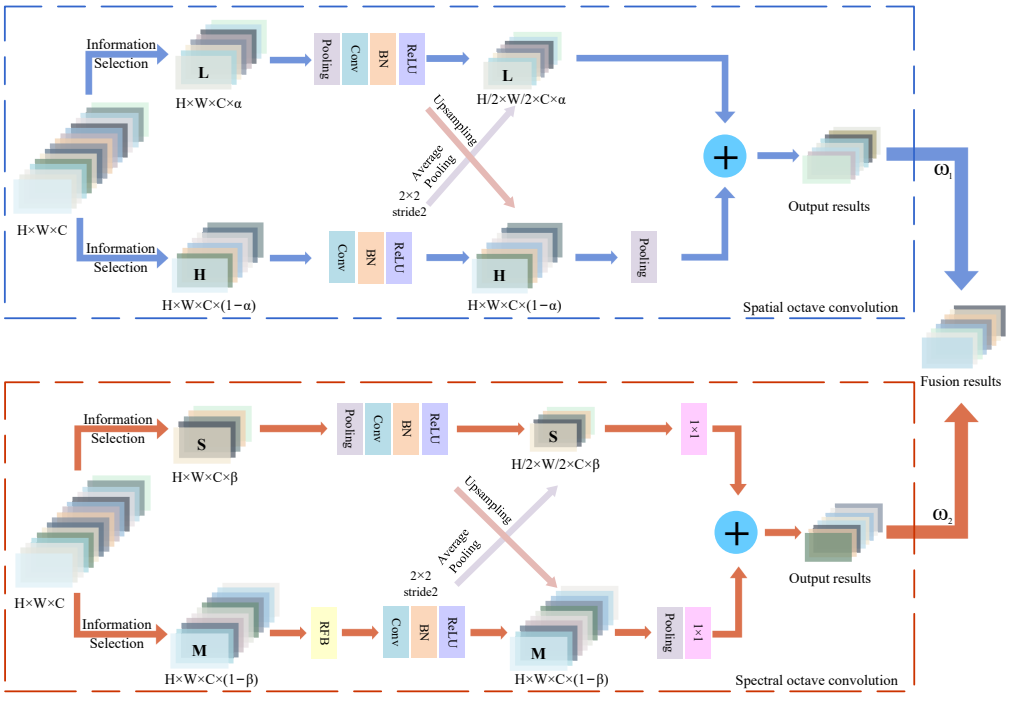

Figure 2. The architecture of the spatial–spectral octave convolution capsule module.

This module includes two key components:

(1) Spatial Octave Convolution

The spatial octave convolution module [43] divides the input hyperspectral image into high-frequency and low-frequency information through information selection along the frequency dimension. The high-frequency and low-frequency information carry out internal information updating and external interaction operations with each other, respectively. This effectively reduces the computational complexity while maintaining the key information and details of the image. In this paper, we use this technique to optimize the performance of processing spatial information models.

(2) Spectral Octave Convolution

As shown in Figure 3, the spectral information is divided into primary and secondary features. Furthermore, the secondary scaling factor β is calculated based on the combined score. If the combined score is higher, the feature contains more information, which is calculated as

$$combined\,score = \gamma(\sigma + E(X)) \tag{1}$$

where σ is the variance of feature map, γ is a constraint coefficient ranging from 0 to 1, and $E(X)$ is the entropy of feature map, which is calculated as

$$E(X) = -\sum_{i=1}^{N} P(i) \cdot \log_2 P(i) \tag{2}$$

where N represents the number of bands in HSIs, which ranges from 0 to 16. $\sum$ represents summation operation and $P(i)$ is the probability of the i-th pixel value. The top $1-\beta$ features selected by the combined score ranking as the main feature scaling factor.

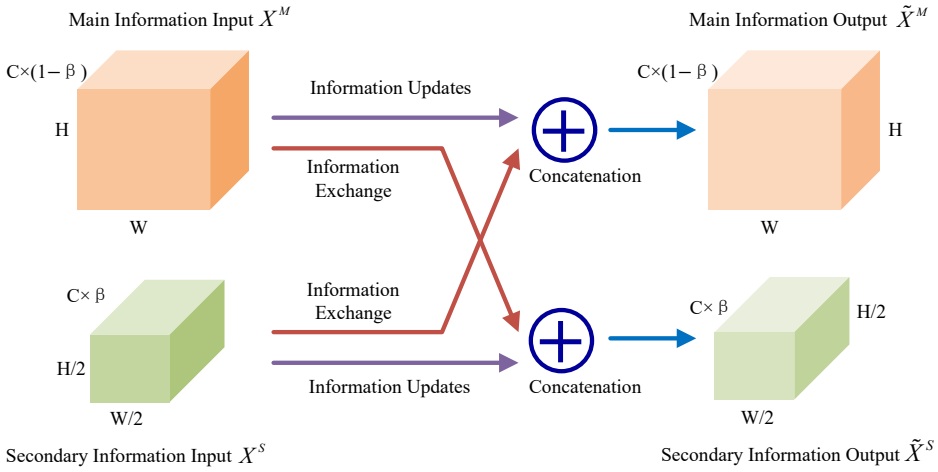

Figure 3. The design of the spectral octave convolution.

$X = \{X^M, X^S\}$, where X is the input feature, X^M is a main feature, and X^S is a secondary feature. The input features X are classified into main features $X^M \in \mathbb{R}^{H\times W\times(1-\beta)C}$ whose spatial resolution is kept constant and secondary features $X^S \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times\beta C}$ whose spatial resolution is reduced to $0.5H \times 0.5W$ along the spectral dimension.

Specifically, in the spectral octave network, the part of information updating inside the main features is $X^{M\to M}$. The part of information updating inside secondary features is $X^{S\to S}$. The part of the information interaction between main features and secondary features is $X^{M\to S}$, and the part of the information interaction between secondary features to the information of main features is $X^{S\to M}$. In order to perform better information updating and interaction, the convolution kernel is divided into the internal weight parameter

$\{W^{M\to M}, W^{S\to S}\}$ and the external weight parameter $\{W^{M\to S}, W^{S\to M}\}$. The output $\widetilde{X}$ can be calculated as follows:

$$\begin{aligned}\widetilde{X}^M &= X^{M\to M} + X^{S\to M} \\ &= conv\left(X^M; W^{M\to M}\right) + upsample\left(conv\left(X^S; W^{S\to M}\right)\right) \quad (3)\\ \widetilde{X}^S &= X^{S\to S} + X^{M\to S} \\ &= conv\left(X^S; W^{S\to S}\right) + conv\left(pool\left(X^M; W^{M\to S}\right)\right) \quad (4)\end{aligned}$$

where $conv(\cdot)$ stands for the convolution operation, $upsample(\cdot)$ denotes an upsampling operation, and $pool(\cdot)$ represents the downsampling operation. The secondary features are upsampled and convolved with main features to generate new main features. The average pooling is performed on the processed main frequency features. It is then convolved with the secondary features to generate new secondary features. The sum of these two newly generated features produces the final spectral feature. It is worth noting that the original main features are processed by receptive field blocks (RFBs) to increase the receptive field of the network. In addition, a 1×1 convolutional layer is added at the end of the network to adjust the scale to meet the output requirements. Finally, the spatial feature results and spectral feature results are fused according to the learned weights, aiming to maximize the use of spatial spectral information in HSIs and enhance the feature extraction effect of the network.

The weight values of spatial and spectral features (ω_1, ω_2) are dynamically learned and adjusted during the training process. As training progresses, the spatial–spectral octave convolution capsule model gradually learns which spatial and spectral features are most critical for the target tracking task and adjusts the weight values of these features accordingly. This dynamic adjustment process helps to maximize the use of spatial and spectral information in the HSI and enhance the feature extraction effect of the network. Specifically, the network initially assigns equal weights of 0.5 to both the spatial and spectral features. During the training process, as the network gradually understands and adapts to the data, it gradually adjusts these weight values so that the features that contribute more to the target tracking task receive higher weights. This change in weights is adaptive, thus reflecting more accurately the importance of different features in the target tracking task. In the end, the network is able to capture the information of the target in the HSI more comprehensively by fusing the spatial and spectral features adjusted according to the learned weights.

