drones

MDPI

Article

An Efficient Adjacent Frame Fusion Mechanism for Airborne Visual Object Detection

Zecong Ye [1,2], Yueping Peng [1,*], Wenchao Liu [1], Wenji Yin [1], Hexiang Hao [1], Baixuan Han [1], Yanfei Zhu [1] and Dong Xiao [1,3]

[1] School of Information Engineering, Engineering University of PAP, Xi'an 710086, China; yzc6666@yeah.net (Z.Y.); liuwch3@mail3.sysu.edu.cn (W.L.); 20210058@ntit.edu.cn (W.Y.); hhx1214s@163.com (H.H.); hbx911wj@163.com (B.H.); yanfei_912@163.com (Y.Z.); 15859053199@139.com (D.X.)
[2] The Youth Innovation Team of Shaanxi Universities, Xi'an 710086, China
[3] Fujian Armed Police Corps, Fuzhou 350000, China
* Correspondence: percy001@163.com

Abstract: With the continuous advancement of drone technology, drones are demonstrating a trend toward autonomy and clustering. The detection of airborne objects from the perspective of drones is critical for addressing threats posed by aerial targets and ensuring the safety of drones in the flight process. Despite the rapid advancements in general object detection technology in recent years, the task of object detection from the unique perspective of drones remains a formidable challenge. In order to tackle this issue, our research presents a novel and efficient mechanism for adjacent frame fusion to enhance the performance of visual object detection in airborne scenarios. The proposed mechanism primarily consists of two modules: a feature alignment fusion module and a background subtraction module. The feature alignment fusion module aims to fuse features from aligned adjacent frames and key frames based on their similarity weights. The background subtraction module is designed to compute the difference between the foreground features extracted from the key frame and the background features obtained from the adjacent frames. This process enables a more effective enhancement of the target features. Given that this method can significantly enhance performance without a substantial increase in parameters and computational complexity, by effectively leveraging the feature information from adjacent frames, we refer to it as an efficient adjacent frame fusion mechanism. Experiments conducted on two challenging datasets demonstrate that the proposed method achieves superior performance compared to existing algorithms.

Keywords: drone-to-drone detection; airborne vision; spatio-temporal information; feature fusion; deep learning

Citation: Ye, Z.; Peng, Y.; Liu, W.; Yin, W.; Hao, H.; Han, B.; Zhu, Y.; Xiao, D. An Efficient Adjacent Frame Fusion Mechanism for Airborne Visual Object Detection. *Drones* **2024**, *8*, 144. https://doi.org/10.3390/drones8040144

Academic Editor: Pablo Rodríguez-Gonzálvez

Received: 17 February 2024
Revised: 25 March 2024
Accepted: 6 April 2024
Published: 7 April 2024

1. Introduction

Drones have become readily accessible and extensively employed in various fields, including mapping [1], security [2,3], agriculture [4], express delivery [5], and numerous others [6]. In the forthcoming years, the use of autonomous and intelligent drones is expected to rise exponentially. To ensure secure flight and mitigate the potential risks associated with drones, the development of drone-to-drone detection technology assumes paramount significance [7–9]. Notably, this research domain remains largely unexplored, offering ample opportunities for investigation and advancement.

In contrast to conventional object detection methods, drone-to-drone detection encounters numerous challenges. Primarily, within an aerial visual context, drone targets often exhibit diminutive sizes and lack distinct texture features, rendering them susceptible to background interference, thereby impeding their detection. Furthermore, the aerial backdrop in air-to-air scenarios encompasses intricate and dynamic elements. The presence of cloud formations in the atmosphere leads to a heightened frequency of drone targets

materializing or vanishing. In contrast to ground-to-air drone detection, drones in mid-flight possess the capability to capture drone targets from oblique or overhead perspectives, resulting in potentially intricate backgrounds comprising urban landscapes or natural terrains. Additionally, drone targets exhibit varied characteristics as they are capable of rotating along any axis. This rotational capability can lead to significant alterations in the appearance of drones, including changes in their shape, size, and color. Moreover, achieving a harmonious trade-off between accuracy and speed is imperative when detecting drone targets. Compared to ground-level imagery, airborne vision typically offers higher-resolution images. However, the processing of such high-resolution images poses a significant challenge for drone platforms in terms of memory and computational capabilities. Simultaneously, object detection tasks utilizing airborne vision frequently necessitate low-latency processing, thereby intensifying the trade-off between model computation and detection accuracy.

