aerospace

MDPI

Article

Recognition of Aircraft Activities at Airports on Video Micro-Satellites: Methodology and Experimental Validation

Rui Zhang *, Xueyang Zhang, Longlong Xiao and Jiayu Qiu

School of Space Command, Space Engineering University, Beijing 101416, China; zxy1135@outlook.com (X.Z.); 18500424822@126.com (L.X.); q18192651989@163.com (J.Q.)
* Correspondence: zhangrui07820@126.com

Abstract: The remote sensing satellite constellation based on micro-satellites is an important means to construct a global and all-sky earth observation system in the future. Therefore, realizing the recognition of aircraft activities on video micro-satellites is a key technology that needs to be solved urgently. In this paper, an efficient algorithm for aircraft activity recognition that can be deployed on video micro-satellites was proposed. First, aircraft detection was performed on the first incoming remote sensing image using a robust DCNN-based object detection model. Then, a multi-target tracking model incorporating geospatial information was built for aircraft tracking and activity recognition. The algorithm was deployed on an embedded AI computer which was a COTS component. The algorithm was verified using remote sensing videos from commercial micro-satellites. Experimental results show that the algorithm can process aircraft targets of different sizes, and is equally effective even with complex environmental backgrounds, lighting conditions, and various movements of the aircraft, such as turning, entering, and exiting. Based on aircraft tracking results and geospatial information, the motion speed of each aircraft can be obtained, and its activity can be divided into parking, taxiing, or flying. The scheme proposed in this paper has good application prospects in the realization of on-orbit event recognition in micro-satellites with limited computing and memory resources.

Keywords: micro-satellite; staring-imaging; aircraft tracking; DCNN; embedded AI processor

check for updates

Citation: Zhang, R.; Zhang, X.; Xiao, L.; Qiu, J. Recognition of Aircraft Activities at Airports on Video Micro-Satellites: Methodology and Experimental Validation. *Aerospace* **2022**, *9*, 414. https://doi.org/10.3390/aerospace9080414

Academic Editors: Vaios Lappas, Angelo Cervone and Vittorio Franzese

Received: 26 April 2022
Accepted: 19 July 2022
Published: 30 July 2022

1. Introduction

Aircraft activity recognition is of great significance in airport capability assessments and military intelligence acquisition, among other applications. Compared with ground-based monitoring methods, remote-sensing-based aircraft activity recognition can obtain results with wider coverage and more comprehensive information. Video satellites are a new type of earth observation satellite; compared with traditional remote sensing satellites, they can continuously observe a certain area and obtain more dynamic information using a staring mode. They are more suitable for tasks with time-sensitive characteristics, such as the recognition of aircraft activities at airports. However, it is difficult to meet the time requirements of emergency information acquisition with the traditional remote sensing process of detection, downlink transmission, and then processing. As shown in Figure 1 (green), the traditional method has poor timeliness and brings a lot of pressure on satellite-to-ground data transmission. It can cost hours to acquire remote sensing information, and a huge amount of remote sensing data needs to be downloaded. In addition, due to the constraints of orbital dynamics, a single satellite cannot achieve global monitoring, or all-day monitoring, of a certain area.

To improve the coverage and duration of observations, satellite constellations are a good solution. However, more satellites mean higher budgets. Micro-satellites, on the other hand, have the advantages of low cost, a short development cycle, and the ability to form a network. A remote sensing satellite constellation based on micro-satellites is a better way to construct a global and all-day earth observation system. Additionally, on-board processing is the most straightforward way to improve the timeliness of observation tasks. As shown in Figure 1 (blue), the remote sensing image was processed on-orbit before the results were downloaded. As a result, the transmission volume is smaller, and the timeliness is improved. In summary, based on the micro-satellite constellations and their on-board processing capabilities, we can build a highly time-efficient global and all-day earth observation system. Therefore, the on-board processing capability of a single video satellite becomes very important.

For the recognition of aircraft activities at airports, the onboard processing of aircraft detection and tracking must be solved first. However, this is not easy, as there are two challenges: the accuracy and efficiency requirements of the algorithm.

In satellite imagery, objects of interest are often small and possibly densely clustered. Typically, the image GSD of video micro-satellites is around one meter. This means that the common civil aircraft will only be approximately 50 × 50 pixels large in extent. Thus, targets have limited details in remote sensing images. Additionally, the objects can have arbitrary orientations when viewed from overhead [1]. Considering the above aspects, the performance requirement of an aircraft detection algorithm is relatively high. The algorithm should have enough recall rate and precision, as well as good positioning accuracy.

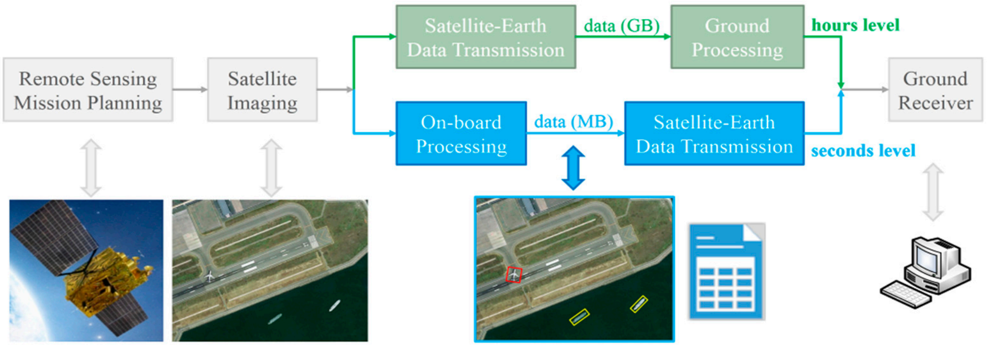


Figure 1. Comparison of remote sensing methods between ground processing and on-orbit processing.

In recent years, deep learning models, especially deep convolutional neural networks (DCNNs), have become state-of-the-art for many practical vision problems [2,3]. Many DCNN-based detectors have been proposed in the research field of object detection. DCNN-based detectors can be divided into two types: two-stage object detectors [4–7] and single-stage object detectors [8–11]. Two-stage object detectors have achieved promising results on common benchmarks; however, their training process is complex and computationally expensive. In contrast, single-stage object detectors are currently state-of-the-art with respect to the trade-off between speed and accuracy. For many practical applications, objects with large aspect ratios, dense distributions, and various orientations need to be detected; for example, in scene text detection or remote sensing object detection, objects can be arbitrarily oriented. The above-mentioned horizontal detectors have fundamental limitations for these applications. Thus, many rotation detectors [12,13] based on a general detection framework have been proposed. In remote sensing object detection, many rotation detectors have been proposed [14–17] and have achieved promising performances on large-scale public datasets for aerial images, e.g., DOTA [18], HRSC2016 [19], and OHD-SJTU [20]. Among these remote sensing targets, the detection of aircraft achieved almost the best accuracy.

For object tracking, most modern computer vision approaches are based on tracking-by-detection [21], where a set of possible objects are first identified via object detectors. Based on the detection results from each frame, a Kalman filter, particle filter, and probabilistic model are often used to accomplish accurate and consecutive object tracking. Meanwhile, deep learning methods have also attracted considerable interest in the visual tracking community as robust visual trackers. According to their architectures, state-of-the-art deep learning-based visual tracking methods are categorized as convolutional neural networks (CNNs) [22,23], Siamese neural networks (SNNs) [24,25], recurrent neural networks (RNNs) [26], or generative adversarial networks (GANs) [27]. Although these state-of-the-art methods have achieved significant progress, they are still not reliable for real-world applications.

To achieve on-board detection and tracking of aircraft in a remote sensing video, the usual tracking-by-detection methods are not feasible. This is because each detection procedure is computationally intensive, and the computing and memory resources of micro-satellites are limited. However, the excellent results achieved by rotation detectors in aircraft detection can serve as a good initial input for the tracking procedure. Thus, in this paper, an aircraft target tracking algorithm based on rotation detectors and image matching is proposed. First, aircraft detection was performed on the first incoming remote sensing image using a robust DCNN-based object detection model. Then, a multi-target tracking model based on image matching was proposed for efficient aircraft tracking. Finally, incorporating geospatial information, the aircraft activity was recognized in terms of its motion speed.

The main contributions of this paper are as follows:

1. Since the high-value activities of aircraft at an airport usually occur on the runway, we focus on detecting aircraft in the area around the runway. The smaller the detection area, the faster the detection procedure. Aircraft detection was implemented using a rotation detector named R3det.
2. The tracking algorithm can effectively cope with various challenging situations, such as the negative influence of various backgrounds and lighting conditions, self-rotation of aircraft, and aircraft entering and exiting.
3. Combining the results of aircraft tracking and geospatial information, aircraft activities were divided into parking, taxiing, and flying in terms of the aircraft's motion speed. The satellite can selectively save and download the video data of interest according to the activity recognition results, reducing the amount of satellite-to-ground data transmission.
4. The algorithm was verified to be efficient and effective using remote sensing videos from commercial micro-satellites.