3.2. Spatial–Spectral Capsule Network

The features extracted by our proposed capsule network have invariant and comorphic properties. While capsule networks effectively reflect hierarchical structures and overcome occlusion challenges, their training process is slow and resource-intensive, posing challenges in the direct application to hyperspectral video object tracking. Since the traditional capsule network lacks information on how to utilize spectral data, the proposed algorithm requires network to be able to learn key features in the target and separate them from the background. We propose a spatial–spectral capsule network to process hyperspectral data and extract the relationships between multiscale features. However, each capsule is used equally for subsequent learning processes without any differential treatment. In order to improve the discriminative ability of different capsules during network tracking, we utilized the parameters used in the spatial–spectral octave convolution to learn spectral weights that reflect different spectral importance.

Figure 4 shows the structure of the spatial–spectral capsule interaction in the spatial–spectral capsule network module. Firstly, the fusion results obtained from the spatial–spectral octave convolution module are fed into the n different capsule layers of the capsule network to extract the unique capsule depth features for each band with vector length m. At the same time, the spectral information is extracted by using the spectral octave convolution module, and the combined score of the different bands is computed. Subsequently, features from the same band but from different capsule layers are efficiently integrated into the $(1 \times n \times m)$ combined capsule of band c using the group connection technique. Then, based on the bands, the obtained combined scores are multiplied element by element with the composite features, and the features are normalized with a sigmoid function to obtain a $(1 \times n \times m \times c)$ capsule feature. In Figure 4, n represents the number of capsule layers, m represents the length of the capsule deep feature vector, and c denotes the different bands. As shown in Figure 4, in the main capsule layer, the extracted spectral information a_s is computed by utilizing a combined score, which is calculated in Equation (1) as follows:

$$a_s = combined\ score = \gamma(\sigma + E(X)) \tag{5}$$

where a_s is the extracted spectral information. At the same time, the underlying capsules are grouped into n $(1 \times 1 \times m)$ capsule groups and merged into a $(1 \times n \times m)$ shape in the channel dimension through group connection operations to prevent information interference from other surrounding capsules:

$$a_{gc} = Group\ Connection(a_s) \tag{6}$$

where a_{gc} is group-connected capsules. The capsules are connected by internal similarity within the capsule network into groups of capsules. The spectral information is element-wise multiplied by channel and capsules. The values are limited by using the sigmoid function:

$$a_f = Sigmoid(a_s \cdot a_{gc}) \tag{7}$$

where a_f is the capsule attention value. This process emphasizes the importance of spectral information, which selectively strengthens or weakens the influence of different channels. The results are applied to the dynamic routing algorithm through the following calculation:

$$\hat{u}_i = a_f \cdot u_i \tag{8}$$

where $\hat{u}_i$ is the low-lever capsule after the spectral capsule network operation, and u_i is the original low-level capsule.

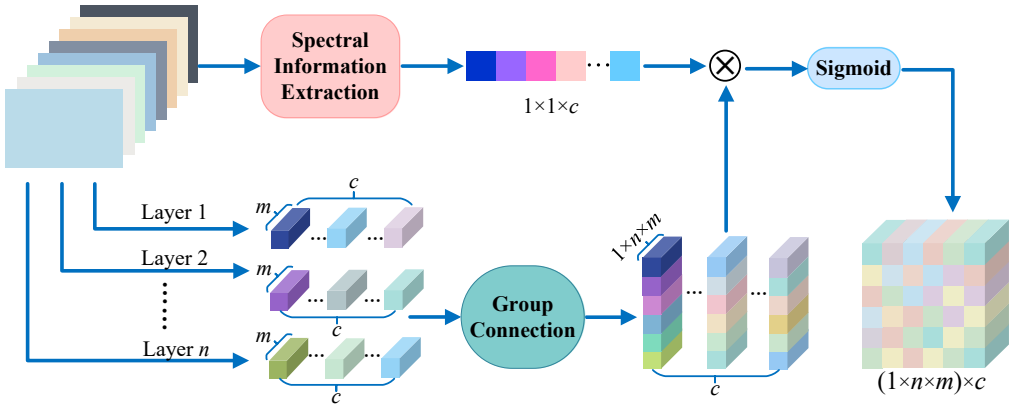

Figure 4. The spatial–spectral capsule module.

The dynamic routing process is illustrated in Figure 5. First, the spatial–spectral features u_i are passed to multiple capsule layers after extracting the vectors generated from the basic features in the image such as edges, texture, etc., through the initial convolutional layer of the spatial–spectral capsule network. The different capsule layers contain low-level capsules and high-level capsules. The low-level capsule $\hat{u}_i$ is responsible for detecting local features in the image, which are usually smaller and more specific, such as facial features like the eyes of the toy shown in Figure 6. The high-level capsule $u_{j|i}$ integrates the information from the low-level capsule to form a representation of more abstract, global features, such as facial or body features like the eyes of the toy shown in Figure 6. The low-level capsule converts its output vectors into prediction vectors via a transformation matrix $W_{i|j}$, and these prediction vectors are sent to high-level capsules that may be relevant. The high-level capsule updates the coupling coefficients c_{ij} with the low-level capsule based on the consistency of the prediction vectors. Through many iterations, the dynamic routing mechanism is able to determine the best connection between the low-level capsule and the high-level capsule to efficiently deliver entity information. Next, the obtained class capsule vectors $\bar{v}_1, \bar{v}_j$ are activated using the squash function. Each capsule in the previous layer learns the relationship between the low-level capsule and the high-level capsule by means of a transformation matrix. Hence, the output probability of each capsule in the next layer is predicted. The prediction from the previous layer to the next layer is expressed as

$$u_{j|i} = W_{ij}\hat{u}_i \tag{9}$$

where $u_{j|i}$ represents the output of the upper-layer capsule i to the next-layer capsule j, $\hat{u}_i$ represents the output vector of the upper-layer capsule i, and W_{ij} is transformation matrix of the upper-layer capsule to the next-layer capsule. In practice, many underlying capsules contribute significantly to subsequent prediction vectors. We assigned a larger weight to the prediction output that matches the target, ensuring it is primarily routed to the corresponding capsule. Simultaneously, we assigned a smaller weight to dissimilar prediction outputs, routing only a minor portion of them to other capsules. This weight is defined by the following coupling coefficients:

$$c_{ij} = \frac{exp(b_{ij})}{\sum_j^k exp(b_{ik})}_{j\in\{1,\cdots,k\}} \tag{10}$$

where c_{ij} represents the coupling coefficient ranging from 0 to 1, $\sum$ stands for the summing operation, exp is an exponential function, and b_{ij} is the initial weight initialized to 0. During the dynamic routing process, the initialization weights are updated based on the similarity between the predicted vector and the vector compression:

$$b_{ij} = b_{ij} + u_{j|i} \cdot \bar{v}_j \tag{11}$$

where $\bar{v}_j$ represents compression of vector v_j, b_{ij} is the initial weight initialized to 0, and $u_{j|i}$ represents the output of the upper layer capsule i to the next layer capsule j. When the prediction vector $u_{j|i}$ is more similar to the output of the next capsule v_j, the larger $u_{j|i} \cdot \bar{v}_j$ is, the more the initial weight b_{ij} will increase, and the coupling coefficient c_{ij} will also increase. Conversely, the smaller $u_{j|i} \cdot \bar{v}_j$ is, the more the initial weight b_{ij} will be smaller, and the coupling coefficient c_{ij} will also become smaller. The output of j-th capsule in the next layer is the sum of the weights of all capsules in the previous layer and the corresponding predicted capsule vectors:

$$v_j = \sum c_{ij} \cdot u_{j|i} \tag{12}$$

where v_j represents the output of the next capsule, c_{ij} represents the coupling coefficient, and $u_{j|i}$ represents the output of the upper layer capsule i to the next-layer capsule j. After obtaining the output v_j, the value of the vector $\|v_j\|$ is used to represent the presence of features. In order to better represent capsule features, the length of the vector is used to represent the probability in the current input. The length of the vector is compressed to between 0 and 1. The nonlinear squeeze function formula is as follows:

$$\bar{v}_j = squash(v_j) = \frac{\|v_j\|^2}{1+\|v_j\|^2} \cdot \frac{v_j}{\|v_j\|} \tag{13}$$

where $\bar{v}_j$ represents the compression of vector v_j , and $\|\|$ is the size of the vector. When $\|v_j\|$ tends to 0, $\bar{v}_j$ tends to $v_j\|v_j\|$, $\bar{v}_j$ is compressed close to 0, and the direction is retained; when $\|v_j\|$ tends to $+\infty$, $\bar{v}_j$ tends to $\frac{v_j}{\|v_j\|}$, $\bar{v}_j$ is compressed close to 1, and the direction is retained. The capsule network employs a dynamic routing to transfer information from the lower-level capsules to the higher-level capsules.