In recent research, several scholars have made enhancements to the general object detection algorithm, specifically tailored for small drone object detection tasks [10,11]. However, as these studies do not incorporate temporal information, their performance falls below optimal levels. Research has demonstrated that the human visual system possesses remarkable sensitivity in detecting object motion. In object detection tasks, humans depend not only on static features but also on temporal variations exhibited by objects [12,13]. Considering this perspective, although the visual properties of small drone targets may be constrained, their motion characteristics can be leveraged to enhance precision. A recent approach known as Tiny Airborne object Detection (TAD) [14] capitalizes on the motion characteristics of drone targets. The authors proposed a framework that diverges from general object detection, as it exclusively employs motion data between consecutive frames to pinpoint the drone target. Unlike prevailing techniques that rely on optical flow estimation or background subtraction for capturing motion cues, their approach demonstrated high efficiency, requiring minimal model parameters and achieving fast processing speed. Initially, the authors constructed a motion pattern model by computing the local similarity of the feature image. Subsequently, the consistency of motion was directly described by calculating the local similarity of motion patterns. Next, a simple network was employed to facilitate the positioning of the object's center. Finally, a separate network branch was utilized to make predictions regarding the coordinates of the bounding box. Nevertheless, this method is accompanied by certain drawbacks. First, the method cannot identify targets whose motion trajectory is perpendicular to the imaging plane of the camera's field of view or hovering targets. Second, this method cannot identify drone targets larger than 32×32 pixels because it can only extract the edge motion features of the target, which can lead to the incorrect positioning of the target. In addition to the aforementioned approach, several researchers [15–17] have proposed utilizing multi-frame information to enhance model performance. However, these methods suffer from issues such as excessive computational steps or a substantial increase in calculations. Meanwhile, in the context of aerial visual scenes, employing an excessive number of video frames to enhance object detection performance, particularly for drone-to-drone detection, holds limited significance. Undoubtedly, the final frame result holds greater importance. Blindly increasing the number of video frames in an attempt to enhance performance will inevitably impede the efficiency of obtaining the final frame result.

To address the aforementioned challenges and effectively leverage temporal information, we propose an efficient mechanism for fusing adjacent frames. This mechanism fully utilizes the motion of target pixels between adjacent frames and the key frame to achieve optimal results. The mechanism we propose can be inserted into general object detection frameworks, similar to the attention mechanism. This allows for the detection of targets with motion trajectories perpendicular to the imaging plane of the camera's field of view or hovering targets and the enhancement of the target characteristics. In addition, the general object detection algorithm can recognize objects with different scales. First, similar to the TAD algorithm [14], we establish pixel correspondence between features of adjacent frames

and key frames by employing local similarity calculation. This approach enables us to model pixel motion, as depicted in Figure 1. Next, we utilize the pixel motion information to acquire data analogous to optical flow, enabling the alignment of adjacent frame features with key frame features. Subsequently, the aligned features and key frame features are fused using similarity weights. Simultaneously, we employ pixel motion information to extract background information from adjacent frame features for the purpose of background subtraction. The main contributions of our work can be summarized as follows:

- We propose an efficient mechanism for adjacent frame fusion, consisting of two modules that entail minimal parameter increment and impose negligible computational overhead. Our mechanism is designed to be plug-and-play, ensuring ease of implementation. It has been validated on two datasets, NPS [18] and FL-Drone [7], demonstrating significantly enhanced effects.
- We propose a feature alignment fusion module, which distinguishes itself from intricate alignment techniques such as optical flow estimation and deformable convolution. Instead, this module utilizes local similarity calculation to align the features of adjacent frames with those of key frames and subsequently use them for feature fusion. Simultaneously, a comprehensive ablation study was conducted to substantiate the effectiveness of the proposed feature alignment fusion module.
- We propose a background subtraction module, drawing inspiration from the background subtraction technique in moving object detection. This module subtracts the background features of the adjacent frames from the foreground features of the key frame to enhance the target features and enhance the model's accuracy.

