

Article

Visual Tracking of Materials and Astronaut Operation Action Recognition in Space Station Cargo Spacecraft Cabins

Jianguo Sheng ^{1,2}, Liang Chang ^{2,3,*} and Zhang Zhang ^{2,3}

¹ College of Information Technology, Shanghai Ocean University, Shanghai 201203, China; sjg2069125@163.com

² Innovation Academy for Microsatellites of CAS, Shanghai 201203, China; zhangz@microstate.com

³ Shanghai Engineering Center for Microsatellites, Shanghai 201203, China

* Correspondence: changl@microstate.com

Abstract

The automatic recognition of cargo pick-and-place actions in spacecraft cabins is important for improving the intelligence level of in-orbit cargo management. This study proposes an integrated visual recognition framework based on improved YOLOv11n, DeepSORT tracking, and SVM-based trajectory classification. To address the challenges of densely stacked cargo, local occlusion, and complex cabin backgrounds, WeightConv and the convolution and attention fusion module (CAFM) are introduced into the YOLOv11n detector to enhance cargo feature representation. Based on the detection results, DeepSORT is used to associate cargo targets across video frames and extract continuous motion trajectories. The trajectory descriptors are then classified by an SVM into three typical operation states: “pick,” “place,” and “not picked.” On the self-constructed cabin dataset, the improved detector obtains a precision of 97.3%. The tracking module obtains an MOTA of 97.1% and an IDF1 of 95.1%, while the trajectory-based SVM achieves an overall classification accuracy of 92.0%. Experimental results demonstrate that the proposed framework provides reliable visual evidence for automating the recording of typical cargo pick-and-place operations in spacecraft cabins, offering viable technical support for enhancing in-orbit cargo management efficiency.

Keywords: cargo spacecraft; cargo management; action recognition; YOLO; DeepSORT; SVM

1. Introduction

With China’s space station moving into regular application and operation, the range and quantity of transported supplies have continued to grow. Cargo spacecraft are the main carriers for transporting materials to and from the station [1], and their increasingly frequent resupply tasks require more detailed records of cabin cargo transfer. RFID has been introduced as an auxiliary tool for cargo management, but it mainly returns static tag information [2]. It cannot describe the continuous motion of a cargo item or determine whether an astronaut has picked up, placed, or only locally adjusted the item. Consequently, an RFID-only management mode is insufficient for generating complete real-time operation logs or supporting rapid cargo localization during missions. A vision-based perception system is therefore needed to continuously track cabin cargo and recog-

Academic Editor: Stefano Silvestrini

Received: 31 May 2026

Revised: 20 June 2026

Accepted: 29 June 2026

Published: 30 June 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

nize pick-and-place operations, thereby improving the intelligence of spacecraft cargo management.

Such a perception system must be designed with both recognition performance and onboard resource limitations in mind. Studies on intelligent aerospace control emphasize that future spacecraft will depend more strongly on environmental perception, autonomous decision-making, and adaptive operation, while computation, communication, and energy remain tightly constrained [3]. Similar engineering constraints appear in small-satellite subsystem design. Khan et al., for example, optimized a printed magnetorquer for a 3-U nanosatellite by jointly considering size, power dissipation, thermal feasibility, low mass, and limited installation space [4]. These considerations indicate that cabin cargo perception should pursue accurate recognition without sacrificing deployability. In this work, CAFM and WeightConv are introduced to improve detection in occluded and densely stacked scenes while preserving the lightweight character of the detector.

Ground-based vision methods also face a domain shift when applied inside spacecraft cabins. Compared with warehouses or retail terminals, the cabin scene has tighter computing margins, irregular cargo stacking, local occlusions, complex background textures, variable illumination, and limited real-space training data. These factors make direct transfer of ground-oriented detection and tracking pipelines unreliable without task-specific adaptation.

The main contributions of this study are summarized as follows.

1. A lightweight YOLOv11n-based cargo detection network is constructed. By introducing CAFM and WeightConv for lightweight feature optimization, the proposed detector enhances cargo feature perception under dense stacking, occlusion, and complex cabin backgrounds while maintaining low computational cost.
2. A complete detection-tracking-recognition framework for cabin cargo operation analysis is established. The improved detector is combined with DeepSORT to extract continuous cargo trajectories from video sequences, and an SVM is further used to analyze trajectory features and classify three typical operation states: pick, place, and not picked.

2. Related Work

Research on cargo management within spacecraft cabins is still in its early stage. In contrast, computer vision has been widely applied in ground-based scenarios such as intelligent warehousing and unmanned retail, providing useful technical references for intelligent cargo management in space environments.

In the field of intelligent warehousing, Xie et al. [5] proposed a lightweight fusion perception model based on improved YOLOv5 and DeepSORT that effectively overcame target occlusion and identity confusion in mixed human-cargo scenarios, achieving a multi-object tracking accuracy (MOTA) of 75.8%. The model was primarily applied to real-time assessment of warehouse zone occupancy and dynamic management of inventory circulation. To address the degradation of recognition accuracy caused by illumination variation and target occlusion in retail terminal scenarios, Ma Jingang [6] developed an intelligent pick-up recognition framework for ice cream products. Through adaptive improvements to YOLOv5 and DeepSORT, this study significantly mitigated the problems of missed detections and identity switches, achieving a product pick-up recognition accuracy of 82% on a self-constructed dataset, and enabling automated statistics on product categories and sales volumes. Zhai Jiankun [7] designed a recognition system for smart cabinet scenarios based on SSD detection and InceptionV3 classification, aiming to address the high cost associated with traditional RFID solutions. The system employed the SSD algorithm to detect hand positions for inferring interaction actions combined with the InceptionV3 network to classify product images captured from regions of interest

(ROIs). Results showed that the system achieved a recognition accuracy of 74% in single-item pick-up scenarios and an overall scene recognition rate of 64%.

Furthermore, this “detection-tracking” technical paradigm has also demonstrated practical applicability in dynamic biological perception. Zhang Zhedong et al. [8] proposed a freshwater fish behavior detection algorithm based on YOLOX and DeepSORT. The method enabled freshwater fish detection, tracking, motion-feature extraction, and random forest-based behavior classification for foraging, burst swimming, and stationary states. Results showed a detection accuracy of 83%, a multi-object tracking accuracy of 84%, and a behavior classification accuracy of 99%. Wang Meili et al. [9] proposed an improved BoT-SORT algorithm integrating depth information and motion trends. This method effectively enhanced sheep tracking association in densely occluded scenes, achieving a 1.7% improvement over the baseline. Tu et al. [10] addressed the long-term monitoring needs of swine farming by proposing an automatic behavior analysis algorithm based on YOLOv8 and OC-SORT. The method can record the duration of four behavioral categories—lying, standing, feeding, and other—for each individual pig. Experimental results demonstrated a multi-object tracking accuracy of 99.7% on a 60 min long-sequence dataset, supporting automated assessment of pig health status.

Although the above studies demonstrate the effectiveness of detection-tracking-based visual perception in ground-based and biological scenarios, spacecraft environments introduce additional domain shifts and deployment constraints. Recent studies on space visual perception have shown that vision-based recognition and tracking in orbital environments are affected by illumination variation, specular reflection, sensor noise, limited real-space data, and onboard resource constraints. Feng et al. [11] investigated online test-time adaptation for in-orbit noncooperative spacecraft pose estimation, showing that source-target domain gaps can degrade the performance of vision models when they are deployed in unseen orbital scenarios. Another self-distillation adaptive pose estimation method [12] further demonstrated that texture, lighting, and noise discrepancies between synthetic and real orbital images require domain-invariant representations and spatial-contextual feature learning. In addition, Zhou et al. [13] studied active visual tracking for space non-cooperative objects and indicated that complex relative motion, limited field of view, and low-resolution onboard sensing make robust target perception particularly challenging in space scenarios.

Although these studies provide important references for space visual perception, most of them focus on spacecraft pose estimation, external non-cooperative object tracking, or general target recognition. For in-cabin cargo management, the perception task further involves dense cargo stacking, local occlusion, complex cabin backgrounds, and fine-grained pick-and-place behavior recognition, which remain insufficiently explored in existing studies.

Closer to the spacecraft cabin scenario, several studies have explored in-cabin visual perception and target recognition. Zhang Rui et al. [14] proposed a human visual tracking algorithm combining deep learning and probabilistic models for in-cabin follower service robots aboard space stations. The method enables stable tracking of humans with diverse clothing and arbitrary poses, providing support for autonomous following by in-cabin service robots. To address visual information degradation caused by uneven illumination and excessive darkness in in-orbit environments, Zhang et al. [15] proposed an unsupervised image enhancement framework named SpaceLight. The reported results showed better enhancement performance than several existing low-light methods on real in-orbit imagery. Cao Zhe et al. [16] constructed an in-cabin target recognition model for space stations based on Darknet-53. Using real video footage from the International Space Station, the model achieved a bounding-box localization accuracy of 95% and a category recognition accuracy of 79%, validating the applicability of deep learning in microgravity

environments. Yuan et al. [17] proposed PBRA-YOLO based on YOLO11 for tool management scenarios within cargo spacecraft cabins. This model addresses detection degradation caused by large tool size variations and uncontrolled stacking in microgravity environments. The mean average precision on a self-constructed dataset improved by 4.2 percentage points over the baseline, validating its effectiveness in complex spatial stacking scenarios.