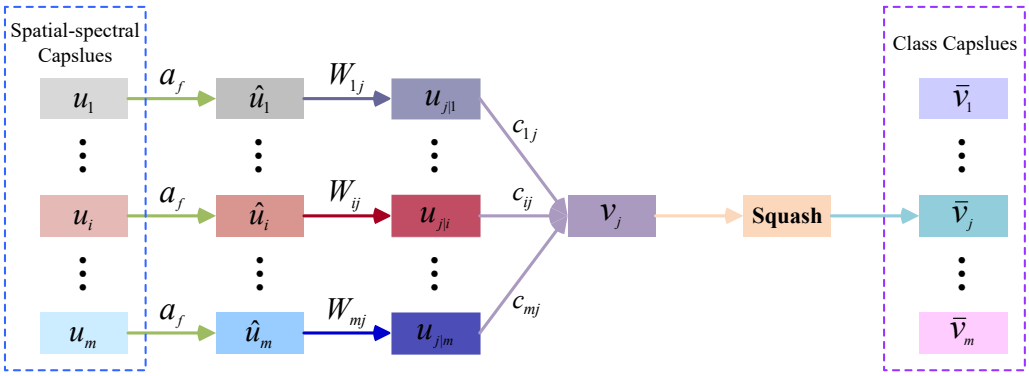

Figure 5. Dynamic routing process.

Figure 6 presents visualizations of the feature maps from the various layers of capsules within the spatial–spectral capsule network. The HSIs used in this experiment cannot be observed by the human eyes, so the first column of Figure 6 shows the RGB images of the four sequences of inputs (ball, bus, face, and toy). The subsequent three columns denote the results for the features of the different layers. From the results, the relationship between the input image, the underlying capsule, and the upper layer of capsules is clearly visible. The second column shows the visualized feature results of the low-level capsule, showing the rough outline edge features of the object in the image, while the third and fourth columns show the features of the different capsules in the middle. As the layers are elevated, the detail information in the feature maps gradually decreases, and the edges are gradually weakened, while the overall structure and shape information is gradually enhanced. This reflects that the capsule network gradually shifts from details to the whole and from local features to global features when integrating information. It can be concluded that the previous low-level capsule layers are responsible for learning the detail information, while the later capsule layers learn to extract the overall features. The lower-level capsules provide the higher-level capsules with the base feature information that is necessary to construct more complex and abstract features. The higher-level capsules form an accurate representation of the overall structure and shape in the image by integrating the information from the lower levels. This representation not only incorporates the local features in the image but also reflects the spatial relationships and hierarchical structure between these features.

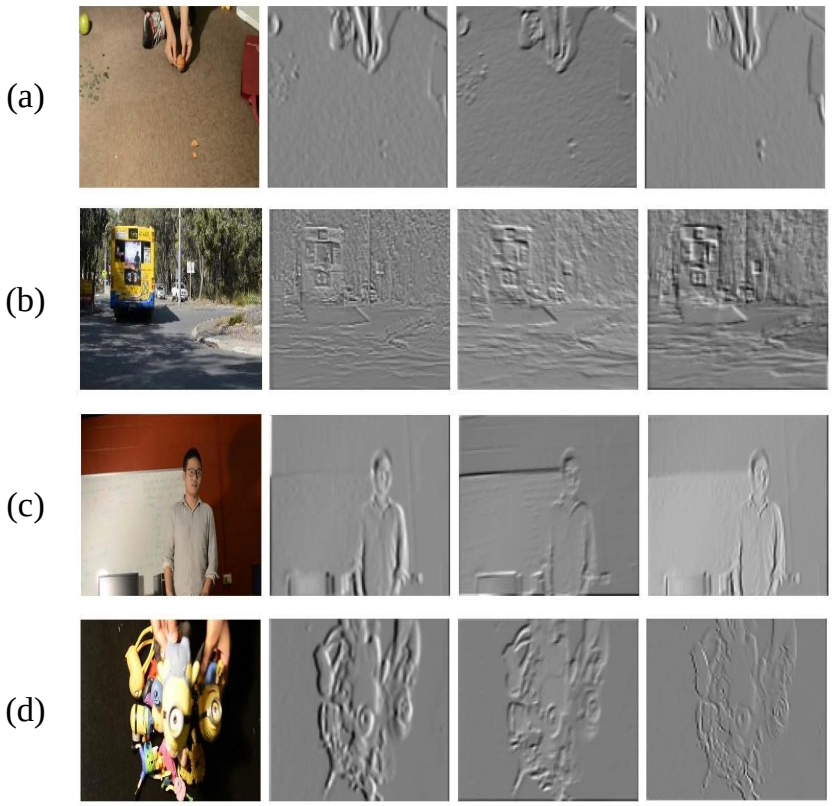

Figure 6. The visualization of partial image feature map results for the underlying capsule convolutional layer of the spatial–spectral capsule network. (**a**) Ball sequence. (**b**) Bus sequence. (**c**) Face sequence. (**d**) Toy sequence.

3.3. Classification and Regression Capsules

A complete capsule model was designed within the classification capsule network for distinguishing between target and background. The classification capsule network consists of two convolutional capsule layers. The first convolutional capsule layer contains eight capsules, and the second convolutional capsule layer contains two capsules. These two convolutional capsule layers are connected through dynamic routing [24]. The classification capsule outputs a classification score, representing the probability of whether the current candidate sample belongs to a positive or negative class. In other words, when the current candidate sample belongs to the target, the classification score value is 1; otherwise, it is 0.

Similarly, a comprehensive capsule structure model was designed in the regression capsule network to predict the coordinates of the optimal bounding box. The regression capsule network includes two convolutional layers. The first convolutional capsule layer comprises eight capsules. The second convolutional capsule layer contains one capsule. These two convolutional capsule layers are connected through dynamic routing. The regression capsule outputs the coordinates of the upper-left and lower-right corners of bounding box.

The loss functions include classification loss and regression loss. The total classification loss is the sum of the losses for the classification capsule layer:

$$\mathcal{L}_{Cls} = \sum_{k}[T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k)\max(0, \|v_k\| - m^-)^2] \quad (14)$$

where m^+ is the reference top capsule vector of 0.9 for which the current class exists, the m^- value is the reference top capsule vector of 0.1 for which the current class does not exist, and λ represents the loss reduction weight for the missing class used to reduce the length of

the digital capsule vector of the class, and it has the value 0.5. When the k-th capsule exists, the T_k value is 1; otherwise, it is 0. The regression loss is calculated using the GIOU loss $\mathcal{L}_{GIoU}$ [54]:

$$\mathcal{L}_{GIoU} = 1 - GIoU = 1 - \left(IoU - \frac{|F - (A \cup B)|}{|F|} \right) \tag{15}$$

where A is the prediction box, B is the ground truth box which is manually labeled, and F is the minimum bounding rectangle between the prediction box and the target box. Therefore, the total loss is the sum of classification loss and regression loss:

$$\mathcal{L}_{total} = (1 - \theta)\mathcal{L}_{Cls} + \theta\mathcal{L}_{GIoU} \tag{16}$$

where θ is the regression loss weight used to balance the classification loss and regression loss.