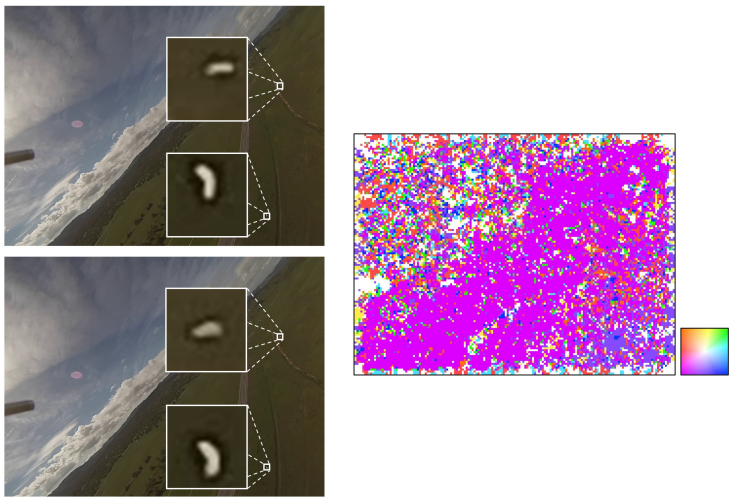

Figure 1. Pixel motion diagram. The previous frame is shown in the upper-left corner, the key frame is shown in the lower-left corner, and the pixel motion diagram of the feature obtained using local similarity calculation is shown on the right. It can be regarded as an optical flow visualization diagram, in which color is used to represent different motion directions, as indicated by the color wheel next to it. Two target motion states can be seen in the image: the upper target moves to the left, and the lower target moves downward, consistent with the corresponding parts of the pixel motion diagram.

2. Related Works

2.1. Small Object Detection

In recent years, various methods have been proposed to address the issue of small object detection. The primary challenge in detecting small objects is the limited representation loss during feature extraction. Existing methods for small object detection have been enhanced by building upon mainstream object detection network models, mainly through data augmentation, multi-scale feature fusion, super-resolution, and increasing the number of detection heads. The data augmentation method involves copying and

pasting the small target or adding extra image data to the dataset using a specific proportional matching strategy. This approach helps enhance the model's robustness to some extent and addresses the issue of unclear visual features and limited target information for the small target [19–21]. The multi-scale feature fusion method aims to enhance the detection accuracy of small targets by leveraging both the low-level high-resolution and high-level strong feature semantic information of the network [22–24]. The super-resolution method reduces the feature difference between small-scale targets and large-scale targets through feature mapping and learning high-resolution feature representation of small targets, thereby improving the detection accuracy of small targets [25–27]. Increasing the number of detection heads involves enhancing the detection capability for small targets, aiming to improve the accuracy of their detection. Examples of algorithms employing this approach include Tph [28] and FasterX [29]. Increasing the number of detection heads can alleviate the negative impact of severe target scale changes. However, since these algorithms do not utilize temporal information, their performance may be suboptimal. Additionally, when existing methods are directly applied to airborne visual scenes, several problems may arise. Data augmentation may only be effective for specific datasets and scenes, while methods involving super-resolution, multi-scale feature fusion, or increasing the number of detection heads can increase the computational burden. Therefore, it is necessary to further balance the relationship between accuracy and speed.