In summary, existing research has laid a solid theoretical and technical foundation in object detection, multi-object tracking, and action recognition under general scenarios, providing valuable references for this study. However, the interior of cargo spacecraft cabins presents unique challenges, including densely stacked materials, severe local occlusion, and complex background interference. At the same time, the computational resources of in-orbit edge devices are extremely constrained, making it difficult for existing general-purpose algorithms to strike a balance between robust feature extraction and lightweight deployment. Moreover, research specifically targeting spacecraft cabin scenarios remains scarce, with few solutions capable of achieving integrated perception that combines cabin material detection, inter-frame trajectory tracking, and pick-and-place action recognition. To address these limitations, this paper proposes a lightweight intelligent visual perception algorithm for material management inside cargo spacecraft cabins, adopting a cascaded “detection–tracking–classification” framework to enable automated recognition of astronauts’ pick-and-place actions. The overall technical pipeline of this study is illustrated in Figure 1.

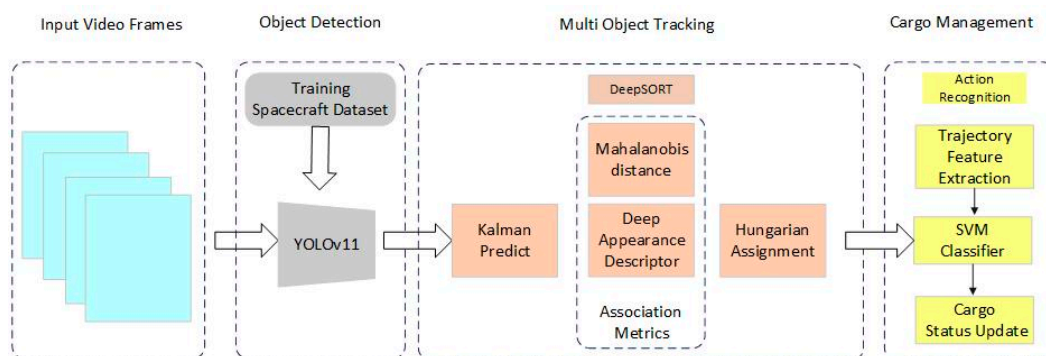


Figure 1. Overall technical pipeline.

3. The Proposed Method

This paper follows a tracking-by-detection (TBD) workflow [18] and processes the video stream at the frame level. The detector first identifies cargo objects and outputs their bounding boxes in each frame. The tracking stage then links these detections over time by combining motion prediction with data association so that continuous object trajectories can be formed for later action analysis.

3.1. YOLO Object Detection Algorithm Design

For cabin cargo perception, the detector must process images quickly while preserving sufficient localization accuracy [19]. Proposal-based detectors such as Faster R-CNN [20] refine candidate regions before classification, which improves localization, but add extra inference steps. Direct-regression detectors such as YOLO [21] and SSD [22] predict classes and bounding boxes in one forward pass, making them more suitable for real-time and resource-constrained applications.

During the in-orbit missions of cargo spacecraft, edge devices have limited computational resources. Although two-stage algorithms offer high accuracy, their substantial computational overhead makes deployment difficult. In contrast, one-stage algorithms,

while maintaining competitive accuracy, better meet the requirements of in-orbit cargo spacecraft applications in terms of computational consumption and real-time performance. YOLO models are frequently used in real-time detection tasks because they provide a practical trade-off between accuracy and speed [23]. Therefore, this paper selects YOLOv11n as the baseline detection model.

However, as a lightweight detection model, YOLOv11n still exhibits limitations in feature representation and extraction when confronted with occlusion caused by densely stacked cargo inside the spacecraft cabin and small targets against complex backgrounds. Therefore, to enhance the robustness of the algorithm in the cabin environment, two improvement strategies are proposed without significantly increasing computational resource consumption. The improved algorithm architecture is shown in Figure 2.

1. Introduction of the convolution and attention fusion module (CAFM) [24]: This module leverages the local perception advantage of CNNs and the global modeling capability of transformers to effectively capture long-range dependencies, thereby addressing the problem of feature loss in occluded environments.
2. Adoption of the weighted convolution operator (WeightConv) [25]: By dynamically adjusting feature weights through a spatial density function, this operator enhances the model's extraction of key information at target centers while suppressing background interference within the cabin.

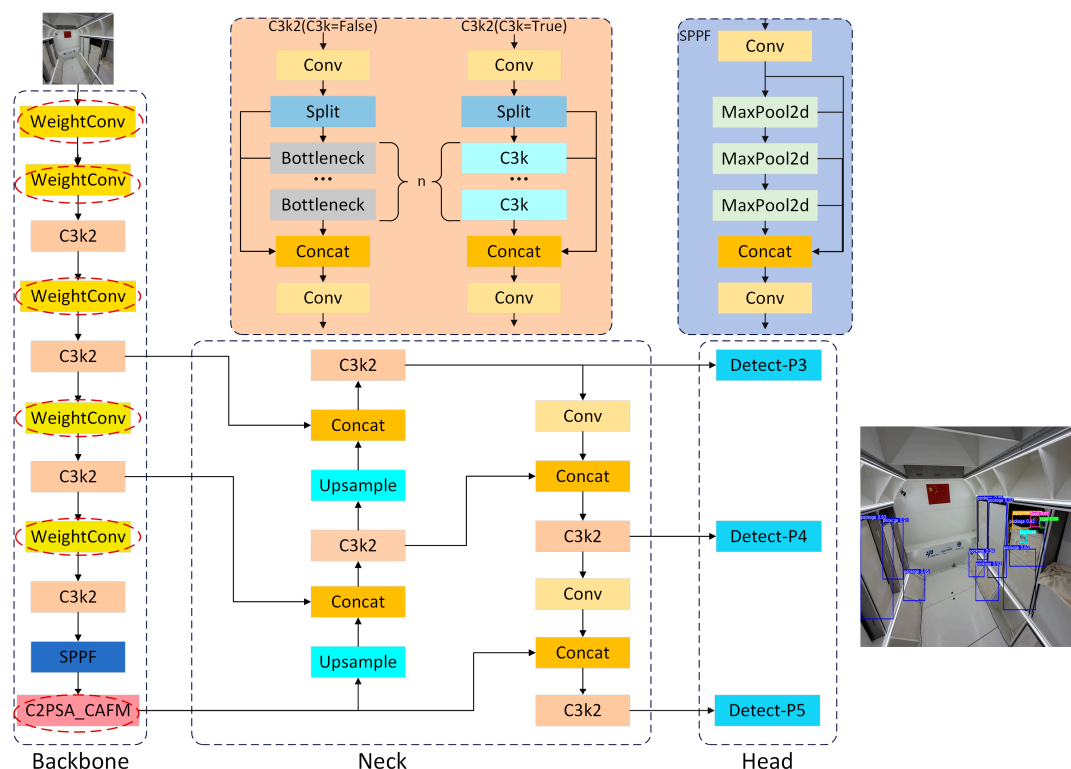


Figure 2. Improved YOLOv11 network architecture.

3.1.1. CAFM

During in-orbit missions of cargo spacecraft, the cargo inside the cabin is densely arranged against a complex background. The original YOLOv11 network employs a C2PSA module based on the attention mechanism, aimed at enhancing the model's perception of global information. However, when processing complex textures, a single attention mechanism tends to overfocus on global context at the expense of local structural information, leading to the loss of edge features for small or occluded targets.

To address this issue, this paper reconstructs the original C2PSA module and introduces the CAFM. This module adopts a dual-branch parallel structure combining convolution and attention, enabling simultaneous modeling of both global and local features to achieve feature complementarity and fusion. Its structure is illustrated in Figure 3.

The CAFM is designed to combine the local-detail extraction capability of convolution with the long-range dependency modeling ability of attention [24]. This design is consistent with the requirements of cabin cargo detection, where convolution helps preserve local structural details, while attention captures broader contextual relationships. In the adopted attention branch, a compact channel-wise attention map is computed after reshaping the query and key tensors, rather than constructing a full spatial attention map. For a feature map of size $\hat{H} \times \hat{W}$ and channel number $\hat{C}$, standard spatial self-attention requires an attention matrix of size $R^{\hat{H}\hat{W} \times \hat{H}\hat{W}}$, whereas the adopted channel-wise reshaping strategy forms a compact matrix of size $R^{\hat{C} \times \hat{C}}$. This structural design reduces the theoretical memory and matrix-multiplication burden of the attention branch while retaining long-range feature modeling ability.

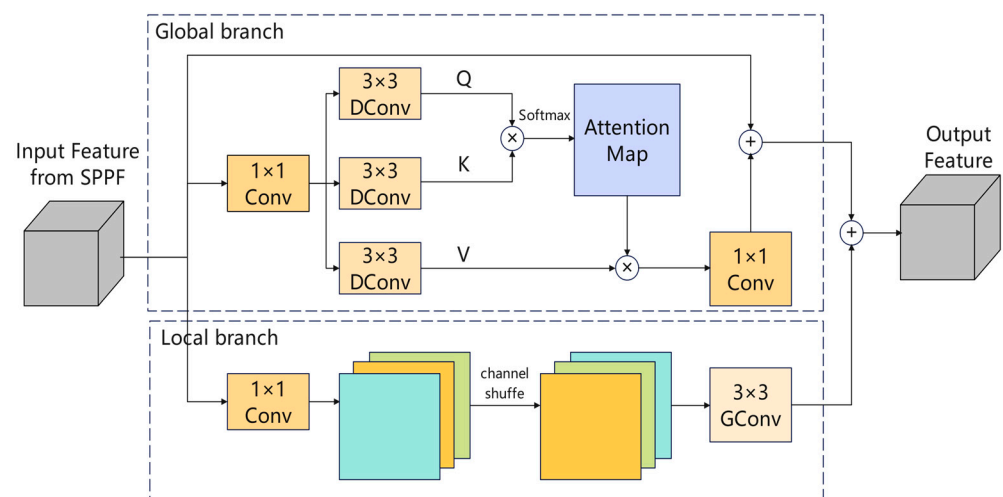


Figure 3. Architecture of the CAFM.