3.4. Confidence Threshold Judgment Model

During tracking, if the target is obscured by other objects, tracking may fail. Therefore, it is crucial to update the template, so we introduced a confidence threshold judgement module. When the tracking target is lost, our tracking approach compares with the adjacent frame and the first frame to update the template and then repositions the target to continue tracking. In computer vision, the features of a target are essential for recognizing and tracking the target. When a target moves or is occluded in a video, its appearance may change, but certain features remain the same or have similarities. Therefore, by comparing the features of the adjacent frames with the initial frame, the region that is most similar to the target can be found and thus the target can be relocated. The process involves obtaining the classification score P_c^1 and predicted bounding box coordinates D_c^1 of the first frame as well as the classification score P_c^t and predicted bounding box coordinates D_c^t of the t-th frame. Features similar to the t-th frame are extracted from adjacent frames through the middle branch of the network. A predicted bounding box coordinate D_c^{t-1} is then used to calculate the score of the prediction box for the current frame:

$$D_c^1 = \left(x_1^1, y_1^1, x_2^1, y_2^1, r^1, h^1 \right) \tag{17}$$

$$D_c^{t-1} = \left(x_1^{t-1}, y_1^{t-1}, x_2^{t-1}, y_2^{t-1}, r^{t-1}, h^{t-1} \right) \tag{18}$$

$$D_c^t = \left(x_1^t, y_1^t, x_2^t, y_2^t, r^t, h^t \right) \tag{19}$$

where (x_1, y_1) is the coordinates of the upper-left corner of the bounding box, (x_2, y_2) is the coordinates of the lower-right corner of the bounding box, r is the width of the predicted bounding box, and h is the height of the predicted bounding box. The IoU score of the prediction box is used to calculate the score of the prediction box of the current frame:

$$\Psi\left(D_c^1, D_c^{t-1}, D_c^t \right) = \begin{cases} 1 & \text{if } D_c^t \in D_c^{t-1},\ D_c^t \in D_c^1 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where Ψ represents the score of the prediction box of the current frame. The judgment conditions are the following: (1) when the IoU value of the t-th frame prediction box and the first frame prediction box is greater than or equal to 0.5; (2) when the IoU value of the t-th frame prediction box and the t-1th frame prediction box is greater than or equal to 0.7. In other cases, the current frame score is 0. These two thresholds are set based on the following considerations: an IoU ≥ 0.5 with the first frame ensures that the target in the selected frame has sufficient overlap with the initial target, thus guaranteeing the consistency of the target identity. An IoU ≥ 0.7 with the previous frame, on the other hand, ensures the continuity of the target between consecutive frames and reduces false tracking

due to mismatch or noise interference. With these two conditions working together, both accurate and continuous tracking results can be filtered to provide a reliable basis for template updating. The overall score of the current frame is calculated as

$$Score_{total} = P_c^1 + P_c^t + \Psi\left(D_c^1, D_c^{t-1}, D_c^t\right) \tag{21}$$

where P_c^1 is the classification score which represents the probability of whether the first frame is the target, P_c^t is the classification score which represents the probability of whether the t-th frame is the target, and $Score_{total}$ represents the overall score of the current frame. Based on the score of the predicted bounding box and the classification score of the current frame, it is determined whether to update the template:

$$Update = \begin{cases} Yes & \text{if } Score_{total} < 3 \\ No & \text{otherwise} \end{cases} \tag{22}$$

where $Update$ represents the update operation. The template will be updated when the following conditions are met: First, the total score $Score_{total}$ is less than 3, which indicates that the target in the t-th frame is a poor match to the initial template, but there is still room for improvement. Second, the IoUs of the selected frame with both the first frame and $t-1$th frame need to satisfy the threshold set above to ensure that the updated template is accurate and continuous. In the updating process, we adopt a weighted average method to fuse the spectral features of the t-th frame with those of the $t-1$th frame to smoothly update the template and avoid tracking instability due to sudden changes. These operations are repeated until the tracking results are obtained.

4. Experiment

4.1. Dataset and Metrics

The dataset, adopted in this study, is the hyperspectral dataset HOT2023. The HOT2023 dataset is accessible at www.hsitracking.com (accessed on 13 February 2025), comprising 40 training and 35 test videos. Each video includes hyperspectral data, RGB data, and false-color data generated from the hyperspectral data. The imaging wavelength of the camera ranges from 470 nm to 620 nm, and the collected HSIs consist of 16 bands with a bandwidth about 10 nm. These video scenes involve 11 challenging conditions, such as background clutters (BCs), deformation (DEF), fast motion (FM), in-plane rotation (IPR), illumination variation (IV), low resolution (LR), motion blur (MB), occlusion (OCC), out-of-plane rotation (OPR), out-of-view (OV), and scale variation (SV).

As shown in the Figure 7, we selected four sequences (ball, bus2, face2, and toy2) in the HOT2023 dataset. These sequences are all from the hyperspectral dataset published in [35]. Table 1 shows the parameters of the four test sequences in detail, and Figure 7 shows the RGB images of the four test sequences. We selected four sequences for testing the performance of the algorithm against occlusion. Additionally, to assess the generalization capability of our tracker, we deliberately selected sequences with various challenges, including scale change, background clutter, in-plane rotation, illumination changes, motion blur, out-of-plane rotation, and out-of-view.



(**a**) ball (**b**) bus2 (**c**) face2 (**d**) toy2

Figure 7. Four RGB sequences.

Table 1. Parameters of the test sequences.

Sequences	ball	bus2	face2	toy2
Frames	625	326	1111	601
Resolution	471 × 232	273 × 149	389 × 222	435 × 194
Initial Size	24 × 24	67 × 75	46 × 60	65 × 78
Challenges	OCC, MB, SV	OCC, SV, IV, FM	IPR, OPR, SV, OCC	BC, OCC, SV, IV, OPR

The performance of the proposed and nine benchmark algorithms was evaluated using the success rate and precision rate. The success rate reflects the estimation accuracy of the predicted bounding box, which is the percentage of all frames in which the overlap area of the target and the ground truth bounding boxes is above a threshold. The overlap score is calculated as

$$OS = \frac{P_s \cap G_s}{P_s \cup G_s} \tag{23}$$

where P_s is the target bounding box area of our algorithm, and G_s is the area of the ground truth. As discussed, when the overlap between the bounding boxes exceeds a threshold, the tracking is considered successful. Specifically, for the coordinate system of the success rate curve, the x axis denotes a series of continuous thresholds from 0 to 1, with different thresholds set. Similarly, the y axis is the success rate. The precision rate reflects positioning precision, which is the percentage of all frames in which the distance between the center of predicted target bounding box and the ground truth is less than a threshold. The distance D between the center point of the target bounding box center and the groundtruth value is calculated as

$$D = \sqrt{(m_1 - m_2)^2 + (n_1 - n_2)^2} \tag{24}$$

where (m_1, n_1) is the center point of the ground truth, and (m_2, n_2) is the predicted center point of the target bounding box. Specifically, for the coordinate system of the precision rate curve, the x axis is a series of continuous thresholds from 0 to 50, and the y axis is precision. The average distance precision rate is reported at a threshold of 20 pixels (DP@20). Additionally, we used the area under the curve (AUC) to measure the performance of each tracking algorithm.

4.2. Implementation Details

The training and inference stages of the OCSCNet-Tracker were implemented in Python using PyTorch on an NVIDIA RTX 3090 GPU. The experimental training data parameters were as follows: the low-frequency scaling factor α of the spatial–spectral octave convolution module was initially set to 0.5. In the spatial–spectral octave convolution module, low-frequency information is often associated with a reduction in spatial resolution. Thus, α can be seen as a parameter that controls the degree of spatial resolution reduction. When α increases, the proportion of low-frequency information increases, which may lead to a decrease in spatial resolution. Conversely, when α decreases, an increase in the proportion of high-frequency information may improve spatial resolution. During training, the value of α is adaptively determined by the optimization algorithm. This means that the value of α is adjusted according to the characteristics of the input data and the task requirements. γ in the combined score was set to 0.7. The constraint coefficient γ is a key parameter that represents the combined share of variance and information entropy in the spectral information in the input image spectral information. This share coefficient reflects the importance and influence of the spectral information in the image as a whole and is crucial for the performance optimization of the algorithm. A suitable value of the combined coefficient, i.e., 0.7 was determined by training and analyzing the training set to ensure the stability and accuracy of the algorithm or model in a particular application

scenario. λ in the classification loss function $\mathcal{L}_{Cls}$ was set to 0.5, and θ in the total loss function $\mathcal{L}_{total}$ was set to 0.5. For offline training, we trained the network with a learning rate of 0.0001. For online updates, the learning rate was set to 0.0003. The routing iteration was set experimentally to three times.