The remainder of this paper is organized as follows. In Section 2, we outline our algorithm. In Section 3, the deep convolutional neural network (DCNN) based aircraft detection is introduced. Aircraft tracking and activity recognition is introduced in Section 4. In Section 5, we evaluate our algorithm on real-world datasets which contain a variety of aircraft activities and provide the experimental results and a discussion. Finally, we present the conclusion in Section 6.

2. Algorithm Overview

As shown in Figure 2, the algorithm is mainly composed of two parts: aircraft detection (at the start, highlighted in yellow), and aircraft tracking and activity recognition (highlighted in blue).

A DCNN-based rotation detector, named R3Det, was used for aircraft detection at airports. Considering that the high-value activities of aircraft at airports usually occur on the runway, we focused on detecting aircraft in the area around the end of the runway. For satellites operating in staring mode, the location of the airport runway area in the image can be easily determined by combining satellite orbit, satellite attitude, and ground geographic information. The smaller the detection image size, the faster the detection speed. We can effectively reduce the time-consuming nature of detection in the start frame by focusing on detecting aircraft in the area around the end of the runway.

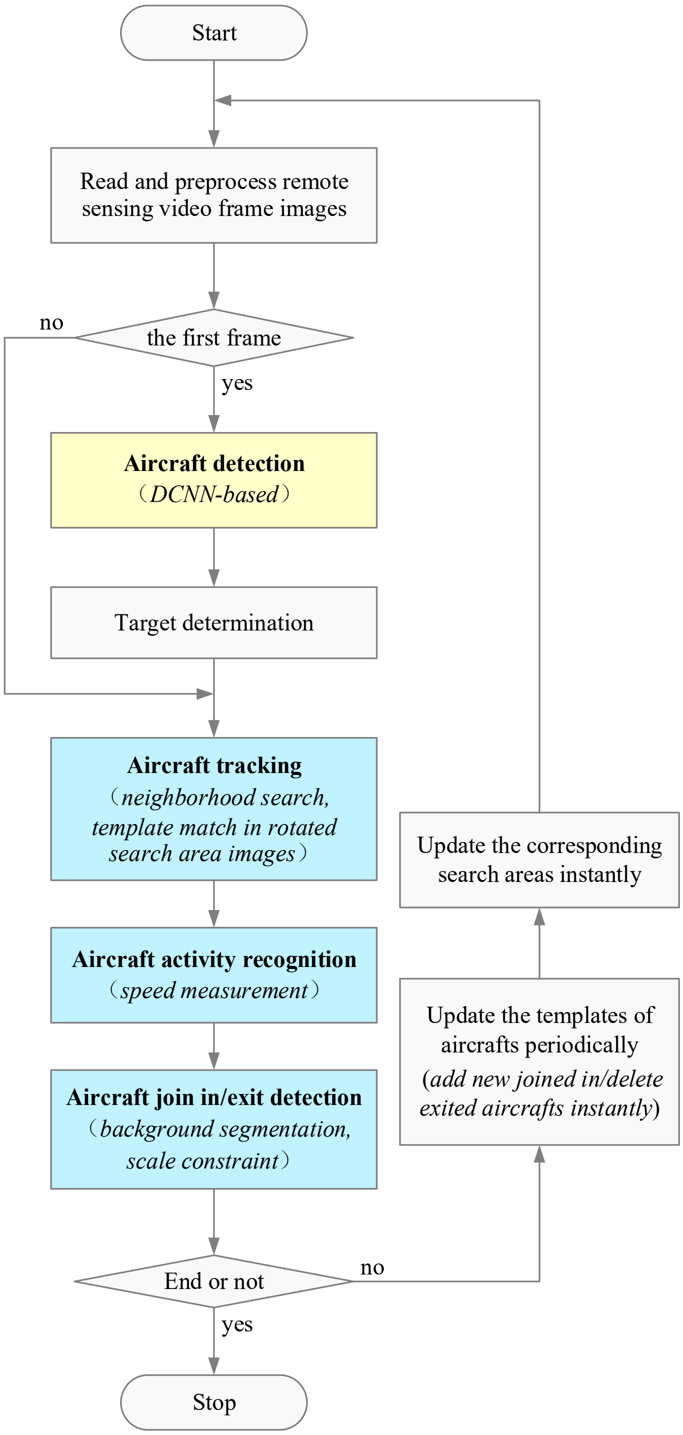

Start
Read and preprocess remote sensing video frame images
the first frame
no
yes
Aircraft detection
（*DCNN-based*）
Target determination
Aircraft tracking
（*neighborhood search, template match in rotated search area images*）
Aircraft activity recognition
（*speed measurement*）
Aircraft join in/exit detection
（*background segmentation, scale constraint*）
End or not
no
yes
Stop
Update the corresponding search areas instantly
Update the templates of aircrafts periodically
(*add new joined in/delete exited aircrafts instantly*)

Figure 2. Flow chart of the algorithm.

Due to the powerful rotation detector, the aircraft in the start frame are reliably detected and located with accurate envelopes. An efficient image matching method is then used to track the aircraft in subsequent frame images. The aircraft images detected in the start frame are used as the reference templates. The tracking procedure aims to find a part with the same size and the highest matching similarity as the template image within a certain search area in the current frame image. There are generally two approaches to solve this problem: grey value-based or feature-based matching. Compared with feature-based matching methods, such as edge-based matching [28], grey-based matching methods can maintain high robustness in the face of changing lighting and background conditions. Thus, the NCC-based [29] image matching method was used for tracking in this study. The corresponding search area image of each template was rotated within a certain angle range to generate multiple search area images. Matching of the template was then performed within these search area images. This can help find accurate matching results for some rotating aircraft targets, such as those that make turns.

The tracking process obtains the position of each aircraft in the image, as well as its movement over time. Combined with the resolution of the satellite image and the acquisition of time information, we can easily obtain the speed (km/h) of the aircraft. Depending on the speed of each aircraft, its activity can be classified as parking, taxiing, or flying.

Aircraft entry and exit detection is performed around the borders of the image. Aircraft entry detection is performed through background segmentation and under a size constraint. Aircraft exit detection is a comprehensive judgment based on the disappearance of the aircraft, as well as its position and speed when it disappeared.

At the end of each loop, each template image and its corresponding search area are updated. Templates are periodically updated to cope with the influence of a changeable background. The next search area corresponding to each aircraft is instantly updated according to its current tracking result. This cycle continues, completing aircraft target tracking and activity recognition of subsequent images in turn.

3. DCNN-Based Aircraft Detection

DCNN-based rotation detectors have achieved good detection results on existing remote sensing datasets, especially for aircraft object detection. In this study, a rotation detector named R3Det [14] was used for aircraft detection in the start frame. Figure 3 shows the flowchart of aircraft detection based on the R3Det detector. The ResNet network [30] was used as the R3Det detector's backbone. Upon this, the feature pyramid network (FPN) and the feature refinement module (FRM) were constructed for rotation target prediction. The R3Det detector was fine-tuned with the DOTA1.0 dataset upon the pre-trained ResNet152 model. When detecting aircraft, an input image of an arbitrary size was divided into manageable slices (1100 × 1100 pixels) and each slice underwent the detecting process with our trained model [1]. Partitioning takes place via a sliding window with the slice's size and overlap (10% by default), as shown in Figure 3.

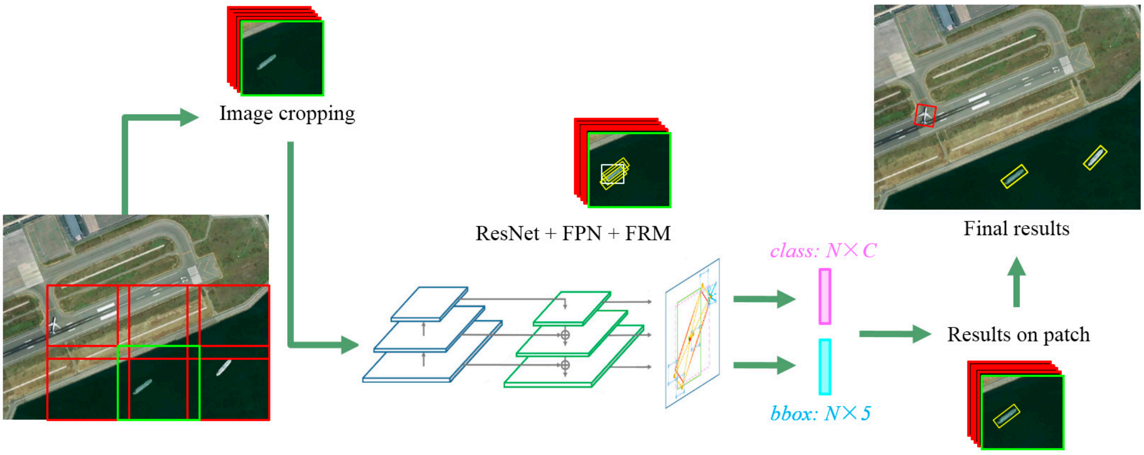


Figure 3. Flowchart of aircraft detection (R3Det-based).

We tested the aircraft detection detectors with 10 remote sensing images of different airports. The main parameters of the remote sensing images used for testing are shown in Table 1. These remote sensing images have the following characteristics.