The global branch captures long-range image context through an attention operation. This branch performs three independent transformations on the input features via 1×1 convolutions and 3×3 depthwise convolutions to extract queries (Q), keys (K), and values (V), each yielding a tensor of dimensionality $\hat{H} \times \hat{W} \times \hat{C}$. Rather than computing a full spatial attention map of size $R^{\hat{H}\hat{W} \times \hat{H}\hat{W}}$, which incurs prohibitive computational cost in conventional transformers, this module introduces a dimension reshaping strategy to substantially reduce complexity. Specifically, Q is reorganized into $\hat{Q} \in R^{\hat{H}\hat{W} \times \hat{C}}$ and K is transposed into $\hat{K} \in R^{\hat{C} \times \hat{H}\hat{W}}$, enabling the computation of a compact channel-wise attention map $A \in R^{\hat{C} \times \hat{C}}$. The attention weights are normalized via softmax and subsequently used to reweight V , with the final output obtained through a residual connection. The global-branch operation is formulated as follows:

$$F_{att} = W_{1 \times 1} \text{Attention}(\hat{Q}, \hat{K}, V) + Y \quad (1)$$

$$\text{Attention}(\hat{Q}, \hat{K}, V) = V \cdot \text{Softmax}\left(\frac{\hat{K}\hat{Q}}{\alpha}\right) \quad (2)$$

where α is a learnable scaling parameter applied before the softmax function to control the magnitude of the $\hat{K}\hat{Q}$ matrix multiplication. Y is the input feature, and $W_{1 \times 1}$ denotes the 1×1 convolution operation.

The local branch complements the attention branch by emphasizing fine spatial structures and edge cues. It first applies a 1×1 convolution to recalibrate the channel dimensionality, then partitions the features into subgroups where depthwise separable convolutions are independently applied to enhance cross-channel feature diversity, and the subgroup outputs are subsequently merged and reorganized along the channel axis. Finally, grouped convolutions are employed to extract the rearranged features. The computational process of the local branch can be formulated as:

$$F_{Conv} = GW_{3 \times 3}(\text{Shuffle}(W_{1 \times 1}(Y))) \quad (3)$$

where Y denotes the input feature, $\text{Shuffle}(\cdot)$ represents the channel shuffling operation, and $GW_{3 \times 3}$ denotes the 3×3 grouped convolution operation.

Finally, the outputs of the local branch and the global branch are summed to yield the final output of the CAFM module:

$$F_{out} = F_{att} + F_{Conv} \quad (4)$$

3.1.2. WeightConv

The interior of cargo spacecraft cabins features complex backgrounds. The convolutional kernels of traditional convolutional networks are treated as uniformly weighted sliding windows, assigning equal weights to all pixels within the window during feature extraction. This approach ignores the spatial structural differences within local image neighborhoods, making it prone to introducing background interference and causing the features of densely packed small targets to become blurred. To address this limitation, this paper introduces a weighted convolution operator into the backbone network of YOLOv11 to replace the original standard convolution. It is a convolution optimization method based on a spatial density function, which constructs a spatial weight matrix to reshape the focus of feature extraction, strengthening the model's attention to key features at target centers while suppressing background noise.

This operator first defines a one-dimensional density vector α , and uses the outer product of the vector α to generate a two-dimensional spatial density matrix Φ :

$$\Phi = \alpha \otimes \alpha^T \quad (5)$$

The generated spatial density matrix Φ is then element-wise multiplied with the original learnable convolution weights W to obtain the new weighted convolution kernel $W_{weighted}$:

$$W_{weighted} = W \odot \Phi \quad (6)$$

where the shape of Φ is consistent with the convolution kernel size.

Finally, the weighted convolution kernel $W_{weighted}$ is applied to the input feature map X to perform convolution and complete feature extraction:

$$Y = W_{weighted}(X) \quad (7)$$

The WeightConv operator introduces a spatial density function into the convolution process so that pixels at different relative positions within the local receptive field contribute with different prior weights [25]. Compared with standard convolution, this operation does not add extra learnable weights. Since the spatial density matrix is fixed after construction, it can be pre-computed before inference and fused with the convolution kernel during implementation, resulting in limited additional runtime overhead. In the spacecraft cabin scenario, this mechanism helps emphasize the target-center regions of cargo objects while reducing interference from surrounding package textures and cluttered cabin backgrounds.

From the perspective of feature representation, CAFM and WeightConv enhance different but complementary aspects of the detector. CAFM improves the joint modeling of global context and local structural details, which is useful for distinguishing partially occluded or densely stacked cargo. WeightConv further strengthens local convolutional feature extraction by assigning higher responses to more informative spatial positions and suppressing background interference. Therefore, the two modules jointly improve feature representation and detection performance in complex cabin scenes while maintaining the lightweight nature of the YOLOv11n baseline.

3.2. Object Tracking Algorithm

Object tracking aims to maintain target identities and estimate their motion states across continuous video frames, thereby providing stable trajectories for subsequent cargo-operation analysis [26]. The early SORT algorithm [27] performs target association using Kalman filtering and the Hungarian matching algorithm. It has low computational cost and good real-time performance, but it mainly relies on motion prediction and bounding-box overlap. Therefore, when cargo objects are temporarily occluded or missed by the detector, identity switches may occur and affect the continuity of cargo-management records.

DeepSORT extends SORT by introducing appearance embeddings into the association process [28]. In addition to motion consistency represented by Mahalanobis distance, it also evaluates visual similarity using cosine distance. This combination of motion and appearance cues helps maintain target identities after short-term occlusion or missed detection. Wojke et al. reported that DeepSORT reduced identity switches by about 45% while maintaining real-time performance, making it suitable for tracking tasks that require stable object identities.

For this study, DeepSORT was selected as the tracking component because it can directly use the bounding-box outputs of the improved detector and provides a mature detection-driven tracking framework. Its balance between association accuracy, implementation maturity, and computational cost is suitable for the relatively short cabin cargo video sequences considered in this work.

DeepSORT Algorithm Pipeline

In this study, the DeepSORT procedure is organized into state prediction, detection-to-track association, and track updating, as shown in Figure 4 [29].

During state prediction, a Kalman filter estimates the target state in the current frame from its previous trajectory state. At the same time, the detector provides current-frame bounding boxes as observations, and the feature extractor extracts the corresponding deep appearance features. In the data association stage, the predicted trajectories and observation data are jointly processed through cascade matching. The algorithm first performs cascaded association for confirmed trajectories by combining the cosine distance of appearance features with the Mahalanobis distance of motion states. It then handles unconfirmed trajectories and remaining detection boxes through IoU matching. In the parameter update stage, the algorithm updates the Kalman filter parameters and track life cycles based on the matching results.

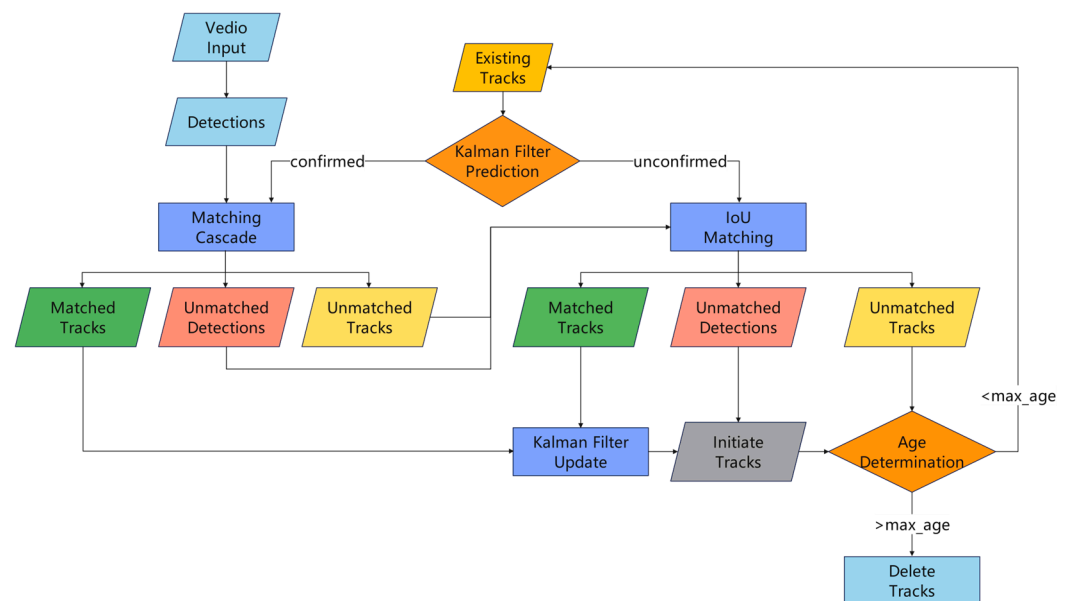


Figure 4. Cabin cargo tracking algorithm pipeline.

3.3. Action Classification Algorithm

3.3.1. Action Feature Extraction and Definition

Based on observations of astronauts' pick-and-place behaviors inside the cabin, behavioral feature data including the starting position, disappearance position, trajectory angle, trajectory straightness, and relative distance of the cargo are extracted using object detection and tracking algorithms. Combined with the actual motion characteristics of the cargo, a set of behavioral judgment rules for cabin cargo operations is defined.