4.3. Comparison of OCSCNet-Tracker with SOTA Algorithms

In this experiment, we compared the performance of the proposed algorithm with several state-of-the-art hyperspectral tracking algorithms, namely, MFI-HVT [55], MHT [35], DeepHKCF [37], MixFormer [56], CNHT [34], ProContEXT [57], SiamFC [58], SiamBAG [8], and SiamOHOT [59]. These trackers include a number of RGB trackers. The RGB trackers are the result of testing on false-color sequences from the HOT2023 dataset. The false-color data are generated from hyperspectral data and are fully equivalent to the original hyperspectral data. MFI-HVT combines deep features with traditional manual features to generate 3D HOG features for target tracking. The MHT method leverages different spectral principles of various materials to extract spectral information from the target, distinguishing even objects with similar colors. DeepHKCF involves converting HSIs into pseudo-color images for processing, with a focus on tracking tasks. The MixFormer method adopts an end-to-end transformer model that utilizes the attention mechanism to simultaneously extract and interact with features. To address the limitations of local-window self-attention, MixFormer combines local-window self-attention with depth-wise convolution in a parallel design, which models cross-window connections to enlarge the receptive field. Additionally, MixFormer introduces bi-directional interactions across branches, providing complementary clues in both the channel and spatial dimensions. These designs enable effective feature mixing among windows and dimensions, improving the overall performance. Furthermore, MixFormer includes a simple regression head to directly output tracking results. In CNHT, information from the initial frame serves as a key for processing and tracking subsequent frames. ProContEXT integrates temporal context information and spatial context information to extract features for target tracking. SiamFC is a traditional twin network method with a structure that shares convolutional layer weights and employs similarity matching for tracking. SiamBAG trains a deep neural network on RGB datasets while using band attention to reorganize HSIs into three-channel color images for tracking. SiamOHOT uses a Siamese network to learn RGB trackers for extracting spatial features while learning hyperspectral trackers for extracting spectral features for tracking. By comparing the OCSCNet-Tracker with these state-of-the-art techniques, we aim to demonstrate its effectiveness and competitiveness in hyperspectral tracking applications.

Figure 8 summarizes the results of the algorithm discussed with nine other algorithms on four sequences. In Figure 8, rectangular boxes of different colors represent the tracking results of various tracking algorithms. Specifically, sky blue corresponds to the proposed algorithm, green represents CNHT, red represents DeepHKCF, pink and purple represent MHT, brown corresponds to SiamFC, blue corresponds to MFI-HVT, yellow represents MixFormer, black represents ProContEXT, dark red represents SiamOHOT, purple corresponds to SiamBAG, and white represents the ground truth. The effectiveness of different algorithms on a frame in the sequence was evaluated based on the proximity between the rectangular boxes with different colors and the white rectangular box. For instance, when the position or size of a rectangular frame is closest to the white rectangular frame, it indicates that the algorithm corresponding to this color has the best tracking effect. Given the multitude of rectangular boxes in Figure 8, direct observation of the optimal tracking effect for a particular algorithm may be challenging. Therefore, success rate curves and precision curves were employed to illustrate the tracking performance.

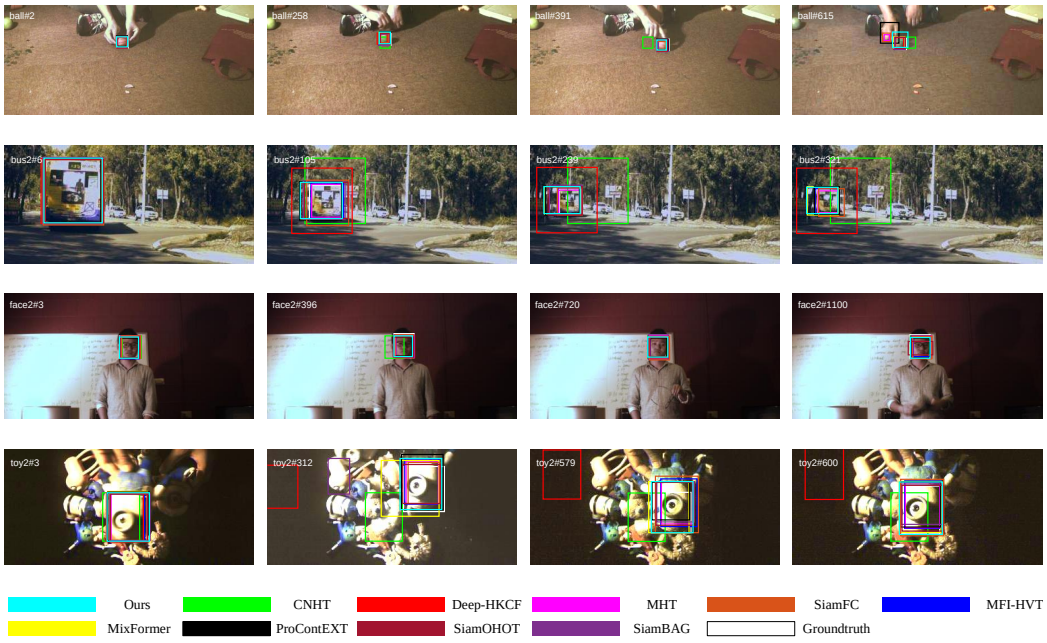

Figure 8. Qualitative results on four sequences.

In Figure 8, the first row illustrates the tracking results of the ball sequence. The small orange ball is observed rolling left and right on the ground, pushed by human fingers. Around frame 497, when the human hand picks up the ball and repeatedly bounces it, a significant occlusion and deformation occur. In the experiment, as it reaches the 67th frame, the scale of the ball changes. Notably, the CNHT algorithm exhibited a lack of robustness, leading to tracking failure. At frame 598, the ball is partially occluded by a human hand picking it up, causing the MHT algorithm, the ProContEXT algorithm, and the SiamFC algorithm to lose the target by tracking the ball in subsequent frames using incorrect information.

In the second row of Figure 8, the tracking results for the bus sequence are depicted. The presence of trees on both sides of the road, as the bus travels from near to far, results in partial sunlight blockage and illumination changes. Additionally, as the bus moves away from sight, it becomes partially blocked by the trees. Around frame 197, a white car emerges from the opposite direction, causing interference. Notably, at approximately the 100th frame, the highly exposed bus posed a challenge for the CNHT algorithm and the DeepHKCF algorithm, leading to tracking failure in the context of illumination variation (IV). After frame 189, as the bus becomes partially obscured by roadside trees, the SiamFC algorithm lost track of the target in subsequent frames. While our tracking results are slightly biased due to excessive background information, subsequent ablation experiments have demonstrated the reasonableness of our updates. Although our algorithm may not have precisely aligned with the ground truth, it effectively framed the entire bus.

The third row in Figure 8 is the tracking result of the face sequence. People shake their heads left and right in front of the camera and perform actions such as nodding, raising their heads, turning their heads, taking off their glasses, and covering their faces. In the experiment, after the left and right shaking action occurred at frame 396, the CNHT algorithm failed to track because it only used positive samples for training. The positions of the objects have not changed much compared with the first frame, so the tracking difficulty of the entire sequence is relatively small. Even if all algorithms can accurately follow the face, it can still be seen that not all algorithms can accurately estimate the scale of the face. For example, the object bounding box generated by SiamFC, ProContEXT algorithm, and MFI-HVT algorithm at frame 720 is very different from the ground truth.

The last row in Figure 8 is the tracking result of the toy sequence. Two minion toys are criss-crossing and moving in front of the camera. There are many small toys placed in the background. After the 155th frame, the image quality changes, and the target disappears from the screen and then slowly reappears. In the experiment, when running up to around frame 310, two identical minion toys are making cross movements in the hand; the SiamBAG algorithm, CNHT algorithm, and DeepHKCF algorithm lost the target, and the MixFormer algorithm failed to accurately frame the target. After frame 579, all algorithms except CNHT and DeepHKCF algorithms maintained good tracking of the Minion toy, but there was a difference in the scale of the target bounding box.