2.2. Video Object Detection

In airborne vision, it is crucial to utilize temporal information from videos to enhance the accuracy of the model. On one hand, data obtained through airborne vision usually consist of video data (i.e., image sequence). On the other hand, when the target cannot be identified in a single frame of a static image, it is necessary to utilize contextual spatio-temporal information from video data to enhance target features. This method of utilizing video information to enhance the model's performance is known as video object detection. Video object detection can be categorized into two groups based on the utilization of temporal information: leveraging the spatio-temporal consistency of target motion and feature aggregation. The spatio-temporal consistency methods of target motion mainly include post-processing methods [30,31] and tracking-based video object detection [32]. The feature aggregation method aggregates frame features at different distances from key frames, including adjacent frame features and long-term frame features, to enhance the features of objects. Examples of this method include FGFA, MEGA, and Transvisdrone. FGFA [33] mainly uses optical flow to align features extracted from adjacent frames with those extracted from the key frame and then fuses these features to enhance detection accuracy. MEGA [34] comprehensively considers global and local feature information and proposes a long-range memory module, allowing the key frame to obtain broader and more complete feature information. Transvisdrone [17] combines YOLOv5 [35] and the Video Swin model [36] to enhance drone detection in challenging scenes by learning the spatio-temporal dependence of drone motion. However, these methods only consider the full utilization of information from adjacent frames or long-term frames, focusing on enhancing model accuracy, without further consideration of how to efficiently leverage this information. They are usually improved based on the two-stage object detection algorithm. Consequently, these methods are time-consuming, and some are not end-to-end, which increases the number of steps in model training.

Starting from the practical application of airborne vision, this paper focuses on efficiently utilizing adjacent frames to enhance the characteristics of small targets, improve the model's performance, avoid significantly increasing computational load, and align more closely with the time-sensitive nature of airborne vision tasks. An efficient plug-and-play adjacent frame fusion mechanism is proposed, which consists of two modules. The first module utilizes the aligned features to fuse and enhance the target features based on their similarity, whereas the second module leverages the background of the adjacent frames to enhance the target features. Unlike the video object detection algorithm mentioned above,

we thoroughly incorporate target motion information and background feature information from adjacent frames. Due to these purposeful designs, the algorithm is better suited for airborne vision scenes and the detection of small targets such as drones.

3. Proposed Method

In this section, we introduce the technical details of our efficient adjacent frame fusion mechanism. Specifically, our method is very simple and mainly includes three key parts, as shown in Figure 2. In the mechanism, the local similarity calculation is first performed to obtain the similarity volume, which is utilized in the subsequent feature alignment fusion module and the background subtraction module. It is worth noting that the local similarity calculation is performed on the features of the key frame and the adjacent frames extracted through the network. Therefore, feature extraction needs to be performed first, which means the efficient adjacent frame fusion mechanism needs to be inserted into the network backbone.

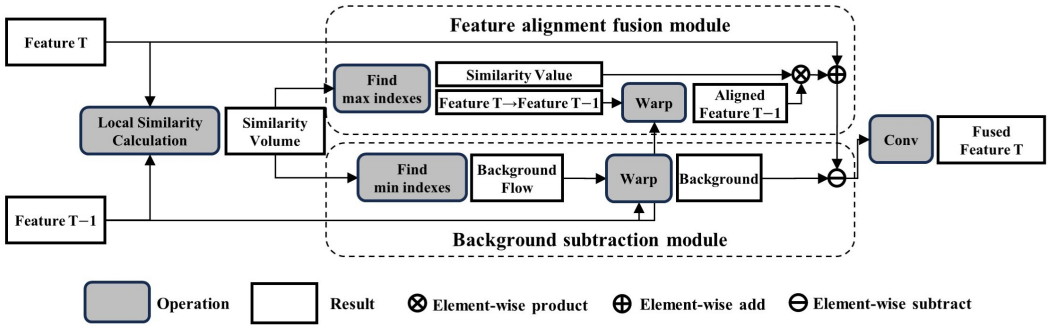

Figure 2. The framework of the efficient adjacent frame fusion mechanism. The mechanism mainly includes three key parts: (1) local similarity calculation, (2) the feature alignment fusion module, and (3) the background subtraction module.

3.1. Local Similarity Calculation

Some researchers [37] have found that in the optical flow field of 100 ImageNet videos calculated by FlowNet, the edge distribution of the optical flow field along the vertical axis and the horizontal axis is mainly concentrated near zero, as shown in Figure 3.