- Includes multiple ground sampling distance (GSD) metrics from 0.2 m to 1.0 m.
- The images contain multi-type aircraft of different sizes. The orientation, distribution, and painting type of the aircraft are arbitrary. Military and civil aircraft are all included.
- The image includes areas such as airport runways, apron, etc.
- Imaging conditions include no clouds and thin clouds.

Table 1. Properties of each test remote sensing image and its test results.

Properties of Test Remote Sensing Images				Test Results		
Image	Size (Pixel)	GSD (m)	Number of Instances	TP	FN	FP
IMG1	3203 × 1370	<0.2	28	28	0	0
IMG2	4346 × 1199	<0.2	54	54	0	0
IMG3	3328 × 3072	<0.3	24	24	0	0
IMG4	3018 × 1065	<0.3	27	27	0	0
IMG5	2413 × 5118	<0.3	30	30	0	0
IMG6	3193 × 5540	<0.3	33	33	0	1
IMG7	2318 × 1914	<0.3	101	100	1	0
IMG8	4347 × 1202	<0.3	154	152	2	1
IMG9	4096 × 3584	<0.5	54	54	0	0
IMG10	7168 × 4096	<1.0	113	110	3	4

The test results are shown in Table 1. TP (true positive) represents a correct detection result. FN (false negative) represents a missed detection result. FP (false positive) represents false detected results. Figure 4 shows the recall and precision for each test sample. Through analysis of the results, we concluded that:

(1) The smaller the ground sampling distance (GSD), the better the detection performance of aircraft targets.

(2) Missed detections mainly occurred in the detection of aircraft parked next to each other on the tarmac.

(3) False positives are helicopters wrongly detected as aircraft in low-resolution images.

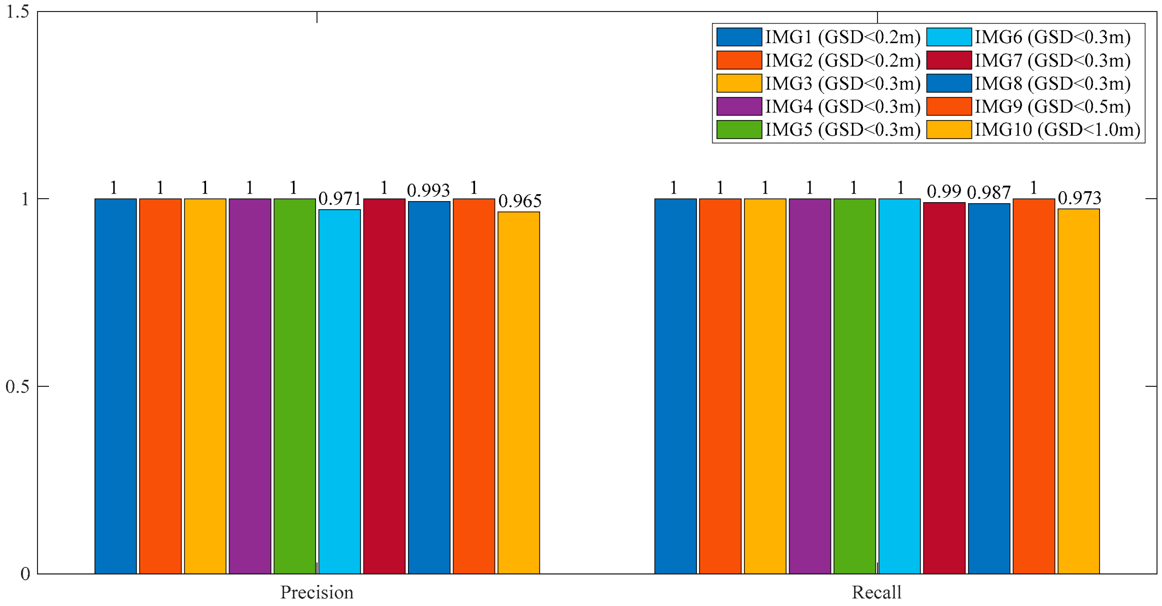


Figure 4. Precision and recall results of each test remote sensing image.

Figure 5 shows some of the detection results. All aircraft in these images were accurately detected with confidences larger than 0.8. Both military (red dotted ellipse) and civilian aircraft (blue dotted ellipse) were accurately detected. Aircraft were also accurately detected in remote sensing images affected by thin clouds and fog, as in Figure 5a,b. Aircraft densely parked on the tarmac (blue dotted ellipse), as well as aircraft taxiing on the runway (white dotted ellipses) were also accurately detected. Additionally, partially occluded aircraft in high-resolution remote sensing images were accurately detected (yellow ellipse).

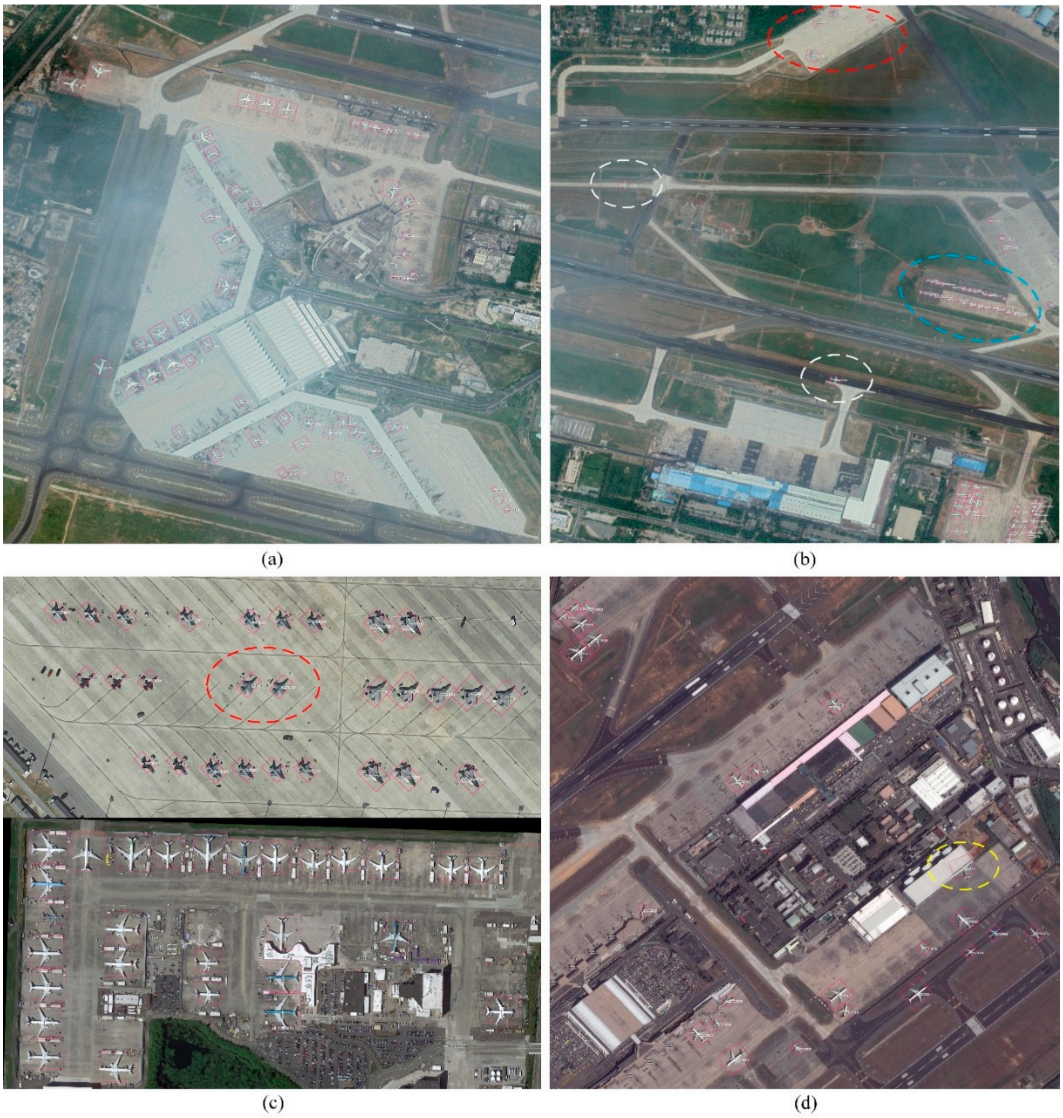


Figure 5. Aircraft detection results (confidence larger than 0.8) in test remote sensing images. (**a**) Aircraft detection results under thin cloud weather conditions; (**b**) Aircraft detection results on the apron and runway under thin cloud weather conditions; (**c**) Detection results of different types of military and civilian aircrafts; (**d**) Partially occluded aircraft can also be accurately detected.

As outlined in Figure 2, results from the detection procedure were used as the input for the tracking and activity recognition procedure. At airports, high-value activities mainly occur on the runway and its surrounding areas. The tests verified that the algorithm has better detection performance on the runway and its surrounding areas. Therefore, in follow-up research on aircraft target tracking algorithms, we can focus on processing only the runway and its surrounding areas to ensure reliable detection of aircraft targets. This can not only reduce the amount of calculation for remote sensing image processing, but also improve the timeliness of aircraft activity recognition. The aircraft detector output results with high recall and accuracy, providing excellent initial conditions for subsequent aircraft tracking.