Figure 9 shows the success rate curve and precision curve of all algorithms on four sequences. In addition, in order to compare the performance of the proposed algorithm and other algorithms more intuitively, the scores are listed in Table 2. The success rate of Table 2 comes from the left side of Figure 9, and the precision comes from the right side of Figure 9. As shown in Figure 9 and Table 2, compared to the other nine existing algorithms, the proposed tracker achieved 89.3 % precision and a 65.2% success rate in all the sequences, which is the highest and the best among all existing HSV-based trackers. In addition, compared to ProContEXT, the best existing hyperspectral video tracking algorithm, the success rate of this algorithm is improved by 0.2%. The ProContEXT algorithm uses both dynamic and static template techniques in the field of video target tracking to perform fine feature extraction on spatial and temporal contexts. It effectively captures the motion trajectory of the target and the dynamic changes in the surrounding environment. Compared to the SiamFC algorithm (based on a fully convolutional network), which is widely used in RGB trackers, our tracker achieved a significant improvement of 16.2% in success rate. Meanwhile, compared to DeepHKCF (based on a deep convolutional neural network), which is widely used in hyperspectral trackers, our tracker achieved an even higher success rate of 26.6%. These remarkable results clearly demonstrate the superior performance of capsule networks in our tracking system.

As can be clearly seen from the detailed data in Table 2, our method achieved a processing speed of 41.9 frames per second (FPS), which is not as high as that of traditional trackers such as SiamFC and CNHT, but it exhibited an average accuracy improvement of approximately 10%. This improvement is mainly due to the capsule network architecture we used. Capsule networks are able to handle spatial hierarchies and internal correlations in images more effectively through their unique dynamic routing mechanism when processing vector features. This improvement in feature extraction capability is directly reflected in the accuracy of our method. Compared to MixFormer, our method provided a significant improvement in accuracy while maintaining a higher FPS. The proposed algorithm achieved a 3.1% improvement in the success rate and a 6.3% increase in the precision value. Although MixFormer utilizes the advanced transformer architecture and demonstrates strong performance in terms of feature extraction, our method outperformed it in terms of overall performance. Notably, the advantage of our method is to accurately capture the features of the target in complex scenes and with dynamically changing targets, thus ensuring high tracking accuracy. This advantage stems from the unique strengths of capsule networks in processing vector features. In conclusion, while our method introduces increased complexity compared to traditional approaches, this resulted in substantial gains in accuracy. When compared to state-of-the-art RGB trackers like MixFormer, our method not only achieved higher accuracy but also preserved a high processing speed, highlighting its effectiveness and superiority in handling tracking challenges. While MixFormer leverages a mixed attention module for simultaneous feature extraction and target information integration, our method exploits spatial–spectral capsule networks to enhance

the separability and establishes relationships between different components and targets at various scales in hyperspectral target tracking.

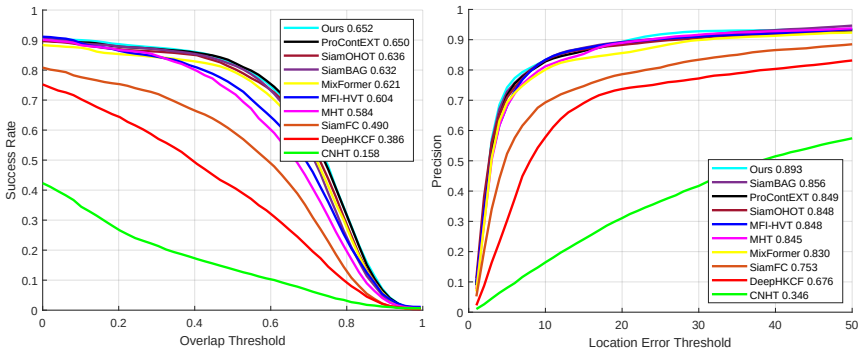

Figure 9. The success rate and precision rate of all sequences.

Table 2. The details comparing success rate and accuracy results in experiments.The best and second best values are highlighted in *red italics* and **blue bold**, respectively.

Methods	Ours	MFI-HVT	MHT	DeepHKCF	MixFormer	CNHT	ProContEXT	SiamFC	SiamBAG	SiamOHOT
Success Rate	*0.652*	0.604	0.584	0.386	0.621	0.158	**0.650**	0.490	0.632	0.636
Precision	*0.893*	0.848	0.845	0.676	0.830	0.346	0.849	0.753	**0.856**	0.848
FPS	41.9	6.3	7.6	80.1	25.5	126.7	27.6	189.0	46.7	36.8

To visually compare the performance of the proposed algorithm with the nine existing algorithms, Table 3 illustrates the performance of all algorithms against sequences with various challenges such as BC, DEF, FM, IPR, IV, LR, MB, OCC, OPR, OV, and SV on the HOT2023 dataset.

Table 3. AUC comparison on hyperspectral/false-color videos. The best and second best values are highlighted in *red italics* and **blue bold**, respectively.

Challenges	MFI-HVT	MHT	DeepHKCF	ProContEXT	CNHT	MixFormer	SiamFC	SiamBAG	SiamOHOT	Ours
BC	**0.655**	0.626	0.427	0.643	0.165	0.620	0.483	0.648	*0.700*	0.628
DEF	0.641	0.648	0.542	0.716	0.239	*0.753*	0.633	0.691	0.717	**0.730**
FM	0.601	0.567	0.378	**0.682**	0.182	0.644	0.632	0.615	0.562	*0.694*
IPR	0.695	0.689	0.550	0.710	0.239	**0.717**	0.590	0.703	*0.733*	0.714
IV	0.518	0.505	0.174	**0.572**	0.099	0.571	0.402	0.534	0.519	*0.632*
LR	0.518	0.426	0.189	0.547	0.031	**0.574**	0.471	0.583	0.500	*0.628*
MB	0.573	0.558	0.413	*0.703*	0.119	0.648	0.583	0.650	0.611	**0.683**
OCC	0.548	0.562	0.313	**0.650**	0.126	0.635	0.530	0.597	0.557	*0.661*
OPR	0.683	0.694	0.521	*0.713*	0.239	0.712	0.613	0.690	0.729	**0.718**
OV	0.616	0.624	0.314	**0.838**	0.136	*0.847*	0.458	0.635	0.584	0.821
SV	0.602	0.581	0.366	**0.627**	0.164	0.604	0.484	0.622	0.628	**0.660**

As different challenges pose different tracking difficulties, the proposed algorithm exhibited varying performance across different challenges. As indicated in Table 3, for a given challenge sequence, the success rate of the proposed algorithm ranked first in challenges like FM, IV, LR, and OCC. Additionally, the proposed algorithm secured the second position in terms of the success rate for challenges including DEF, MB, OPR, and SV. The main limitation of our algorithm lies in handling the BC and IPR challenges. When facing the challenge of BC, the presence of background objects that closely resemble the target can cause higher-level capsules in the network to misinterpret background features as parts of the target. This misidentification leads to tracking errors and degraded performance. Similarly, when the target in a video sequence undergoes IPR, its appearance and shape change significantly. As a result, the underlying capsule struggles to effectively extract fine-grained features. This makes it difficult to accurately localize the target based solely on the overall composition of its features.

4.4. Ablation Study

To validate the performance of each module in the algorithm and the effectiveness of the spectral information in the tracker, we performed a series of ablation experiments using the HOT2023 dataset. These experiments were designed to compare the difference in performance when specific modules in the tracker are enabled and disabled. The results of the experiments are shown in Table 4: (i) It is evident that when all three modules of this algorithm were used, the success rate reached 66.1%, and the precision reached 86.6%. Table 4: (ii) indicates that when the confidence threshold judgement module was removed and no update was performed, the success rate decreased by 6.3%, and the precision decreased by 14.3%. There are not many cases of target loss due to the capsule network; however, this indicates that the update module is useful for this algorithm. Also, (iii) illustrates the case when instead of the spatial–spectral capsule network, a normal capsule network was used to capture the relationship between spatial features. As is evident, the spectral octave convolution together with the confidence threshold judgement module can also be useful, with a decrease of only 3.9% compared to (i). The (iv) denotes the case when only the spatial–spectral octave convolution module was used and the other parts were tracked using the detection-based capsule network framework. It can be seen that the success rate reached 43.8%, and the precision decreased by 20.3%. Although the success rate is not great, the whole process ran faster than before at 65FPS. The (v) represents the case when the spatial–spectral octave convolution module was not used. As is illustrated, the resulting success rate of 59.1%, which is 7% lower than (i), indicates that the spatial–spectral octave convolution module is beneficial for the algorithm. (vi) illustrates that when only the spatial–spectral capsule network is used, the success rate reached 56.6%, which shows that it is important to distinguish between the different capsule weights when compared to (iv). (vii) demonstrates the case when the confidence judgement module was used to update the template in the detection-based capsule network framework. As can be observed, the success rate was only 33.3% because of fewer cases of missing targets, but this may also be influenced by other factors. In order to verify the effectiveness of the spectral information in improving the performance of the system, we designed and carried out a comparative experiment with two situations (i, vii): either the first two modules responsible for processing the spectral information were not activated, or they were activated simultaneously. The analysis of the experimental data shows that the accuracy of the tracking system was significantly improved when the two spectral information processing modules were activated, with a 40.1% increase in accuracy compared to the nonactivated state. This improvement is not only reflected in the overall performance index of the system but also significantly increases the robustness and adaptability of the system in dealing with complex environmental challenges (e.g., IV, OCC, etc.). In summary, combining the three modules for tracking presents the best results for the occlusion challenge.