- Starting position (P_{start}) and disappearance position (P_{end}): These refer to the starting and ending position coordinates of the cargo along its tracked trajectory during a period of continuous motion. R_{pkg} is defined as the interior region of the two-dimensional bounding box of the associated package.
- Trajectory angle (θ): The angle between the overall displacement direction of the cargo and the direction pointing toward the center of the package, used to determine whether the cargo is moving away from the package (larger θ) or approaching the package (smaller θ).
- Trajectory straightness (S): This is computed as the displacement between the start and end points divided by the accumulated trajectory length. Assuming the trajectory consists of N sequential points $P_1, P_2, \dots, P_N$, the formula is:

$$S = \frac{\|P_N - P_1\|}{\sum_{i=1}^{N-1} \|P_{i+1} - P_i\|} \quad (8)$$

- Relative distance (D): The Euclidean distance between the center coordinates of the target cargo (X_{obj}, Y_{obj}) and the center coordinates of its associated package (X_{pkg}, Y_{pkg}). The formula is:

$$D = \sqrt{(X_{obj} - X_{pkg})^2 + (Y_{obj} - Y_{pkg})^2} \quad (9)$$

“Pick” indicates that the cargo is taken out of the package by the astronaut, typically characterized by a large displacement and high trajectory straightness, with the trajectory clearly moving away from the package and a large trajectory angle. “Place” indicates that the cargo is placed into the package from outside, typically characterized by the cargo starting outside the package region, disappearing with high overlap with the package region, with the trajectory clearly approaching the package and a small trajectory angle. “Not picked” indicates that the cargo has only been partially moved by the astronaut

without any substantial transfer, with the trajectory often exhibiting significant back-and-forth movement, resulting in low straightness.

Based on the above feature definitions and the observed motion characteristics of the three cargo-operation categories, Table 1 summarizes the reference threshold-based criteria used to support action annotation and feature interpretation. In this table, θ_{th} , S_{th} , and D_{th} denote the reference thresholds for trajectory angle, trajectory straightness, and relative distance, respectively. These thresholds were empirically defined from the separability of trajectory features in the training samples and were used only as annotation guidelines, rather than fixed inference parameters. The final action recognition was performed by an SVM, which learned nonlinear decision boundaries from the extracted trajectory features and manually checked labels.

Table 1. Reference motion criteria for action annotation

Action Category	Judgment Condition
Pick	$P_{start} \in R_{pkg} \ \& \ D > D_{th} \ \& \ S > S_{th} \ \& \ \theta > \theta_{th}$
Place	$P_{start} \notin R_{pkg} \ \& \ P_{end} \in R_{pkg} \ \& \ D \leq D_{th} \ \& \ \theta \leq \theta_{th}$
Not Picked	$P_{start} \in R_{pkg} \ \& \ (D \leq D_{th} \ \ S \leq S_{th})$

3.3.2. Support Vector Machine

Support vector machine (SVM) is a supervised classifier that represents samples in a feature space and searches for a separating hyperplane with the maximum possible margin between categories [30].

For the cargo-operation task in this study, the three action labels are distinguished by nonlinear combinations of trajectory descriptors. An RBF-kernel SVM was therefore used to model these nonlinear boundaries from the extracted trajectory features. The multi-class problem was handled with a one-versus-one strategy, where the three action categories were converted into paired binary decisions and the final label was assigned according to the voting result of the decision functions [8].

Before model training, the trajectory features were normalized with the mean and variance calculated from the training set. The classifier was implemented with scikit-learn using an RBF kernel, $C = 1.0$, and $\gamma = \text{scale}$, and these settings were kept unchanged during testing. This configuration was selected because it can describe nonlinear class boundaries with a small computational burden, which matches the scale of the trajectory-feature dataset used here.

In this experiment, accuracy is used to quantitatively evaluate the classification results, and a confusion matrix is employed to intuitively present the model's predictive performance on easily confused actions.

3.4. Evaluation Metrics

3.4.1. Detection Algorithm Evaluation Metrics

To evaluate the overall detection performance of the improved algorithm, this paper adopts the following quantitative metrics [31]: precision (P), recall (R), mean average precision (mAP), floating point operations (GFLOPs), parameters, and FPS.

- Precision: Measures the accuracy of the model when predicting positive samples.

$$P = \frac{TP}{TP + FP} \quad (10)$$

- Recall: Measures the model's ability to identify positive samples.

$$R = \frac{TP}{TP + FN} \quad (11)$$

Here, TP (true positive) is the number of correctly detected cargo objects, FP (false positive) is the number of non-cargo or incorrect detections counted as cargo, and FN (false negative) is the number of missed cargo objects.

- Mean Average Precision: Measures the overall detection performance of the model across all categories.

$$AP = \int_0^1 P(R) dR \quad (12)$$

$$mAP = \frac{1}{C} \sum_{i=1}^C AP_i \quad (13)$$

where $P(R)$ is the precision-recall curve function, C is the total number of target categories to be detected, and AP_i is the average precision of the i -th category target.

Parameter count and GFLOPs are reported to describe model size and computational cost, while FPS is used to characterize inference speed on the test platform.

3.4.2. Tracking Algorithm Evaluation Metrics

To evaluate the performance of the tracking algorithm, the TrackEval tool is employed along with the following metrics [32]: multiple object tracking accuracy (MOTA), multiple object tracking precision (MOTP), identity F1 score (IDF1), and higher-order tracking accuracy (HOTA) [33].

- MOTA: Measures the overall performance of the tracker, incorporating three dimensions of error: false detections (FP), missed detections (FN), and identity switches (IDs):

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \quad (14)$$

where FN_t denotes the number of missed detections in frame t , FP_t denotes the number of false detections in frame t , GT_t denotes the number of ground truth targets in frame t , and $IDSW_t$ denotes the number of identity switches in frame t .

- MOTP: Measures the accuracy of the tracker's position estimation for targets:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \quad (15)$$

where $d_{t,i}$ denotes the distance error between the i -th match in frame t , and c_t denotes the number of successful matches in frame t .

- IDF1: Measures the tracker's ability to maintain consistent target identity:

$$IDF1 = \frac{2 \times IDTP}{2 \times IDTP + IDFP + IDFN} \quad (16)$$

where IDTP denotes the number of detection frames with correctly assigned ID throughout the entire video, IDFP denotes the number of detection frames with incorrectly assigned ID, and IDFN denotes the number of ground truth targets with missed ID assignments.

- HOTA: A higher-order comprehensive evaluation metric that measures both detection accuracy and association accuracy:

$$HOTA = \sqrt{DetA \times AssA} \quad (17)$$

where $DetA$ denotes the target detection accuracy score under multiple thresholds, and $AssA$ denotes the corresponding data association accuracy score.

4. Experiment

4.1. Datasets

To verify the effectiveness and generalization capability of the improved detection and tracking methods, we conducted a comprehensive evaluation on two self-constructed datasets (CargoShipDetect and CargoShipTrack) as well as two public datasets (COCO [34] and MOT20 [35]).

The two self-constructed datasets aim to authentically reproduce the scene of astronauts picking and placing cargo inside the cabin. The raw videos were all captured by fixed cameras installed inside the cabin, with a collected video resolution of 1280×720 pixels and a frame rate of 30 f/s. Annotation work was completed using the DarkLabel tool. Sample images from the datasets are shown in Figure 5.

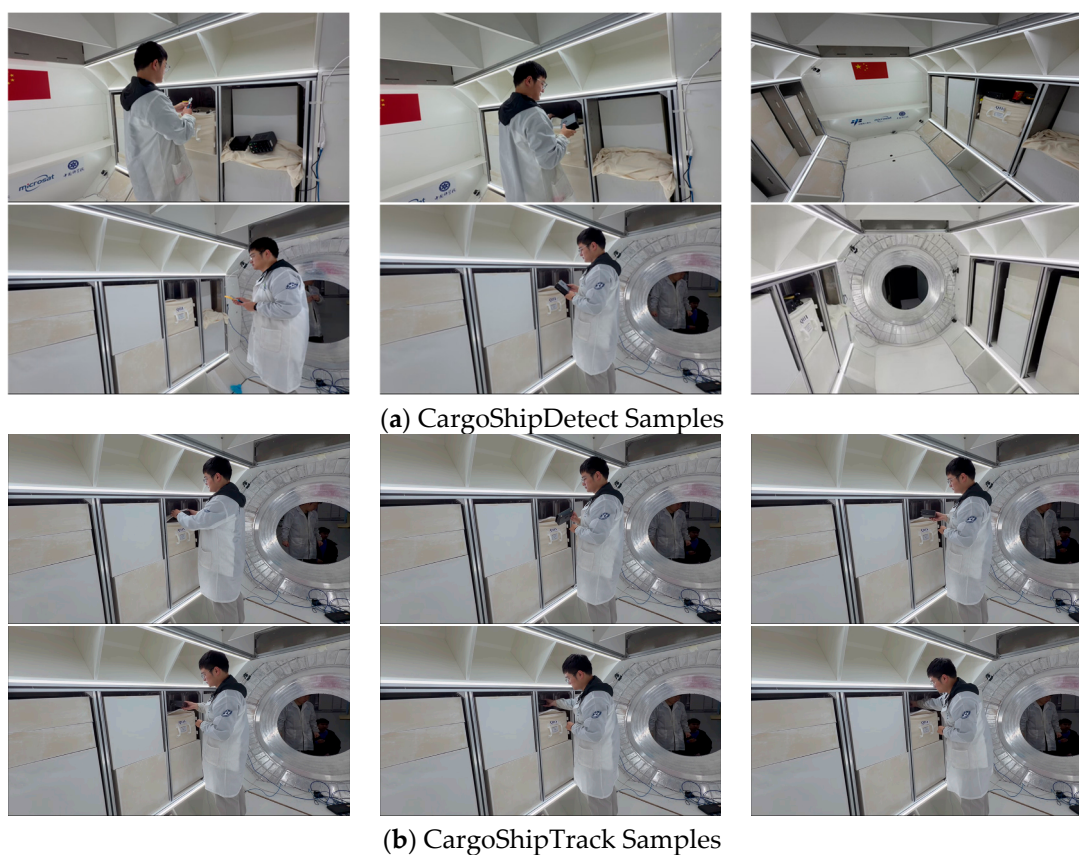


Figure 5. Dataset samples.