4. Aircraft Tracking and Activity Recognition

4.1. Flow of Aircraft Tracking and Activity Recognition

Using the DCNN-based rotation detector to detect the start frame, we identified the aircraft that needed to be tracked. To use the image matching method for aircraft tracking, we first obtained the template image of the aircraft from the rotation detection results. The aircraft in the template image is enveloped by a non-rotating bounding box, as shown in Figure 6. The width and height of the rotation bounding box (purple rectangle) of the detected aircraft are w^{R} and h^{R}, respectively. The width and height of its corresponding horizontal bounding box (blue rectangle) are w^{H} and h^{H}, respectively. Thus, for the aircraft, the width of its template image is $w^{\mathrm{tpl}} = w^{\mathrm{H}} - \sqrt{w^{\mathrm{H}}h^{\mathrm{H}} - w^{\mathrm{R}}h^{\mathrm{R}}}/2$, and the height is $h^{\mathrm{tpl}} = h^{\mathrm{H}} - \sqrt{w^{\mathrm{H}}h^{\mathrm{H}} - w^{\mathrm{R}}h^{\mathrm{R}}}/2$.

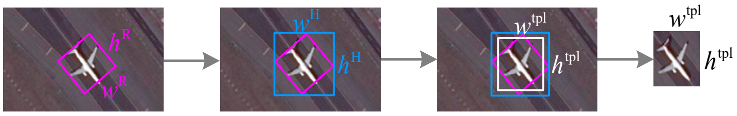


Figure 6. Determining the initial aircraft template image from the rotation detection results in the start frame.

The next process is to locate each aircraft in subsequent images and recognize its activity type. Figure 7 shows the algorithm for aircraft tracking and activity recognition. The algorithm processes each frame image in turn. In each loop, the algorithm mainly accomplishes four tasks.

Firstly, the algorithm searches for the best matching image for each aircraft within a certain search area. The search area is calculated based on the previous tracking result of the aircraft. The search area is centered on the aircraft and is twice its size. Before matching, the search area image with a certain step size is rotated within a certain angle range to obtain a batch of search area images. The best match for the aircraft target is then identified from this batch of images and the matching result is located. Finally, we identify the position of the matching result in the current input image and achieve tracking for the target aircraft. This matching method is not only suitable for aircraft with translation and turning motion, but can also effectively prevent mistaken tracking of nearby aircraft.

Secondly, the moving speed of each tracked aircraft is calculated using the multi-cycle backward difference method. The recognition result of the aircraft activity is then given according to its movement speed.

Thirdly, detection of newly entering objects at the edge of the field of view is performed based on the background subtraction method and size constraint. If the bounding box of the newly added target does not overlap with all the tracking results, it can be judged as another aircraft that needs to be tracked.

Additionally, the template image of the aircraft is periodically updated using the latest tracking results. Thus, the tracking procedure will have high adaptability to changeable or complex backgrounds.

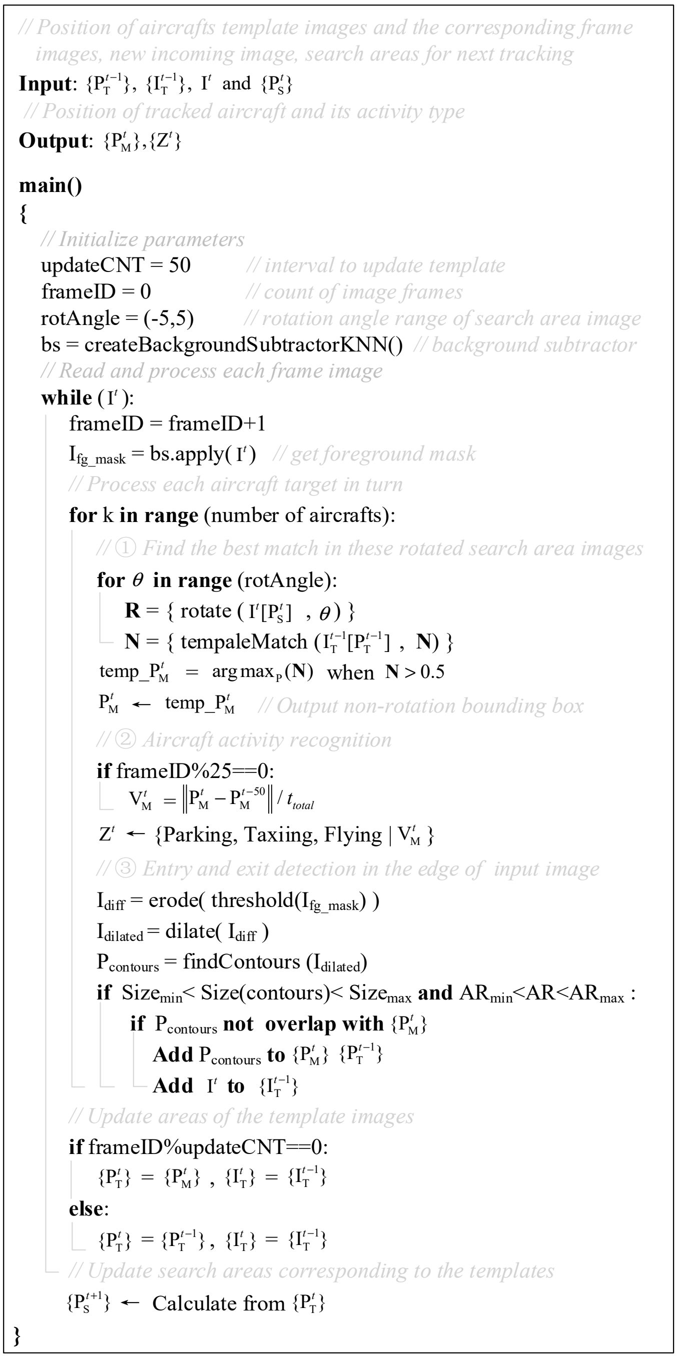
4.2. NCC-Based Template Matching for Aircraft Tracking

The normalized correlation coefficient (NCC) [29] based image matching method was used to achieve aircraft tracking. The template matching algorithm consists of two steps. First, both the aircraft template image and the image of the search area were normalized according to the following equations.

$$T'(x,y) = \frac{T(x,y) - \frac{1}{w_T \times h_T}\left(\sum\limits_{0\le x\le w_T, 0\le y\le h_T} T(x,y)\right)}{\sqrt{\sum\limits_{0\le x\le w_T, 0\le y\le h_T} T(x,y)^2}} \tag{1}$$

$$S'(x,y) = \frac{S(x,y) - \frac{1}{w_S \times h_S}\left(\sum_{0 \le x \le w_S, 0 \le y \le h_S} S(x,y)\right)}{\sqrt{\sum_{0 \le x \le w_S, 0 \le y \le h_S} S(x,y)^2}} \quad (2)$$

Here, $T(x,y)$ and $S(x,y)$ are the pixel values in the aircraft template image and search area image, respectively. $T'(x,y)$ and $S'(x,y)$ are the pixel values in the normalized aircraft template image and search area image, respectively. w_T, h_T are the width and height of the aircraft template image, and w_S, h_S are the width and height of the search area image. The normalized correlation coefficients of these two normalized images were then calculated.



```
// Position of aircrafts template images and the corresponding frame
   images, new incoming image, search areas for next tracking
Input: $\{P_T^{t-1}\}$, $\{I_T^{t-1}\}$, $I^t$ and $\{P_S^t\}$
 // Position of tracked aircraft and its activity type
Output: $\{P_M^t\}$,$\{Z^t\}$

main()
{
   // Initialize parameters
   updateCNT = 50          // interval to update template
   frameID = 0             // count of image frames
   rotAngle = (-5,5)       // rotation angle range of search area image
   bs = createBackgroundSubtractorKNN() // background subtractor
   // Read and process each frame image
   while ($I^t$):
      frameID = frameID+1
      $I_{fg\_mask}$ = bs.apply( $I^t$)  // get foreground mask
      // Process each aircraft target in turn
      for k in range (number of aircrafts):
         // ① Find the best match in these rotated search area images
         for $\theta$ in range (rotAngle):
            $\mathbf{R}$ = { rotate ( $I^t[P_S^t]$ , $\theta$) }
            $\mathbf{N}$ = { tempaleMatch ($I_T^{t-1}[P_T^{t-1}]$ , $\mathbf{N}$) }
         $temp\_P_M^t = \arg\max_P(\mathbf{N})$ when $\mathbf{N} > 0.5$
         $P_M^t \leftarrow temp\_P_M^t$  // Output non-rotation bounding box
         // ② Aircraft activity recognition
         if frameID%25==0:
            $V_M^t = \left\| P_M^t - P_M^{t-50} \right\| / t_{total}$
         $Z^t \leftarrow$ {Parking, Taxiing, Flying | $V_M^t$ }
         // ③ Entry and exit detection in the edge of  input image
         $I_{diff}$ = erode( threshold($I_{fg\_mask}$) )
         $I_{dilated}$ = dilate( $I_{diff}$ )
         $P_{contours}$ = findContours ($I_{dilated}$)
         if $Size_{min}$< Size(contours)< $Size_{max}$ and $AR_{min}$<AR<$AR_{max}$ :
            if $P_{contours}$ not  overlap with $\{P_M^t\}$
               Add $P_{contours}$ to $\{P_M^t\}$ $\{P_T^{t-1}\}$
               Add  $I^t$ to  $\{I_T^{t-1}\}$
      // Update areas of the template images
      if frameID%updateCNT==0:
         $\{P_T^t\} = \{P_M^t\}$ , $\{I_T^t\} = \{I_T^{t-1}\}$
      else:
         $\{P_T^t\} = \{P_T^{t-1}\}$, $\{I_T^t\} = \{I_T^{t-1}\}$
      // Update search areas corresponding to the templates
      $\{P_S^{t+1}\} \leftarrow$ Calculate from $\{P_T^t\}$
}
```