Table 4. Ablation study on HOT2023. ✓indicates that the module is enabled, and × indicates that the module is not enabled.

Variations	Spatial-Spectral Octave	Spatial-Spectral Caps	Judgment	Success Rate	Precision
i	✓	✓	✓	0.661	0.866
ii	✓	✓	×	0.598	0.723
iii	✓	×	✓	0.622	0.778
iv	✓	×	×	0.438	0.663
v	×	✓	✓	0.591	0.772
vi	×	✓	×	0.566	0.657
vii	×	×	✓	0.333	0.465

5. Conclusions

This research proposes a hyperspectral video target tracker based on octave convolution and a spatial–spectral capsule network. The spatial–spectral octave convolution module reduces the amount of image frame data. The approach enhances the spatial and spectral information, facilitating spectral feature extraction. Moreover, the spatial–spectral capsule network ensures effective handling of target occlusion and scale issues. Finally, a confidence threshold judgment module is introduced, effectively mitigating the impact of target loss on subsequent tracking.

Extensive experimental evaluations demonstrate that the proposed method achieved a precision of 89.3% and a success rate of 65.2%. A detailed analysis of the experimental data reveals that our approach maintained strong performance even in complex tracking scenarios, demonstrating high accuracy under occlusion challenges. Notably, our method outperformed the state-of-the-art hyperspectral tracker, SiamOHOT, by 10.4% for the OCC challenge. However, it is worth mentioning that when facing BC and IPR challenges, our algorithm lagged behind SiamOHOT by 7.2% and 1.9%, respectively.

To address the aforementioned challenges, we propose several future directions for improvement and extension. First and foremost, we will explore adaptive background modeling methods, leveraging the joint spatial and spectral information to distinguish between the target and the background, thereby reducing the interference of background clutter on tracking performance. Moreover, we will continue to optimize and refine the capsule network by improving its architecture, enhancing feature extraction capabilities, and exploring more robust training strategies to strengthen its adaptability to complex scenarios. Last but not least, to broaden the applicability of our algorithm, we plan to extend the wavelength range of the datasets examined and investigate its potential applications in the near-infrared hyperspectral domain. Through these advancements, we expect to significantly increase the accuracy, robustness, and practical usability of our algorithm, making it more effective across a broader range of real-world applications.

Author Contributions: Conceptualization, D.Z. and M.W; methodology, M.W.; software, K.H.; validation, M.W. and K.H.; formal analysis, M.W., K.H., and W.Z.; investigation, M.W. and W.Z.; resources, M.W.; data curation, M.W. and K.H.; writing—original draft preparation, M.W.; writing—review and editing, D.Z., W.Z., Y.L., and P.V.A.; visualization, M.W. and K.H.; supervision, D.Z., L.W., Y.L., and H.Z.; project administration, D.Z., P.V.A., and Y.A.; funding acquisition, D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant (62105258), in part by the 111 Project under Grant (B17035), in part by the Wuxi Innovation and Entrepreneurship Fund "Taihu Light" Science and Technology (Fundamental Research) Project under Grant (K20221046, K20241045), and in part by the JSPS KAKENHI under Grant (JP23K16911).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Islam, M.A.; Xing, W.; Zhou, J.; Gao, Y.; Paliwal, K.K. Hy-Tracker: A Novel Framework for Enhancing Efficiency and Accuracy of Object Tracking in Hyperspectral Videos. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5521514. [CrossRef]
2. Li, Z.; Xiong, F.; Zhou, J.; Lu, J.; Qian, Y. Learning a Deep Ensemble Network with Band Importance for Hyperspectral Object Tracking. *IEEE Trans. Image Process.* **2023**, *32*, 2901–2914. [CrossRef] [PubMed]
3. Aleissaee, A.A.; Kumar, A.; Anwer, R.M.; Khan, S.; Cholakkal, H.; Xia, G.S.; Khan, F.S. Transformers in remote sensing: A survey. *Remote Sens.* **2023**, *15*, 1860. [CrossRef]
4. Zhao, D.; Zhu, X.; Zhang, Z.; Arun, P.V.; Cao, J.; Wang, Q.; Zhou, H.; Jiang, H.; Hu, J.; Qian, K. Hyperspectral video target tracking based on pixel-wise spectral matching reduction and deep spectral cascading texture features. *Signal Process.* **2023**, *209*, 109033. [CrossRef]

5. Yan, J.; Zhang, K.; Sun, Q.; Ge, C.; Wan, W.; Sun, J.; Zhang, H. Spatial–spectral unfolding network with mutual guidance for multispectral and hyperspectral image fusion. *Pattern Recognit.* **2025**, *161*, 111277. [CrossRef]
6. Zhang, Z.; Hu, B.; Wang, M.; Arun, P.V.; Zhao, D.; Zhu, X.; Hu, J.; Li, H.; Zhou, H.; Qian, K. Hyperspectral video tracker based on spectral deviation reduction and a double siamese network. *Remote Sens.* **2023**, *15*, 1579. [CrossRef]
7. Wang, Y.; Liu, Y.; Ma, M.; Mei, S. A spectral–spatial transformer fusion method for hyperspectral video tracking. *Remote Sens.* **2023**, *15*, 1735. [CrossRef]
8. Li, W.; Hou, Z.; Zhou, J.; Tao, R. SiamBAG: Band attention grouping-based Siamese object tracking network for hyperspectral videos. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5514712. [CrossRef]
9. Zhao, D.; Cao, J.; Zhu, X.; Zhang, Z.; Arun, P.V.; Guo, Y.; Qian, K.; Zhang, L.; Zhou, H.; Hu, J. Hyperspectral Video Target Tracking Based on Deep Edge Convolution Feature and Improved Context Filter. *Remote Sens.* **2022**, *14*, 6219. [CrossRef]
10. Chen, L.; Zhao, Y.; Kong, S.G. Sfa-guided mosaic transformer for tracking small objects in snapshot spectral imaging. *ISPRS J. Photogramm. Remote Sens.* **2023**, *204*, 223–236. [CrossRef]
11. Zhang, K.; Yan, J.; Zhang, F.; Ge, C.; Wan, W.; Sun, J.; Zhang, H. Spectral-Spatial Dual Graph Unfolding Network for Multispectral and Hyperspectral Image Fusion. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5508718. [CrossRef]
12. Zhao, D.; Yan, W.; You, M.; Zhang, J.; Arun, P.V.; Jiao, C.; Wang, Q.; Zhou, H. Hyperspectral Anomaly Detection Based on Empirical Mode Decomposition and Local Weighted Contrast. *IEEE Sens. J.* **2024**, *24*, 33847–33861. [CrossRef]
13. Zhang, R.; Jin, S.; Zhang, Y.; Zang, J.; Wang, Y.; Li, Q.; Sun, Z.; Wang, X.; Zhou, Q.; Cai, J.; et al. PhenoNet: A two-stage lightweight deep learning framework for real-time wheat phenophase classification. *ISPRS J. Photogramm. Remote Sens.* **2024**, *208*, 136–157. [CrossRef]
14. Avola, G.; Matese, A.; Riggi, E. An Overview of the Special Issue on "Precision Agriculture Using Hyperspectral Images". *Remote Sens.* **2023**, *15*, 1917. [CrossRef]
15. Dong, S.; Wang, L.; Du, B.; Meng, X. ChangeCLIP: Remote sensing change detection with multimodal vision-language representation learning. *ISPRS J. Photogramm. Remote Sens.* **2024**, *208*, 53–69. [CrossRef]
16. Zhao, Y.; Tian, Y.; Lei, S.; Li, Y.; Hua, X.; Guo, D.; Ji, C. A Comprehensive Correction Method for Radiation Distortion of Multi-Strip Airborne Hyperspectral Images. *Remote Sens.* **2023**, *15*, 1828. [CrossRef]
17. Zhang, Z.; Jiang, D.; Chang, Q.; Zheng, Z.; Fu, X.; Li, K.; Mo, H. Estimation of anthocyanins in leaves of trees with apple mosaic disease based on hyperspectral data. *Remote Sens.* **2023**, *15*, 1732. [CrossRef]
18. Wang, S.; Qian, K.; Shen, J.; Ma, H.; Chen, P. AD-SiamRPN: Anti-Deformation Object Tracking via an Improved Siamese Region Proposal Network on Hyperspectral Videos. *Remote Sens.* **2023**, *15*, 1731. [CrossRef]
19. Ahmed, S.A.; Mohsin, M.; Ali, S.M.Z. Survey and technological analysis of laser and its defense applications. *Def. Technol.* **2021**, *17*, 583–592. [CrossRef]
20. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [CrossRef]
21. Liang, L.; Zhang, S.; Li, J.; Plaza, A.; Cui, Z. Multi-Scale Spectral-Spatial Attention Network for Hyperspectral Image Classification Combining 2D Octave and 3D Convolutional Neural Networks. *Remote Sens.* **2023**, *15*, 1758. [CrossRef]
22. Li, N.; Zhao, Y.; Pan, Q.; Kong, S.G.; Chan, J.C.W. Illumination-invariant road detection and tracking using LWIR polarization characteristics. *ISPRS J. Photogramm. Remote Sens.* **2021**, *180*, 357–369. [CrossRef]
23. Liu, S.; Li, H.; Wang, F.; Chen, J.; Zhang, G.; Song, L.; Hu, B. Unsupervised transformer boundary autoencoder network for hyperspectral image change detection. *Remote Sens.* **2023**, *15*, 1868. [CrossRef]
24. Li, Y.; Zhao, W.; Cambria, E.; Wang, S.; Eger, S. Graph routing between capsules. *Neural Netw.* **2021**, *143*, 345–354. [CrossRef] [PubMed]
25. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.; Li, J.; Pla, F. Capsule networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2145–2160. [CrossRef]
26. Mei, Z.; Yin, Z.; Kong, X.; Wang, L.; Ren, H. Cascade residual capsule network for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 3089–3106. [CrossRef]
27. Lyu, Y.; Yang, M.Y.; Vosselman, G.; Xia, G.S. Video object detection with a convolutional regression tracker. *ISPRS J. Photogramm. Remote Sens.* **2021**, *176*, 139–150. [CrossRef]
28. Zhu, F.; Cui, J.; Dou, K. Spatio-temporal hierarchical feature transformer for UAV object tracking. *ISPRS J. Photogramm. Remote Sens.* **2023**, *204*, 442–452. [CrossRef]
29. Voigtlaender, P.; Luiten, J.; Torr, P.H.; Leibe, B. Siam r-cnn: Visual tracking by re-detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6578–6588.
30. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
31. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. Atom: Accurate tracking by overlap maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4660–4669.