The CargoShipDetect dataset supports detector training and validation. It was annotated in a YOLO-style detection format, including object categories and bounding boxes. Through manual screening, 10,000 images were collected, including frames extracted from part of the CargoShipTrack training videos and additional independently collected cabin images. The training set, validation set, and test set contain 8000, 1500, and 500 images, respectively. The dataset covers multiple types of supplies essential for astronauts, including standard cargo bags, food, and tools inside the cabin.

The CargoShipTrack dataset is used to evaluate the continuous tracking capability of the tracking algorithm during astronauts' cargo pick-and-place operations. It was annotated in an MOT-style sequence format, including frame-level bounding boxes and object identities for tracking evaluation. The dataset contains a total of 20 video sequences, with 10 videos used for training, 7 independent videos used for validation, and 3 videos used for testing. Each video is approximately 50 s in length. The validation and test videos were kept independent from the video frames used to construct the detection dataset.

The COCO dataset is a widely used public dataset in the field of computer vision, covering 80 common categories, including 118,287 training images, 5000 validation images, and 40,670 test images. The images are predominantly from real complex scenes, characterized by strong background interference, a high proportion of small targets, and frequent mutual occlusion between objects.

The MOT20 dataset is one of the important benchmark test datasets in the field of multi-object tracking, collected from complex real-world scenes, containing 8 video sequences, of which 4 are used for training and 4 for testing. Compared with the earlier MOT15 [36] and MOT16 [37], the most notable characteristic of MOT20 is its high target density, with an average of more than 120 targets per frame. MOT20 contains high-density crowds, complex backgrounds, and dense occlusion, which are similar to dense cargo-stacking conditions inside cargo spacecraft cabins. Therefore, it is used to evaluate the re-identification capability and tracking robustness of DeepSORT in complex scenes.

4.2. Experiment Environment

All experiments were implemented in Python 3.10 with PyTorch 1.13.1, and CUDA 11.6 was used for GPU acceleration. The training and evaluation procedures were performed on a workstation equipped with an Nvidia GeForce RTX 3090 GPU.

For detector training, model parameters were optimized using SGD. The training configuration included 300 epochs, an initial learning rate of 0.01, a batch size of 16, and an input image size of 640×640 pixels.

Furthermore, to ensure the objectivity of the evaluation results, we conducted detailed comparative experiments on detection algorithms including Faster RCNN, SSD, and YOLOv5n under the same testing environment. The tracking algorithm and pick-and-place action classification were also trained and tested on the above hardware platform.

4.3. Object Detection Evaluation

The COCO benchmark is used to examine detector behavior in a general object-detection setting.

Table 2 compares representative detectors under the same evaluation setting. The two-stage network Faster R-CNN performs better in recall and mean average precision. However, its computational overhead reaches 134 GFLOPs and its inference speed is only 10 frames/s, making it difficult to meet the deployment constraints of cargo spacecraft edge computing devices. In comparison, the baseline model YOLOv11n maintains a precision of 61.2% and a processing speed of 70 frames/s under the premise of extremely low computational cost and parameter count.

Table 2. Performance comparison of detection algorithms on the COCO dataset

Model	+CAFM	+WeightConv	P/%	R/%	mAP50-95/%	GFLOPs	Params/M	FPS
Faster RCNN	×	×	58.5	50.9 ¹	36.9 ¹	134	41.7	10
SSD	×	×	40.6	34.4	24.6	34.9	35.6	21
YOLOv5n	×	×	60.7	46.8	34.0	7.8	2.7	80 ¹
YOLOv11n	×	×	61.2	45.9	35.6 ²	6.5 ²	2.6 ¹	70 ²
YOLOv11n	√	×	61	46.2	33.9	6.5	2.7	60
YOLOv11n	×	√	62.1 ²	45.1	33.5	5.0	2.6	67
Ours	√	√	63.7 ¹	47.2 ²	35.3	5.0 ¹	2.7 ²	66

¹ Superior outcomes; ² next-best results. Here and in the following tables, √ indicates that the corresponding module is applied, and × indicates that it is not.

After introducing the CAFM and WeightConv modules, the proposed algorithm improves precision and recall by 2.5 and 1.3 percentage points, respectively, compared with

the baseline model. The computational cost is reduced to 5 GFLOPs and the inference speed remains 66 frames/s. The comparison indicates that the proposed detector keeps a favorable balance between accuracy and inference efficiency in the tested general dataset.

Figure 6 shows the evaluation curves of the improved algorithm trained for 300 epochs on the CargoShipDetect dataset. The loss decreases rapidly in the early stages of training and gradually stabilizes with no significant fluctuations in the later stages, indicating that the model quickly fits the cabin cargo features without overfitting. Precision, recall, and mean average precision all steadily improve and maintain high stability, demonstrating that the model has reached an ideal convergence state.

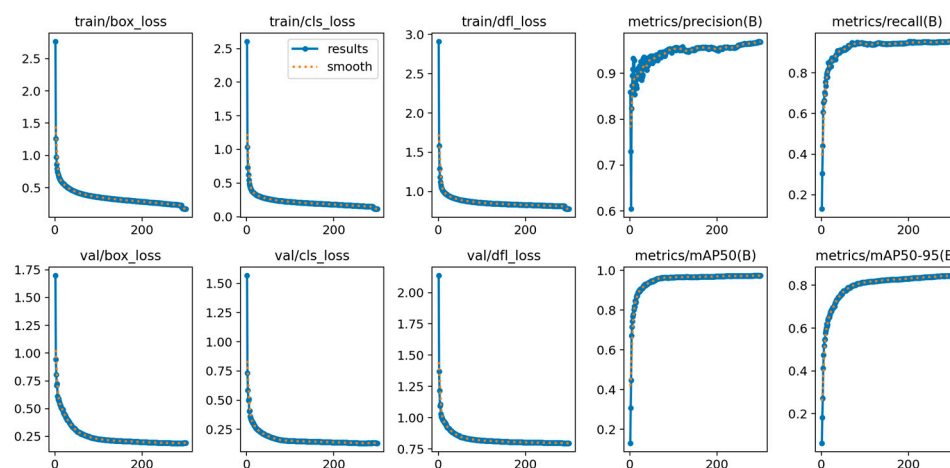


Figure 6. Improved model training results.

To verify the effectiveness of each improved module in handling complex cabin backgrounds and dense occlusion, we conducted ablation experiments on the CargoShipDetect dataset, with results shown in Table 3. As can be seen from the table, after introducing solely CAFM, the model's precision reaches 96.9%, while mAP50-95 increases from 83.9% to 85.9%, indicating that the global perception mechanism of this module helps mitigate the feature loss caused by occlusion. After introducing solely the WeightConv operator, the model's recall and mAP50-95 increase to 96.1% and 85.2%, respectively, reflecting the operator's advantages in focusing on key features and suppressing background noise. When both modules are fused simultaneously, the precision and recall improve by 0.6 and 0.3 percentage points, respectively, compared to the baseline model, with mAP50-95 increasing by 2.3 percentage points to reach 86.2%. Although the improvements in precision and recall are relatively modest, the increase in mAP50-95 suggests that the fused modules improve localization quality under stricter IoU thresholds and enhance feature representation in the current experimental setting.

Table 3. Ablation experiments

Model	+CAFM	+WeightConv	P/%	R/%	mAP50/%	mAP50-95/%
YOLOv11n	×	×	96.7	95.9	97.7	83.9
YOLOv11	√	×	96.9	95.9	97.8	85.9
YOLOv11	×	√	96.5	96.1	97.7	85.2
Ours	√	√	97.3 ¹	96.2 ¹	98.0 ¹	86.2 ¹

¹ Superior outcomes.

To further examine the edge-inference efficiency of the proposed detector, a preliminary deployment test was conducted on an Atlas 310B1 embedded platform. Both models were converted into OM format using FP16 precision. The input resolution was fixed at

640 × 640 with a batch size of 1. A total of 100 test images were used, and each image was inferred 500 times after 50 warm-up runs, resulting in 50,000 timed inference runs for each model. The results are shown in Table 4.

Table 4. Inference-speed comparison on the Atlas 310B1 embedded platform

Model	H2D Latency/ms	NPU Compute Time/ms	D2H Latency/ms	Throughput/FPS
YOLOv11n	1.046 ¹	11.144	0.161	89.73
Ours	1.055	9.996 ¹	0.159 ¹	100.04 ¹

¹ Superior outcomes.

As shown in Table 4, the improved detector achieves an average NPU compute time of 9.996 ms and a throughput of 100.04 FPS, compared with 11.144 ms and 89.73 FPS for the YOLOv11n baseline. This corresponds to a 10.31% reduction in NPU compute time and an 11.49% increase in throughput. The result indicates that the introduced CAFM and WeightConv modules do not cause a prohibitive inference burden on the Atlas 310B1 platform. Under the current FP16 deployment setting, the improved detector maintains practical edge-inference efficiency while improving detection performance on the self-constructed dataset.