Figure 7. Algorithm for aircraft tracking and activity recognition.

$$R(x_0, y_0) = \sum_{0 \leq \hat{x} \leq w_T, 0 \leq \hat{y} \leq h_T} \left(T'(\hat{x}, \hat{y}) * S'(x_0 + \hat{x}, y_0 + \hat{y}) \right) \tag{3}$$

Here, $(\hat{x}, \hat{y})$ are the coordinates in the normalized aircraft template image, (x_0, y_0) are the coordinates of the upper left corner of the square matching area in the normalized search area image, and $R(x_0, y_0)$ is the value of the normalized correlation coefficient (NCC). The closer the value $R(x_0, y_0)$ is to one, the more similar the template image and the image in the matching area of the search area image are.

On this basis, an aircraft tracking algorithm based on neighborhood search and matching was proposed. Figure 8 shows the main flow of each tracking procedure, which consists of four steps.

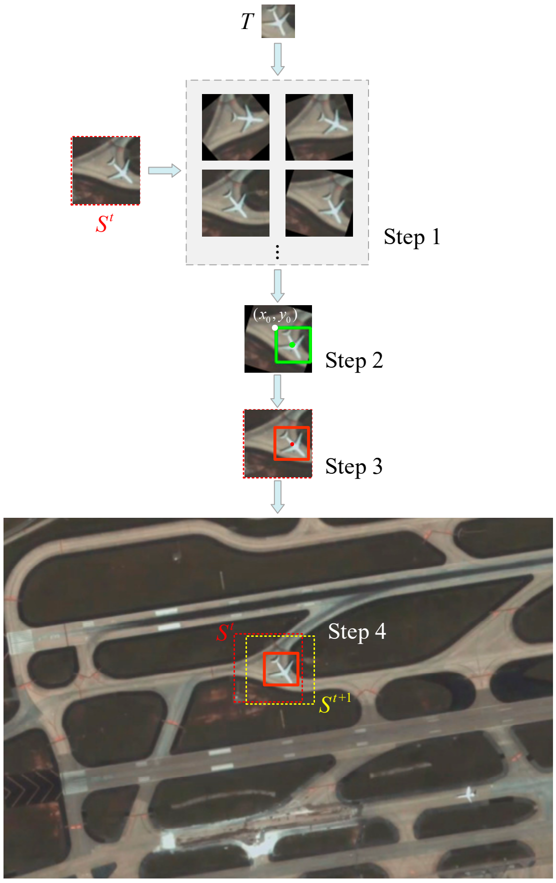


Figure 8. Tracking procedure for matching aircraft in rotated search area images.

Step 1: To track aircraft in an airport, the matching algorithm needs to be able to adapt to the matching of aircraft in translational and rotational motions. Thus, the search area image was first rotated with a certain step size (2°) within a certain angle range ($-10°$~$10°$) to obtain a batch of search area images. Then, NCC-based matching was performed between the aircraft template image T and all obtained search area images. We chose to rotate the search area image instead of rotating the aircraft template image, because rotating the template image generates invalid data at its edges, which seriously affects the matching result.

Step 2: The best match is identified as the matching result with the largest normalized cross-correlation coefficient. The coordinates at the center point of the best matching area in the search area image (the green center point) are identified according to the matching result.

Step 3: The search area image that contains the best match is transformed back to its original (non-rotated) state. Then, the coordinates at the center of the best matching area can be obtained, represented in the coordinate system of the original search area image (the red center point). Finally, the center point and the size of the aircraft template image are combined, and the matching result can be located in the original search area image (the red rectangle).

Step 4: The position of the matching result (the red rectangle) in the current input image is calculated; this is the tracking result of the aircraft in the current frame. According to the current tracking result, the search area for the next tracking procedure (yellow dotted rectangle) can be calculated. As outlined in Figure 7, the aircraft template image is updated using the tracking result every 50 tracking cycles.

Based on the combination of rotated image matching and a neighborhood search, the tracking process can achieve good performance. On the one hand, based on the design of rotated image matching, the algorithm enables stable tracking of aircraft with translational and turning motions. During the aircraft tracking procedure of our algorithm, the aircraft search area is only twice the size of the currently tracked aircraft. Usually, there is a certain safety distance between aircraft targets in their working state. Therefore, based on the neighborhood search and the safe working distance requirement of aircrafts, nearby aircraft do not enter the search area of the currently tracked aircraft. The algorithm can effectively prevent false tracking of nearby aircraft.

4.3. Aircraft Activity Recognition

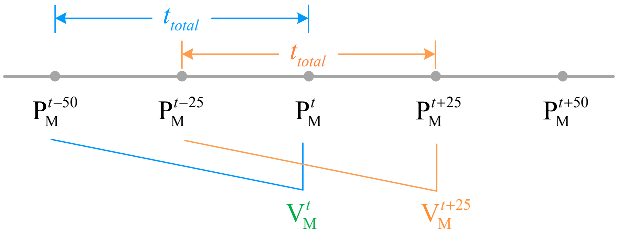
The velocity of each tracked aircraft was calculated every 25 frames using the multi-cycle backward difference method. As shown in Figure 9, the current speed was taken to be the total moving distance over the latest 50 frames divided by the total time, t_{total}. The calculation formula is as follows.

$$\mathrm{V}_{\mathrm{M}}^{t} = \frac{\|\mathrm{P}_{\mathrm{M}}^{t} - \mathrm{P}_{\mathrm{M}}^{t-50}\|}{t_{total}} \tag{4}$$

Here, $\mathrm{V}_{\mathrm{M}}^{t}$ is the current speed, $\mathrm{P}_{\mathrm{M}}^{t}$ is position of the tracked aircraft at the current frame, $\mathrm{P}_{\mathrm{M}}^{t-50}$ is the position of the aircraft 50 frames away from the current frame, and t_{total} is the time elapsed while the aircraft moved from position $\mathrm{P}_{\mathrm{M}}^{t-50}$ to $\mathrm{P}_{\mathrm{M}}^{t}$. Thus, we can obtain a relatively stable speed measurement of the aircraft every other 25 frames. Finally, the recognition result of the aircraft's activity can be given based on the speed of the aircraft:

$$\begin{cases} \text{Parking} \quad 0 \le \mathrm{V}_{\mathrm{M}}^{t} \le (2 \times \mathrm{GSD} \times 3.6) \quad \mathrm{km/h} \\ \text{Taxiing} \quad (2 \times \mathrm{GSD} \times 3.6)\ \mathrm{km/h} < \mathrm{V}_{\mathrm{M}}^{t} \le 230 \quad \mathrm{km/h} \\ \text{Flying} \quad 230\ \mathrm{km/h} < \mathrm{V}_{\mathrm{M}}^{t} \le 1100\ \mathrm{km/h} \end{cases} \tag{5}$$

Here, we considered the possible positioning error caused by matching aircraft in the rotated search area image. If the movement of the aircraft is less than two pixels, the aircraft is considered to be in the parking state. If its speed exceeds 230 km/h, the aircraft is considered to be in the flying state. In other cases, it is considered to be in the taxiing state.



t_{total}
t_{total}
$\mathrm{P}_{\mathrm{M}}^{t-50}$
$\mathrm{P}_{\mathrm{M}}^{t-25}$
$\mathrm{P}_{\mathrm{M}}^{t}$
$\mathrm{P}_{\mathrm{M}}^{t+25}$
$\mathrm{P}_{\mathrm{M}}^{t+50}$
$\mathrm{V}_{\mathrm{M}}^{t}$
$\mathrm{V}_{\mathrm{M}}^{t+25}$

Figure 9. Aircraft activity differentiated according to its speed of movement.