32. Wei, X.; Bai, Y.; Zheng, Y.; Shi, D.; Gong, Y. Autoregressive Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 9697–9706.
33. Tumanyan, N.; Singer, A.; Bagon, S.; Dekel, T. DINO-Tracker: Taming DINO for Self-Supervised Point Tracking in a Single Video. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; pp. 367–385.
34. Qian, K.; Zhou, J.; Xiong, F.; Zhou, H.; Du, J. Object tracking in hyperspectral videos with convolutional features and kernelized correlation filter. In Proceedings of the Smart Multimedia, Toulon, France, August 24–26, 2018; pp. 308–319.
35. Xiong, F.; Zhou, J.; Qian, Y. Material based object tracking in hyperspectral videos. *IEEE Trans. Image Process.* **2020**, *29*, 3719–3733. [CrossRef]
36. Hou, Z.; Li, W.; Zhou, J.; Tao, R. Spatial–spectral weighted and regularized tensor sparse correlation filter for object tracking in hyperspectral videos. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5541012. [CrossRef]
37. Uzkent, B.; Rangnekar, A.; Hoffman, M.J. Tracking in aerial hyperspectral videos using deep kernelized correlation filters. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 449–461. [CrossRef]
38. Zhang, Z.; Zhu, X.; Zhao, D.; Arun, P.V.; Zhou, H.; Qian, K.; Hu, J. Hyperspectral video target tracking based on deep features with spectral matching reduction and adaptive scale 3d hog features. *Remote Sens.* **2022**, *14*, 5958. [CrossRef]
39. Tang, Y.; Huang, H.; Liu, Y.; Li, Y. A Siamese network-based tracking framework for hyperspectral video. *Neural Comput. Appl.* **2023**, *35*, 2381–2397. [CrossRef]
40. Liu, Z.; Wang, X.; Zhong, Y.; Shu, M.; Sun, C. SiamHYPER: Learning a hyperspectral object tracker from an RGB-based tracker. *IEEE Trans. Image Process.* **2022**, *31*, 7116–7129. [CrossRef]
41. Chen, Y.; Yuan, Q.; Tang, Y.; Xiao, Y.; He, J.; Liu, Z. SENSE: Hyperspectral Video Object Tracker via Fusing Material and Motion Cues. *Infin. Fusion* **2024**, *109*, 1566–2535. [CrossRef]
42. Zhao, D.; Zhang, H.; Huang, K.; Zhu, X.; Arun, P.V.; Jiang, W.; Li, S.; Pei, X.; Zhou, H. SASU-Net: Hyperspectral Video Tracker Based on Spectral Adaptive Aggregation Weighting and Scale Updating. *Expert Syst. Appl.* **2025**, *272*, 126721. [CrossRef]
43. Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Yan, S.; Feng, J. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3435–3444.
44. Xu, Q.; Xiao, Y.; Wang, D.; Luo, B. CSA-MSO3DCNN: Multiscale octave 3D CNN with channel and spatial attention for hyperspectral image classification. *Remote Sens.* **2020**, *12*, 188. [CrossRef]
45. Xu, Q.; Wang, D.; Luo, B. Faster multiscale capsule network with octave convolution for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 361–365. [CrossRef]
46. Kan, Z.; Li, S.; Hou, M.; Fang, L.; Zhang, Y. Attention-based octave network for hyperspectral image denoising. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 1089–1102. [CrossRef]
47. Tang, X.; Meng, F.; Zhang, X.; Cheung, Y.M.; Ma, J.; Liu, F.; Jiao, L. Hyperspectral image classification based on 3-D octave convolution with spatial–spectral attention network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 2430–2447. [CrossRef]
48. Papoutsis, I.; Bountos, N.I.; Zavras, A.; Michail, D.; Tryfonopoulos, C. Benchmarking and scaling of deep learning models for land cover image classification. *ISPRS J. Photogramm. Remote Sens.* **2023**, *195*, 250–268. [CrossRef]
49. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
50. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–19.
51. Wang, Q.; Teng, Z.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Learning attentions: Residual attentional siamese network for high performance online visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4854–4863.
52. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6728–6737.
53. Zhu, X.; Zhang, H.; Hu, B.; Huang, K.; Arun, P.V.; Jia, X.; Zhao, D.; Wang, Q.; Zhou, H.; Yang, S. DSP-Net: A dynamic spectral-spatial joint perception network for hyperspectral target tracking. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5510905. [CrossRef]
54. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
55. Zhang, Z.; Qian, K.; Du, J.; Zhou, H. Multi-features integration based hyperspectral videos tracker. In Proceedings of the Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing, Amsterdam, The Netherlands, 24–26 March 2021; pp. 1–5.
56. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. Mixformer: End-to-end tracking with iterative mixed attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618.

57. Lan, J.P.; Cheng, Z.Q.; He, J.Y.; Li, C.; Luo, B.; Bao, X.; Xiang, W.; Geng, Y.; Xie, X. Procontext: Exploring progressive context transformer for tracking. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
58. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016; pp. 850–865.
59. Sun, C.; Wang, X.; Liu, Z.; Wan, Y.; Zhang, L.; Zhong, Y. Siamohot: A lightweight dual siamese network for onboard hyperspectral object tracking via joint spatial-spectral knowledge distillation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5521112. [CrossRef]

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.