To intuitively demonstrate the performance advantages of the improved model in the cabin environment, representative images containing mutual occlusion and small target features were selected, and inference was performed separately using the baseline and improved models, as shown in Figure 7.

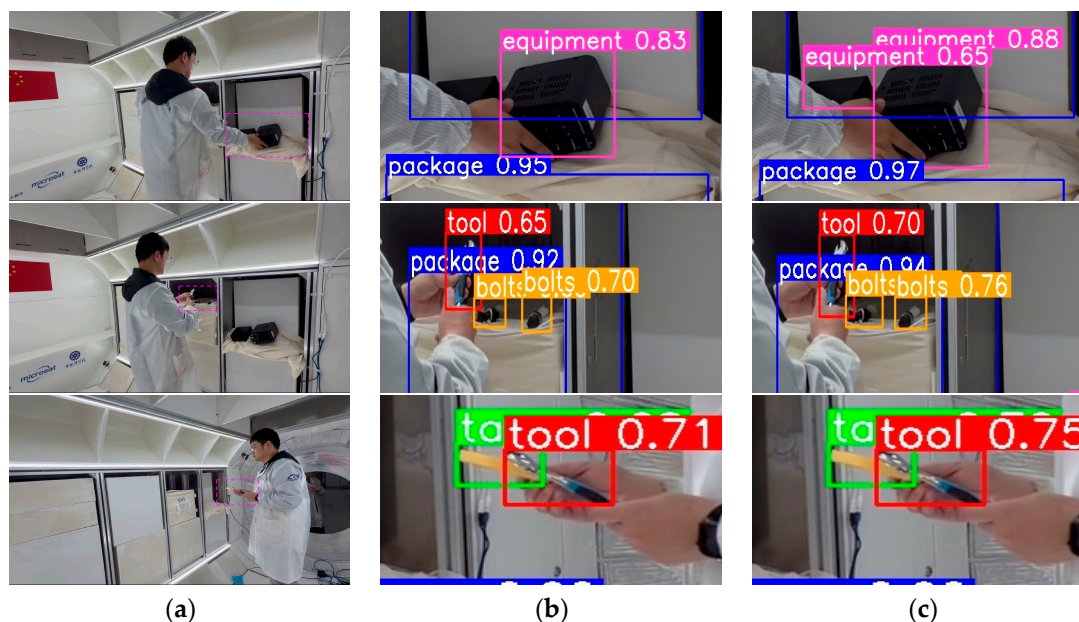


Figure 7. Performance comparison between baseline and improved models in the cabin. (a) Original Image. (b) Baseline recognition results (magnified view). (c) Our recognition results (magnified view).

Figure 7a presents the original images, while Figure 7b,c respectively display the detection results of the baseline and improved models, with local magnified views further highlighted through dashed boxes to better illustrate the recognition details. From the comparison results, it can be observed that the baseline model exhibits obvious missed detections when facing dense overlapping and local occlusion, and the detection confi-

dence for small targets is generally low. In contrast, under the interference of local occlusion and complex backgrounds, the improved model increases the recall rate and confidence scores of several key targets. These qualitative results further indicate that the improved model enhances feature extraction for partially occluded and densely arranged cargo in the tested cabin scenes.

4.4. Object Tracker Evaluation

To verify the stability of the DeepSORT algorithm in handling high-density occlusion and densely packed targets, this paper compares the SORT and DeepSORT algorithms on the public dataset MOT20, with results shown in Table 5. Experiments demonstrate that the SORT algorithm, which relies solely on motion information, is extremely prone to trajectory interruptions when facing high-density occlusion, exhibiting an extremely high number of identity switches. In contrast, when DeepSORT is equipped with various front-end detectors, its number of identity switches is significantly lower than that of the SORT algorithm. When the improved algorithm proposed in this paper is used as the front-end detector combined with DeepSORT for tracking, although MOTA decreases by 2.2% compared to the baseline YOLOv11n, HOTA improves to 49.3%, IDF1 improves to 59.5%, and IDSW decreases to 2758. These results suggest that the improved detector can help reduce identity switches under dense occlusion, although more recent trackers may provide stronger association performance in crowded scenes.

Table 5. Performance of tracking algorithms on the MOT20 dataset

Detector	Tracker	HOTA, %	MOTA, %	MOTP, %	IDF1, %	IDSW
Faster RCNN	SORT	30.7	47.1	83.3	33.0	14,094
Faster RCNN	DeepSORT	33.3	56.0	84.0	36.3	11,664
YOLOv11n	SORT	44.6	62.1 ²	84.5	50.8	6456
YOLOv11n	DeepSORT	45.0	63.7 ¹	82.3	52.4	4536
Ours	SORT	47.8 ²	57.0	87.1 ¹	55.8 ²	3610 ²
Ours	DeepSORT	49.3 ¹	61.5	84.8 ²	59.5 ¹	2758 ¹

¹ Superior outcomes; ² next-best results.

Tracking performance in the cabin scenario was evaluated on CargoShipTrack, and the results are listed in Table 6. With the same DeepSORT tracker, replacing the baseline detector with the improved detector increases MOTA to 97.1%, while HOTA and IDF1 reach 86.2% and 95.1%, respectively. The number of ID switches is also reduced from 42 to 37. Among the tested combinations, improved YOLO + DeepSORT provides the most stable tracking results on the self-constructed dataset and supplies trajectory data for the following action-recognition stage.

Table 6. Performance comparison of tracking algorithms on the CargoShipTrack dataset

Detector	Tracker	HOTA, %	MOTA, %	MOTP, %	IDF1, %	IDSW
YOLOv11n	SORT	80.0	87.2	89.0	89.9	56
YOLOv11n	DeepSORT	85.9	96.9	89.4	94.8	42
Ours	SORT	80.5	87.2	89.1	90.7	55
Ours	DeepSORT	86.2 ¹	97.1 ¹	89.6 ¹	95.1 ¹	37 ¹

¹ Superior outcomes.

4.5. Action Classification Algorithm Evaluation

We first performed feature extraction for three typical actions: “pick,” “place,” and “not picked.” Based on the trajectories generated by detection and tracking, the temporal

variation curves of relative distance, trajectory straightness, and trajectory angle were analyzed, as shown in Figure 8. From the characteristic curves in Figure 8, it can be observed that different actions exhibit significant separability in the multidimensional feature space. At the same time, Figure 9 intuitively presents the dynamic evolution process of the three typical actions across consecutive frames.

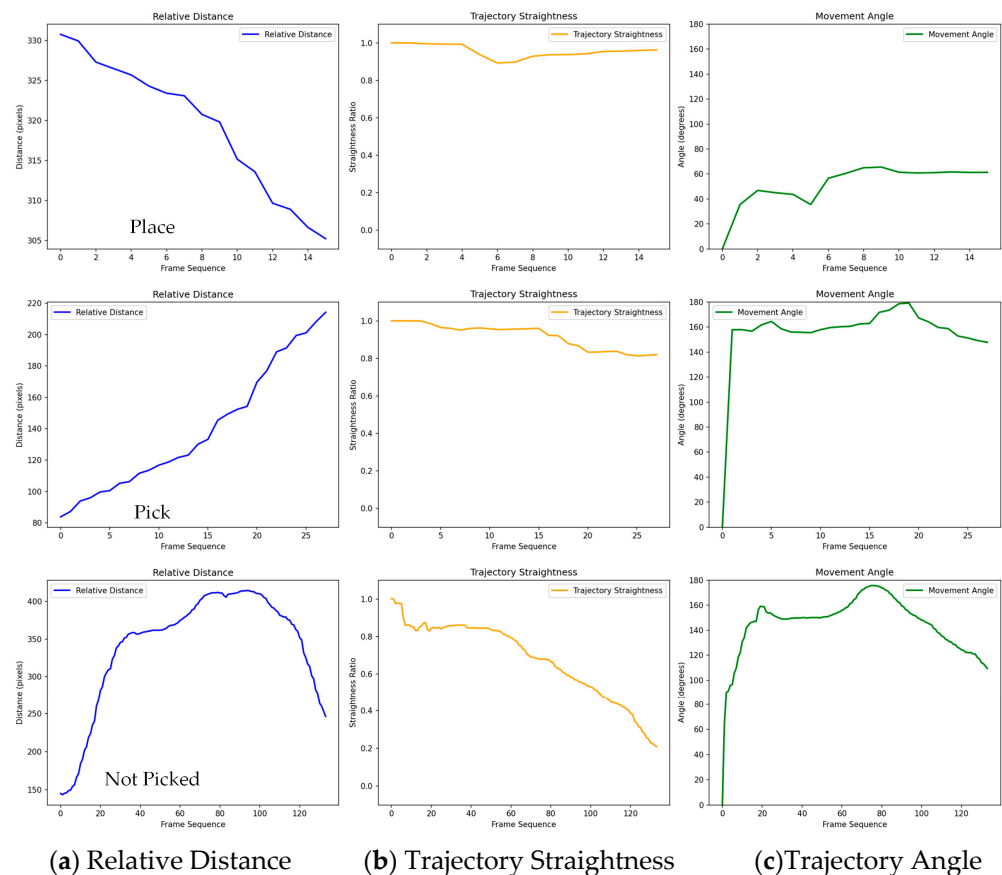


Figure 8. Characteristic variation curves of three actions.