4.4. Aircraft Entry and Exit Detection

For satellites operating in staring mode, the airport background changes slightly in the acquired video images. Therefore, we use the background subtraction method [31] to detect new incoming aircraft. To further reduce the amount of computation, we only detect new incoming aircraft around the edges of the image. The size of the edge range for the detection operation is equal to the size of the largest aircraft in the current field of view. Aircraft entry detection includes the following five operations:

1. Build a KNN-based background subtractor for background detection.
2. Obtain the foreground mask image and perform image binarization processing.
3. Morphologically process candidates in the foreground region based on the rectangular kernel.
4. Eliminate obviously wrong results within dimensional constraints. A target whose size is smaller than 90% of the smallest aircraft size or larger than 110% of the largest aircraft size in the current field of view is considered to be obviously wrong.
5. Judge whether the candidate result is the tracked aircraft that has moved into the edge area. This can be easily achieved by judging whether their bounding boxes overlap.

Aircraft exit detection is simpler. If an aircraft fails to be tracked for 25 consecutive frames, and its latest tracked position was in the image boundary area, the aircraft is considered to have exited.

5. Experiments and Results

5.1. Platform and Datasets

The algorithm proposed in this paper is only suitable for video satellites working in staring mode. Thus, we used remote sensing videos from commercial video satellites to perform experimental validation of the proposed algorithm.

From Section 3 we know that the smaller the GSD (ground sample distance) of remote sensing images, the better the detection performance of the algorithm. Considering the size differences of different types of aircraft, we suggest that the GSD of video remote sensing images should be approximately 1 m or less than 1 m. Therefore, two video instances of high resolution with a GSD of less than 1 m (0.92 m), and low resolution with a GSD of larger than 1 m (1.13 m), are used for verification here. Table 2 outlines the main parameters of the two remote sensing videos. These two videos have a frame rate of 25 fps but are of different image sizes. The scenes in both videos are at the end of an airport runway and its surrounding environment.

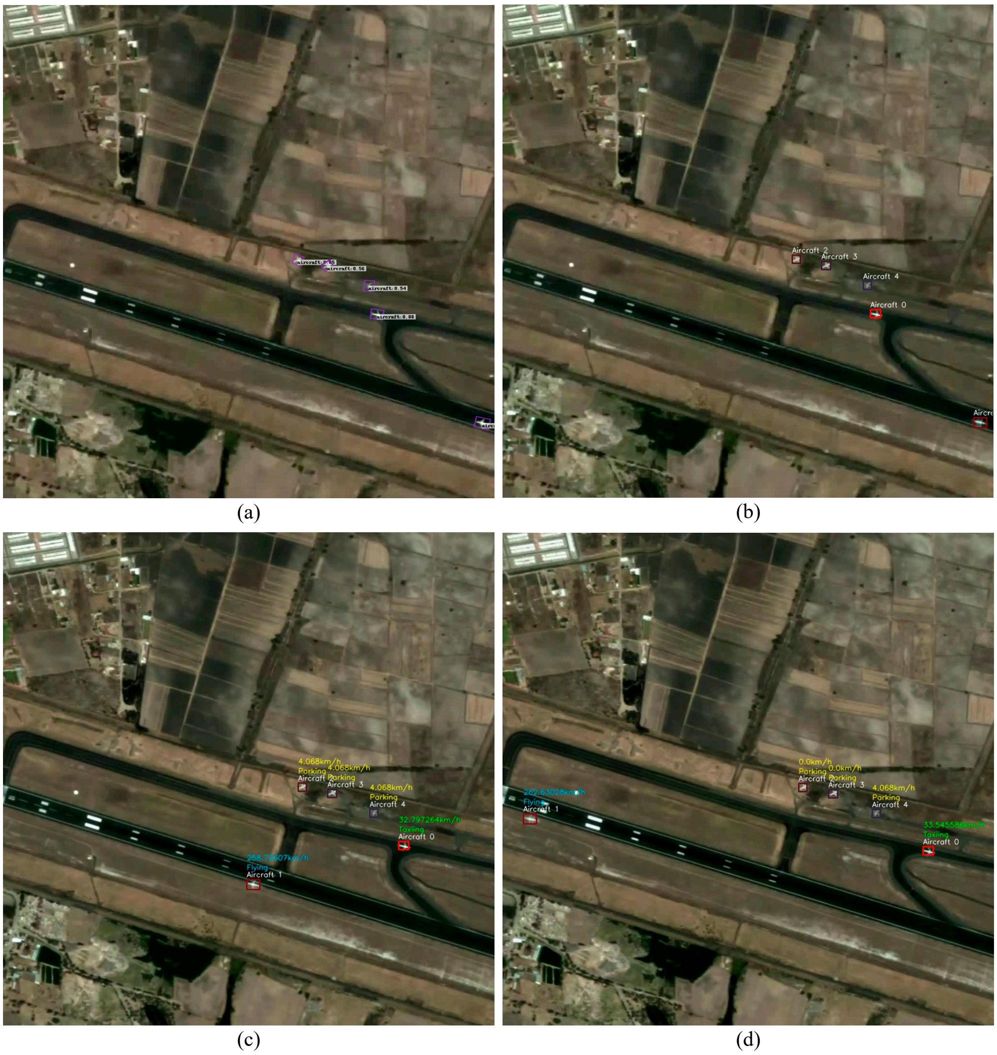
Table 2. Parameters of remote sensing videos.

Test Video	**Parameters**			
	GSD (m)	**Image Size (pixel)**	**Frame Rate (fps)**	**Lengt H(s)**
Video 1	1.13	1024 × 1024	25	14
Video 2	0.92	1100 × 1024	25	28

The complete algorithm was deployed and tested on an embedded AI processor called NVIDIA Jetson AGX Xavier.

5.2. Experimental Results

Figure 10 shows some screenshots of the experimental results on test video 1. The image shows the end of an airport runway and its adjacent farmland. There are five aircraft in the image. The three aircraft parked at the top of the picture are relatively small in size. Some of the aircraft have tones similar to their nearby environmental backgrounds and are not clearly visible in the image.



(a) (b)

(c) (d)

Figure 10. Screenshots of the experimental results from test Video 1. (**a**) Aircraft detection results in the first frame of the video; (**b**) Aircrafts detected in the starting frame are all tracked stably; (**c**) The speed of each tracked aircraft is measured and from this the type of aircraft activity is determined; (**d**) Each aircraft is stably tracked and its type of activity is determined.

Figure 10a shows the aircraft detection results in the start frame. The aircraft were detected with the DCNN-based rotation detector described in Section 3. The five aircraft were all correctly detected in the start frame and located with purple bounding boxes. As shown in Figure 10b, the five detected aircraft were treated as targets that needed to be tracked in the following frames. These aircraft were assigned different identification numbers and were located using different colored bounding boxes. In subsequent frames, the algorithm tracked the five aircraft using image matching and measured their speed to identify the type of activity. The measured speed and recognized activity status of each aircraft were updated every 25 frames. The current activity status of the aircraft was displayed using different colors: parking was displayed in yellow, taxiing was displayed in green, and flying was displayed in blue. As shown in Figure 10c,d, all five aircraft were continuously and correctly tracked. Three aircraft (Aircraft 2, 3, and 4) that were parked on the tarmac were correctly identified to be parking. Aircraft 0, taxiing on the runway, was correctly recognized to be taxiing. Aircraft 1 was correctly recognized to be flying.

Figure 11 shows the speed vs. time curves for every tracked aircraft in Video 1. Curves of five different colors represent the five different aircraft targets. The two horizontal dashed lines in the figure represent the speed thresholds used to divide the three different aircraft activities. When the aircraft speed is in the lowest zone, the state of the aircraft is parking. When the aircraft speed is in the middle zone, the state of the aircraft is taxiing. When the aircraft speed is in the upper zone, the state of the aircraft is flying.

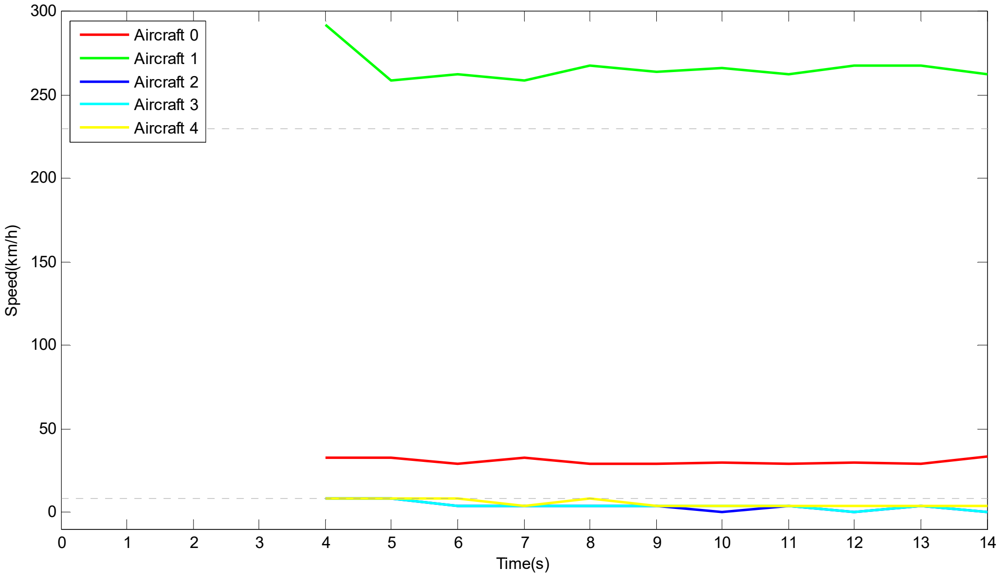


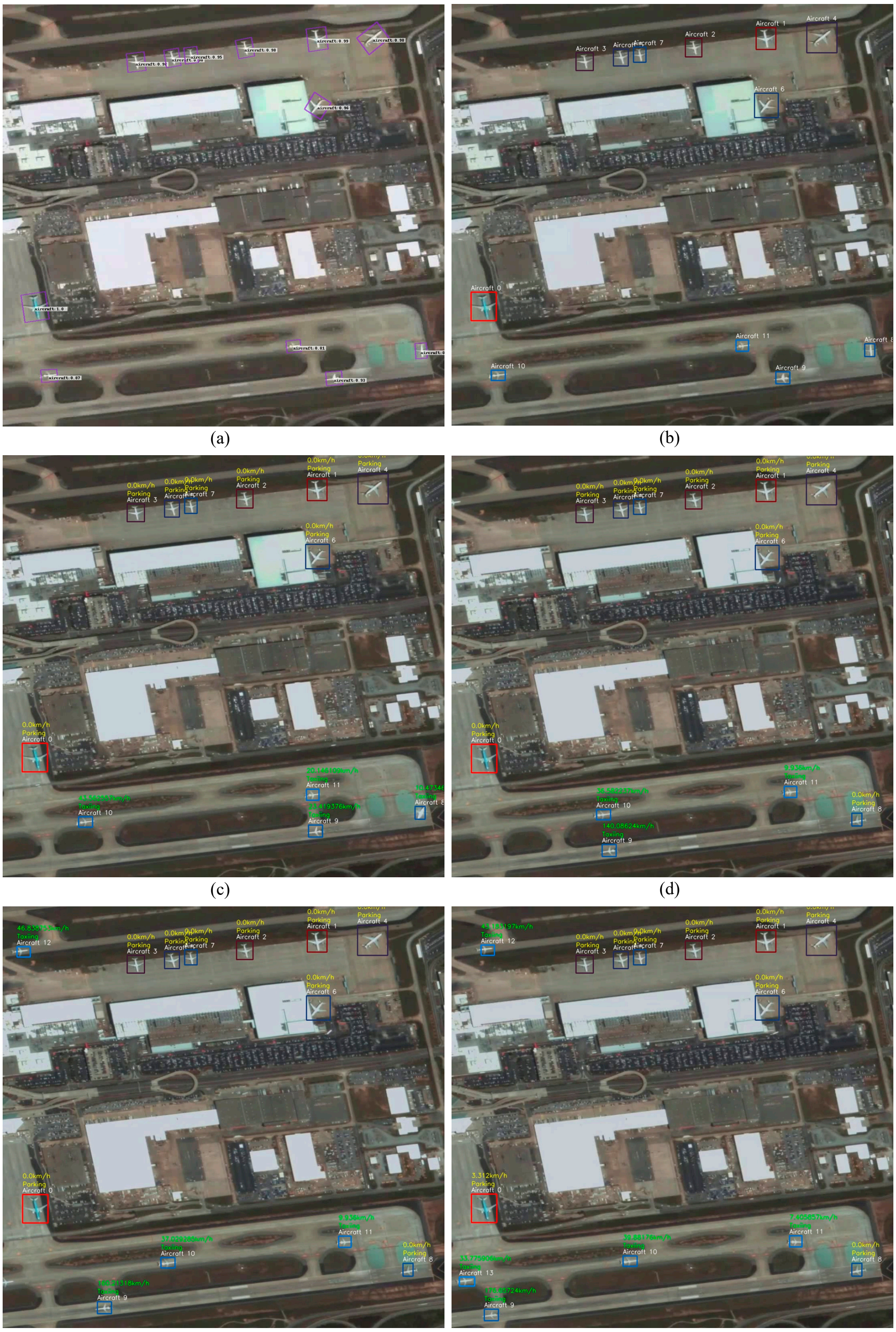
Figure 11. Speed measurements for each aircraft in test Video 1.

From the experimental test results in Video 1, the algorithm is identified as having the following advantages. Firstly, the algorithm supports the reliable detection and stable tracking of weak and small aircraft targets (e.g., Aircraft 4). Secondly, the algorithm can stably track aircraft at different speeds. Thirdly, the algorithm can accurately track aircraft targets under complex and changeable backgrounds (e.g., Aircraft 1). Lastly, based on the aircraft tracking and speed measurement, the algorithm can further identify the current activity status of the aircraft.

In Figure 12, some screenshots of the experimental results from Video 2 are displayed. The images show the end of an airport runway and its adjacent apron. There are more aircraft in the remote sensing images, and they are of various shapes and sizes. In addition, the motion of the aircraft in the remote sensing images is more complex.

Figure 12a shows the aircraft detection results in the start frame. All twelve aircraft in the image were accurately detected and located with purple bounding boxes. As shown in Figure 12b, the twelve detected aircraft were treated as targets that needed to be tracked in the following frames. They were assigned different identification numbers and were located using different colored bounding boxes. In subsequent frames, each aircraft was tracked, and its activity status was given based on its velocity measurement. At the beginning, the top eight aircraft (Aircraft 0~Aircraft 7) were parked, and the bottom four aircraft (Aircraft 8~Aircraft 11) were queuing to take off (as shown in Figure 12c). Later, the first aircraft on the runway took off, and the next aircraft turned around and was ready to take off (Figure 12d). At the end of the video, two taxiing aircraft enter from the left side of the field of view (Figure 12e,f).

Figure 13 shows the speed vs. time curves for every tracked aircraft in Video 2, similar to Figure 11. Curves of fourteen different colors represent the fourteen different aircraft targets in Video 2. At the 25th second, the algorithm obtains the speed measurement of the newly entered Aircraft 12. Similarly, the algorithm obtains the speed measurement of the newly entered Aircraft 13 at the 27th second.



(a) (b) (c) (d) (e) (f)

Figure 12. Screenshots of the experimental results from test Video 2. (**a**) Aircraft detection results in the first frame of the video; (**b**) Aircrafts detected in the starting frame are all tracked stably; (**c**) The speed of each tracked aircraft is measured and from this the type of aircraft activity is determined; (**d**) The aircraft is stable tracked even with complex environmental backgrounds, lighting conditions, and various movements of the aircraft, such as turning (Aircraft 8); (**e**) A newly entering aircraft (Aircraft 12) is also identified and stably tracked; (**f**) Another entering aircraft (Aircraft 13) is identified and stably tracked.

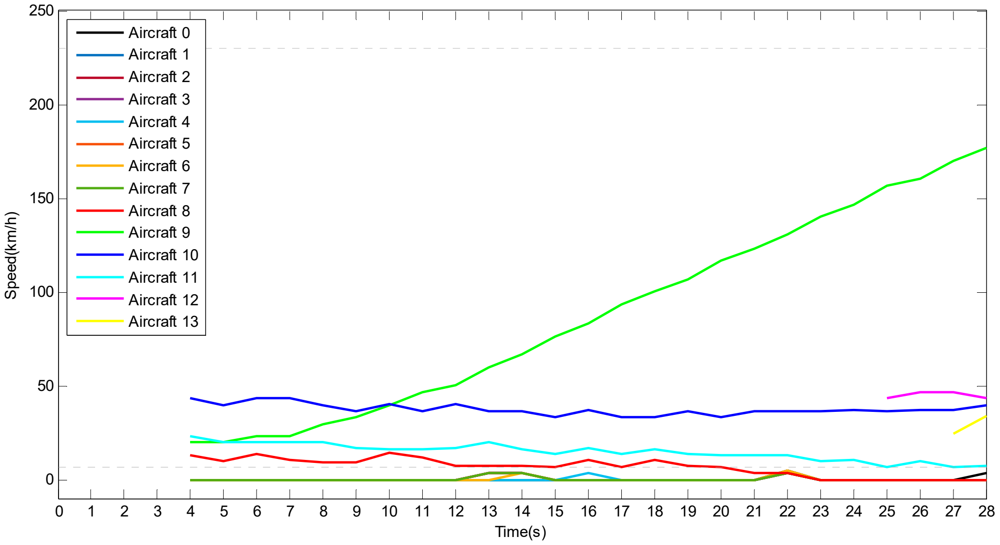


Figure 13. Speed measurements for each aircraft in test Video 2.

From the experimental results in Video 2, we can conclude that the algorithm has the following advantages. Firstly, the algorithm supports the stable tracking of turning aircraft (e.g., Aircraft 8) due to the improved rotation matching method. Secondly, the algorithm has strong robustness, even under complex lighting conditions. As shown in Figure 12c, there was a great deal of light reflected on the surface of Aircraft 8, but the algorithm can still track it stably. Lastly, the algorithm supports the detection of newly entered aircraft. As shown in Figure 12e,f, the newly entered Aircraft 12 and 13 were accurately detected. They were then accurately tracked, and their activity status was identified.