Based on the improved detector and DeepSORT tracking results, 200 representative cargo trajectory segments were extracted from the CargoShipTrack video sequences for SVM-based action classification. The action category of each trajectory segment was manually labeled “pick,” “place,” or “not picked” according to the observed cargo-operation process. The dataset contained 60 pick samples, 70 place samples, and 70 not picked samples.

A valid trajectory segment was required to contain at least 15 consecutive trajectory records after tracking post-processing, and fragments with a maximum displacement lower than 30 pixels were discarded as nearly static background motions. Track discontinuities were handled by the DeepSORT track-management mechanism with $\text{max_iou_distance} = 0.8$, $\text{max_age} = 30$, and $\text{n_init} = 3$. Only confirmed tracks were used for feature extraction. In addition, a trajectory point was added to the feature history only when the corresponding track had been updated within the two most recent frames ($\text{time_since_update} \leq 2$). Thus, short missed detections could be bridged without immediately terminating the trajectory, whereas a trajectory segment was finalized when the corresponding object disappeared from the confirmed active tracks or when the video ended.

For each valid trajectory segment, the SVM used relative distance, trajectory straightness, and trajectory angle as input descriptors. The samples were split into training and

test subsets at a ratio of 7:3. Figure 10 presents the confusion matrix obtained on the test subset. The overall accuracy reaches 92%, showing that these trajectory descriptors are sufficient to distinguish the three typical cargo-operation states in the current dataset.

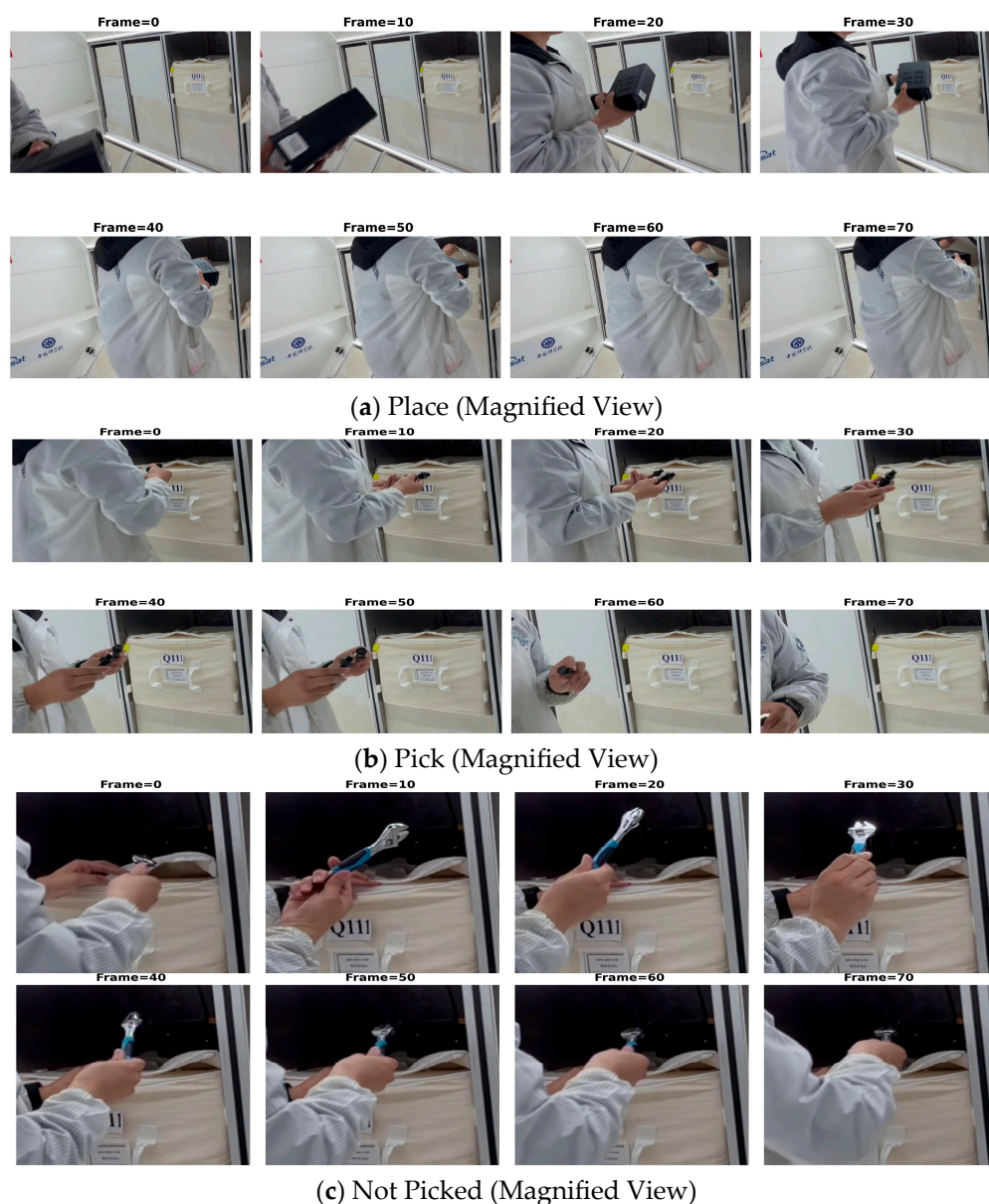


Figure 9. Demonstration of three actions.

As can be seen from Figure 10, the model achieves clear separability and completely correct recognition for the “pick” and “place” actions, which show relatively opposite displacement patterns. A small number of classification errors are concentrated between “not picked” and the other two actions, mainly because short-distance local handling in the not-picked category may produce trajectory features similar to weak pick or place motions. Overall, the proposed framework can automatically determine typical cabin cargo pick-and-place behaviors.

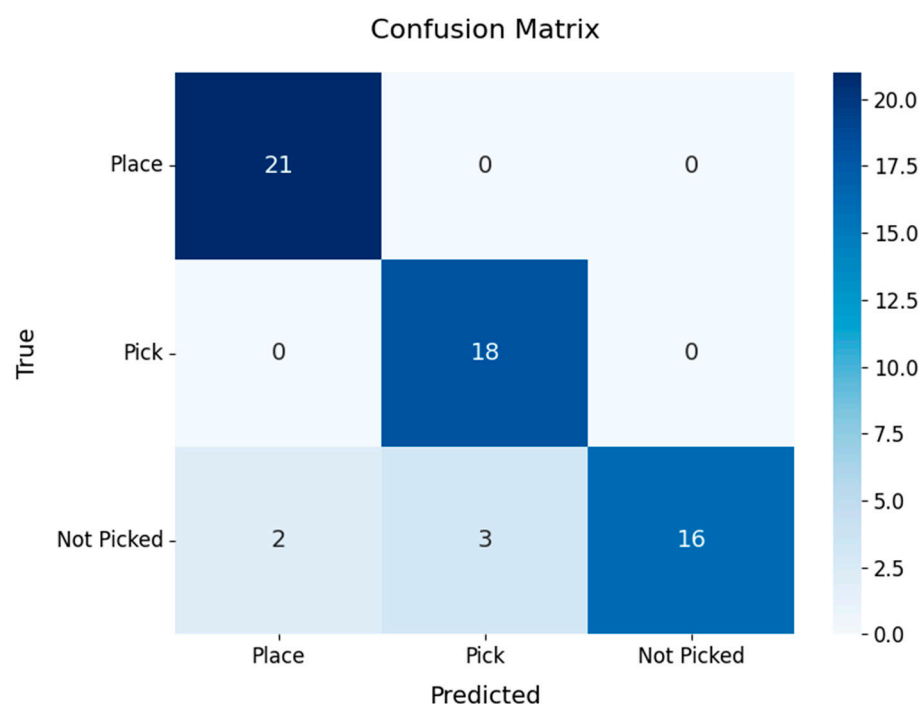


Figure 10. SVM classification results.

5. Conclusions

1. This study developed a visual cargo-management framework for spacecraft cabins by integrating improved YOLO detection, DeepSORT tracking, and SVM-based action classification. WeightConv and CAFM are introduced into the detector to enhance cargo feature representation under complex cabin backgrounds, dense stacking, and partial occlusion. Based on the detected cargo objects, DeepSORT is used to obtain continuous trajectories across video frames, and the extracted trajectory descriptors are further used to recognize typical astronaut cargo-operation behaviors.
2. Experimental results on the self-constructed dataset show that the improved detector achieves a precision of 97.3%. For multi-object tracking, the proposed framework obtains HOTA, MOTA, MOTP, and IDF1 values of 86.2%, 97.1%, 89.6%, and 95.1%, respectively. For action classification, the trajectory-based SVM reaches an overall accuracy of 92.0%. These results indicate that the proposed detection-tracking-recognition framework can support the analysis and automatic recording of typical cargo pick-and-place operations under the current experimental conditions.
3. Compared with traditional RFID-based cargo management, which mainly provides passive static information, the proposed visual framework further introduces trajectory tracking and action recognition. It can provide dynamic information about cargo movement and operation states, thereby offering technical support for more intelligent in-orbit cargo management. However, further deployment-oriented validation is still needed before the method can be applied under practical onboard deployment conditions.
4. Several limitations remain in the current study. First, the action classifier is based on manually designed trajectory descriptors and an RBF-SVM, which is appropriate for the relatively small trajectory-feature dataset used in this work. For more complex multistep cargo manipulation scenarios, richer temporal modeling may be required. Future work will compare LightGBM, lightweight neural networks, LSTM-based models, and transformer-based temporal models, and will explore the integration of spatiotemporal context, depth information, and multi-camera perception to improve action understanding under occlusion.