We also tested and compared the time-consumption of the algorithm on an RTX2080Ti-based server and on the embedded AI computing unit called Jetson AGX Xavier. Table 3 outlines the time-consuming statistics of the two experiments on different platforms. We evaluated the time consumption of the two main parts of the algorithm separately. DCNN-based aircraft detection takes approximately 4 s and 20 s on the server and embedded AI unit, respectively. Aircraft tracking and activity recognition takes less than 200 ms on both computing platforms. Video 2, which contains more aircraft targets, consumed more time. Compared with the traditional ground processing methods, the method in this paper significantly improves the efficiency of aircraft activity recognition and reduces the amount of satellite-to-ground data transmission. Although there is a delay of tens of seconds, the method in this paper can serve as an important reference for the exploration of on-board intelligent processing and satellite intelligent decision-making.

Table 3. Time consumption comparison of the algorithm on different computing platforms.

Test Data	Time Consumption			
	RTX2080Ti		Jetson AGX Xavier	
	Detection	Tracking and Activity Recognition	Detection	Tracking and Activity Recognition
Video 1	3.47 s	4.84 ms	19.33 s	18.27 ms
Video 2	3.76 s	51.9 ms	19.78 s	172.76 ms

5.3. Discussion

The algorithm proposed in this paper offers a solution for the recognition of aircraft activity at airports. By combining the DCNN-based object detection and an improved template matching method, aircraft can be accurately detected and stably tracked, and its activity can be identified based on its speed. The algorithm can process aircraft targets of different sizes, and is equally effective under different complex environmental backgrounds, lighting conditions, and various movements of the aircraft, such as turning, entering, and exiting.

The algorithm's time consumption has been evaluated on an embedded computing platform, where it exhibited a delay of tens of seconds in the two testing videos. However, compared with tracking-by-detection methods, the algorithm in this paper achieves a relatively efficient performance under the premise of ensuring high accuracy. Nonetheless, the algorithm can still effectively support applications in which short delays are acceptable, such as monitoring whether an airport is operating or not. Based on on-board dynamic event recognition, selective download of segment data of interest can be performed. Compared with traditional ground processing methods, the proposed method significantly improves the efficiency of event acquisition and reduces the amount of satellite-to-ground data transmission.

With appropriate improvements, the algorithm is expected to be used for time-critical tasks, such as real-time detection and tracking of take-off aircraft. These high-value dynamic events usually occur at the end of the runway. Therefore, geographic information can be incorporated to further narrow the range of images processed by the algorithm. By reducing the detection range and the number of aircraft in the field of view, the time consumption of the algorithm can be significantly reduced. Additionally, using satellites with on-board, real-time processing capabilities to form a constellation would enable the continuous real-time tracking of areas of interest and the relay tracking of moving targets.

6. Conclusions

The recognition of aircraft activity at airports is of great significance in both civil and military fields. In this paper, an efficient aircraft activity recognition algorithm, which supports deploying video micro-satellites, was proposed. Firstly, the aircraft in the start frame was detected with a robust rotation detector. Then, aircraft tracking was achieved using a neighborhood search and an improved rotating image matching method. Combining the tracking result and geospatial information, the speed of the aircraft was obtained. Finally, aircraft activity was classified as parking, taxiing, or flying according to the speed of the aircraft. We experimentally verified the algorithm on real remote sensing videos. The experimental results demonstrated the effectiveness and efficiency of the algorithm. When deployed on video micro-satellites with limited computing and memory resources, the algorithm can still effectively support applications where short delays are acceptable, such as monitoring whether an airport is operating. The scheme in this paper also provides a general framework for on-orbit target tracking and event recognition.

Author Contributions: Conceptualization, R.Z. and X.Z.; methodology, R.Z., X.Z. and L.X.; software, R.Z. and J.Q.; validation, R.Z. and L.X.; writing—original draft preparation, L.X. and J.Q.; writing—review and editing, R.Z. and X.Z.; visualization, J.Q.; funding acquisition, R.Z. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful to all reviewers for their thoughtful comments and suggestions.

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

Acronyms/Abbreviations

Ground sample distance (GSD), artificial intelligence (AI), deep convolutional neural network (DCNN), feature pyramid network (FPN), feature refinement module (FRM), normalized correlation coefficient (NCC), commercial off-the-shelf (COTS), KNN (K-nearest neighbor).

References

1. Etten, A.V. You Only Look Twice: Rapid Multi-Scale Object Detection in Satellite Imagery. *arXiv* **2018**, arXiv:1805.09512.
2. Samadzadegan, F.; Dadrass Javan, F.; Ashtari Mahini, F.; Gholamshahi, M. Detection and Recognition of Drones Based on a Deep Convolutional Neural Network Using Visible Imagery. *Aerospace* **2022**, *9*, 31. [CrossRef]
3. Yang, Y.; Gong, H.; Wang, X.; Sun, P. Aerial target tracking algorithm based on faster R-CNN combined with frame differencing. *Aerospace* **2017**, *4*, 32. [CrossRef]
4. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
6. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
7. Albahli, S.; Nawaz, M.; Javed, A.; Irtaza, A. An improved faster-RCNN model for handwritten character recognition. *Arab. J. Sci. Eng.* **2021**, *46*, 8509–8523. [CrossRef]
8. 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.
9. 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 European Conference on Computer Vision (ECCV), 11–14 October 2016; Springer; pp. 21–37.
10. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
11. Zheng, W.; Tang, W.; Jiang, L.; Fu, C.W. SE-SSD: Self-ensembling single-stage object detector from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 14494–14503.
12. Zhu, Y.; Du, J.; Wu, X. Adaptive period embedding for representing oriented objects in aerial images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7247–7257. [CrossRef]
13. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Chen, K.; Xia, G.S.; Bai, X. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1452–1459. [CrossRef] [PubMed]
14. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. *arXiv* **2019**, arXiv:1908.05612.
15. Yang, X.; Yang, X.; Yang, J.; Ming, Q.; Wang, W.; Tian, Q.; Yan, J. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; Volume 34.
16. Sun, X.; Wang, P.; Wang, C.; Liu, Y.; Fu, K. PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 50–65. [CrossRef]
17. Ming, Q.; Miao, L.; Zhou, Z.; Dong, Y. CFC-Net: A critical feature capturing network for arbitrary-oriented object detection in remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [CrossRef]
18. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
19. Liu, Z.; Yuan, L.; Weng, L.; Yang, Y. A high resolution optical satellite image dataset for ship recognition and some new baselines. *Int. Conf. Pattern Recognit. Appl. Methods* **2017**, *2*, 324–331.
20. Yang, X.; Yan, J.; He, T. On the arbitrary-oriented object detection: Classification based approaches revisited. *arXiv* **2020**, arXiv:2003.05597.
21. Zhang, R.; Wang, Z.K.; Zhang, Y.L. A person-following nanosatellite for in-cabin astronaut assistance: System design and deep-learning-based astronaut visual tracking implementation. *Acta Astronaut.* **2019**, *162*, 121–134.
22. Wang, Y.; Wei, X.; Tang, X.; Shen, H.; Zhang, H. Adaptive Fusion CNN Features for RGBT Object Tracking. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 7831–7840. [CrossRef]
23. Gurkan, F.; Cerkezi, L.; Cirakman, O.; Gunsel, B. TDIOT: Target-Driven Inference for Deep Video Object Tracking. *IEEE Trans. Image Process.* **2021**, *30*, 7938–7951. [CrossRef] [PubMed]
24. Nandy, A.; Haldar, S.; Banerjee, S.; Mitra, S. A survey on applications of siamese neural networks in computer vision. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–5.
25. Luo, Y.; Shen, H.; Cao, X.; Wang, T.; Feng, Q.; Tan, Z. Conversion of Siamese networks to spiking neural networks for energy-efficient object tracking. *Neural Comput. Appl.* **2022**, *34*, 9967–9982. [CrossRef]
26. Ma, C.; Yang, F.; Li, Y.; Jia, H.; Xie, X.; Gao, W. Deep human-interaction and association by graph-based learning for multiple object tracking in the wild. *Int. J. Comput. Vis.* **2021**, *129*, 1993–2010. [CrossRef]
27. Panetta, K.; Kezebou, L.; Oludare, V.; Agaian, S. Comprehensive underwater object tracking benchmark dataset and underwater image enhancement with GAN. *IEEE J. Ocean. Eng.* **2021**, *47*, 59–75. [CrossRef]

28. Hofhauser, A.; Steger, C.; Navab, N. Edge-based template matching with a harmonic deformation model. In Proceedings of the International Conference on Computer Vision and Computer Graphics, Warsaw, Poland, 10–12 November 2008; Springer: Berlin/Heidelberg, Germany; pp. 176–187.
29. Stojanovic, I.; Taskovski, D.; Kraljevski, I. Normalized Correlation Coefficients for Searching JPEG Images. In Proceedings of the 7th Information Technologies 2002, Zabljak, Yugoslavia, 24 February–2 March 2002.
30. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-first AAAI conference on artificial intelligence, San Francisco, CA, USA, 4–9 February 2017.
31. Zivkovic, Z.; Van Der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit. Lett.* **2006**, *27*, 773–780. [CrossRef]