5. Second, although DeepSORT provides a stable tracking basis for the current framework, more recent trackers such as ByteTrack, OC-SORT, and StrongSORT may offer stronger association performance in dense or severely occluded scenes. These methods will be further evaluated in future studies. In addition, the current experiments were mainly conducted on terrestrial GPU platforms and self-constructed cabin video sequences, with preliminary deployment tests on a terrestrial embedded platform. Future work will further examine the framework under more diverse cabin operating conditions, representative onboard hardware constraints, and environmental disturbances such as illumination fluctuation, camera-viewpoint variation, and vibration.

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

Funding: This research received no external funding.

Data Availability Statement: The data supporting this study’s findings are available from the corresponding author upon reasonable request.

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

References

1. Chen, H.; Ji, Y.; Xuan, J.; Meng, Y.; Wu, J. Analysis of the Development of International Space Station Cargo Transport System and Its Enlightenment for China. *Missiles Space Veh.* **2025**, *5*, 24–30. <https://doi.org/10.7654/j.issn.2097-1974.20250509>.
2. Fetisov, V.A.; Taratun, V.E. RFID Technology for Automation of Transport and Storage Operations in Space. In Proceedings of the 2018 Wave Electronics and Its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, Russia, 26–30 November 2018; pp. 1–5. <https://doi.org/10.1109/WECONF.2018.8604386>.
3. Bao, W.M.; Qi, Z.Q. Thoughts on the Development of Aerospace Control Technology in the Evolution of Intelligence. *Sci. Sin. Technol.* **2025**, *55*, 1437–1446. <https://doi.org/10.1360/SST-2025-0078>.
4. Khan, S.A.; Ali, A.; Shiyu, Y.; Fahad, S.; Tong, J. Optimized Design and Analysis of Printed Magnetorquer for a 3-U Nano-Satellite. *J. Aerosp. Eng.* **2021**, *35*, 04021103. [https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0001343](https://doi.org/10.1061/(ASCE)AS.1943-5525.0001343).
5. Xie, T.; Yao, X. Smart Logistics Warehouse Moving-Object Tracking Based on YOLOv5 and DeepSORT. *Appl. Sci.* **2023**, *13*, 9895. <https://doi.org/10.3390/app13179895>.
6. Ma, J. Research on the Algorithm of Commodity Pick-Up Recognition. Master’s Thesis, Southwest University of Science and Technology, Mianyang, China, 2023. <https://doi.org/10.27415/d.cnki.gxngc.2023.001137>.
7. Zhai, J. Commodity Recognition System of Intelligent Vending Machine Based on Deep Learning. Master’s Thesis, Huazhong University of Science and Technology, Wuhan, China, 2020. <https://doi.org/10.27157/d.cnki.ghzku.2020.006478>.
8. Zhang, Z.; Wang, C.; Chen, J.; Okinda, C.; Liu, L. Research on Freshwater Fish Behavior Detection Algorithm Based on YOLOX and DeepSORT. *J. Intell. Agric. Mech.* **2025**, *6*, 88–97.
9. Wang, M.; Yang, E. Sheep Multi-Object Tracking Method Integrating Depth Information and Motion Trends. *Trans. Chin. Soc. Agric. Mach.* **2025**, *56*, 475–481. <https://doi.org/10.6041/j.issn.1000-1298.2025.05.045>.
10. Tu, S.; Du, J.; Liang, Y.; Cao, Y.; Chen, W.; Xiao, D.; Huang, Q. Tracking and Behavior Analysis of Group-Housed Pigs Based on a Multi-Object Tracking Approach. *Animals* **2024**, *14*, 2828. <https://doi.org/10.3390/ani14192828>.
11. Feng, S.; Wang, T.; Guo, Y.; Qiu, J. Online Test-Time Adaptation for On-Orbit Noncooperative Spacecraft Pose Estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2026**, *62*, 449–460. <https://doi.org/10.1109/TAES.2025.3623610>.
12. Feng, S.; Jin, D.; Cui, W.; Wang, T.; Ma, M. Self-Distillation Adaptive Pose Estimation Method for Cross-Domain Noncooperative Spacecraft Based on Optical Sensor Data. *IEEE Sens. J.* **2025**, *25*, 41758–41767. <https://doi.org/10.1109/JSEN.2025.3615727>.
13. Zhou, D.; Sun, G.; Lei, W. Space Non-Cooperative Object Active Tracking with Deep Reinforcement Learning. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *58*, 4902–4916.

14. Zhang, R.; Wang, Z. People Visual Tracking for Assistant Robots in Spacecraft Based on Deep Learning. *Aerosp. Shanghai* **2018**, *35*, 1–9. <https://doi.org/10.19328/j.cnki.1006-1630.2018.05.001>.
15. Zhang, Z.; Feng, J.; Chang, L.; Deng, L.; Li, D.; Si, C. SpaceLight: A Framework for Enhanced On-Orbit Navigation Imagery. *Aerospace* **2024**, *11*, 503. <https://doi.org/10.3390/aerospace11070503>.
16. Cao, Z.; Cao, Y.; Li, J.; Xu, C.; Tan, Q. Research on Target Detection in Space Station Cabin Based on Convolutional Neural Network. *J. Northwestern Polytech. Univ.* **2020**, *38*, 84–89.
17. Yuan, H.; Zhu, H.; Sui, T.; Zhang, Z. PBRA-YOLO: An Enhanced Model for Tool Detection in Cargo Spacecraft Cabins. In Proceedings of the Second International Conference on Image Processing, Intelligent Control, and Computer Engineering (IPICE 2025), Zhengzhou, China, 25–27 July 2025; pp. 743–755. <https://doi.org/10.1117/12.3087049>.
18. Sun, Z.; Chen, J.; Chao, L.; Ruan, W.; Mukherjee, M. A Survey of Multiple Pedestrian Tracking Based on Tracking-by-Detection Framework. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 1819–1833. <https://doi.org/10.1109/TCSVT.2020.3009717>.
19. Zhao, Y.; Rao, Y.; Dong, S.; Zhang, J. Survey on Deep Learning Object Detection. *J. Image Graph.* **2020**, *25*, 629–654. <https://doi.org/10.11834/jig.190307>.
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. <https://doi.org/10.1109/CVPR.2016.91>.
22. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
23. Ma, Z.; Wang, C.; Chen, C.; Chen, J.; Zheng, G. DDF-YOLO: A Small Target Detection Model Using Multi-Scale Dynamic Feature Fusion for UAV Aerial Photography. *Aerospace* **2025**, *12*, 920. <https://doi.org/10.3390/aerospace12100920>.
24. Hu, S.; Gao, F.; Zhou, X.; Dong, J.; Du, Q. Hybrid Convolutional and Attention Network for Hyperspectral Image Denoising. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 5504005. <https://doi.org/10.1109/LGRS.2024.3370299>.
25. Cammarasana, S.; Patanè, G. Optimal Weighted Convolution for Classification and Denoising. *arXiv* **2025**, arXiv:2505.24558. <https://doi.org/10.48550/arXiv.2505.24558>.
26. Li, X.; Zha, Y.; Zhang, T.; Cui, Z.; Zuo, W.; Hou, Z.; Lu, H.; Wang, H. Survey of Visual Object Tracking Algorithms Based on Deep Learning. *J. Image Graph.* **2019**, *24*, 2057–2080. <https://doi.org/10.11834/jig.190372>.
27. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple Online and Realtime Tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Rouen, France, 25–28 September 2016; pp. 3464–3468. <https://doi.org/10.1109/ICIP.2016.7533003>.
28. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649. <https://doi.org/10.48550/arXiv.1703.07402>.
29. Bai, Y.; Liu, D. Lightweight Traffic Flow Detection Based on Improved YOLOv10. *Mod. Inf. Technol.* **2025**, *9*, 77–82. <https://doi.org/10.19850/j.cnki.2096-4706.2025.08.015>.
30. Zhao, Q.; Zhang, Z.; Huang, Y.; Fang, J. TPE-RBF-SVM Model for Soybean Categories Recognition in Selected Hyperspectral Bands Based on Extreme Gradient Boosting Feature Importance Values. *Agriculture* **2022**, *12*, 1452. <https://doi.org/10.3390/agriculture12091452>.
31. Zhou, Y.; Zhang, T.; Li, Z.; Qiu, J. Improved Space Object Detection Based on YOLO11. *Aerospace* **2025**, *12*, 568. <https://doi.org/10.3390/aerospace12070568>.
32. Tu, S.; Zeng, Q.; Liang, Y.; Liu, X.; Huang, L.; Weng, S.; Huang, Q. Automated Behavior Recognition and Tracking of Group-Housed Pigs with an Improved DeepSORT Method. *Agriculture* **2022**, *12*, 1907. <https://doi.org/10.3390/agriculture12111907>.
33. Luiten, J.; Ošep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; Leibe, B. HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. *Int. J. Comput. Vision* **2021**, *129*, 548–578. <https://doi.org/10.1007/s11263-020-01375-2>.
34. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.
35. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; Leal-Taixé, L. MOT20: A Benchmark for Multi-Object Tracking in Crowded Scenes. *arXiv* **2020**, arXiv:2003.09003. <https://doi.org/10.48550/arXiv.2003.09003>.

36. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv* **2015**, arXiv:1504.01942. <https://doi.org/10.48550/arXiv.1504.01942>.
37. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A Benchmark for Multi-Object Tracking. *arXiv* **2016**, arXiv:1603.00831. <https://doi.org/10.48550/arXiv.1603.00831>.

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.