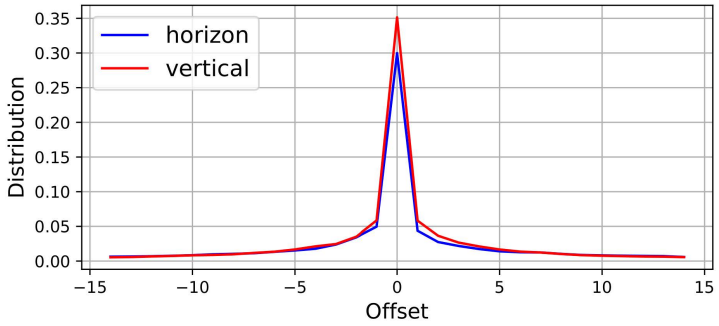

Figure 3. Optical flow field of sampled 100 ImageNet VID videos computed by FlowNet in horizontal and vertical dimensions [37].

Compared with the ImageNet video dataset, the targets in airborne vision are typically smaller. Due to the long distance between the target and the airborne camera, the target does not move significantly in adjacent frames. In addition, the presence of the downsampling layer in the neural network reduces the size of the feature image. Therefore, unlike optical flow estimation [38,39], to obtain the motion trajectory of pixels, we utilize local similarity calculation as a simple and effective way to establish the motion relationship between adjacent frame feature pixels and key frame feature pixels. Local similarity calculation, also

known as cost volume, is widely used in optical flow estimation [38,40]. In the TAD [14] algorithm, this method is used to obtain pixel motion modeling and motion consistency modeling. The method calculates the cosine similarity between the feature vector of each position in the feature map and its surrounding feature vector. The process is formulated as follows:

$$Sim(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

$$S^k_{ijmn} = Sim(f^T_{ij}, f^{T-1}_{(i-\lfloor k/2 \rfloor+m)(j-\lfloor k/2 \rfloor+n)}), 0 \leq m, n < k. \tag{2}$$

where $Sim(A, B)$ is the cosine similarity between A and B. $f^T \in \mathbb{R}^{H \times W \times D}$ represents the features from the key frame. f^T_{ij} represents the feature vector of position (i, j) in f^T. The meaning of $f^{T-1}_{(i-\lfloor k/2 \rfloor+m)(j-\lfloor k/2 \rfloor+n)}$ is similar to f^T_{ij}, representing the feature vector from the previous frame. S^k_{ijmn} represents the cosine similarity between the feature vector of position (i, j) in f^T and the k neighborhood feature vector in f^{T-1}, and $S \in \mathbb{R}^{H \times W \times k \times k}$ represents the similarity volume. For ease of understanding, the calculation is shown in Figure 4.

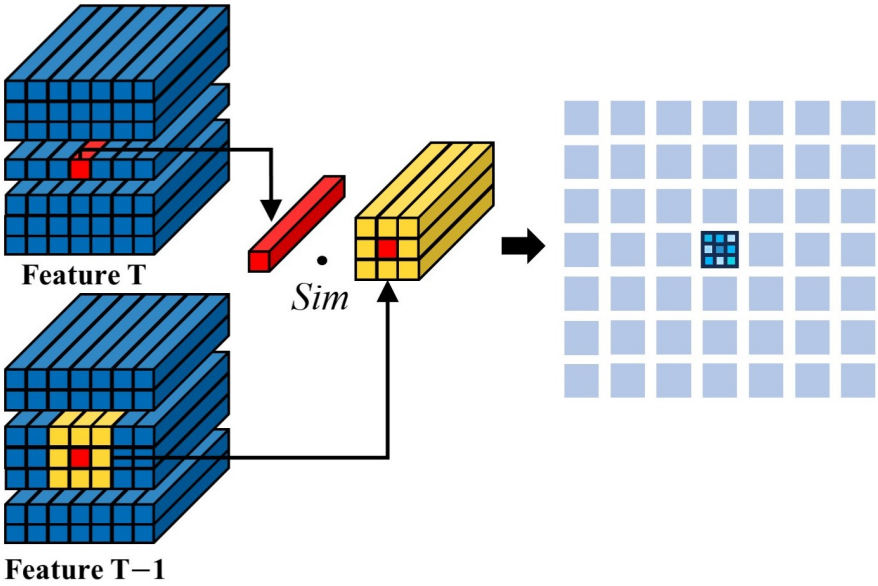

Figure 4. Local similarity calculation.

3.2. Feature Alignment Fusion Module

The feature alignment fusion module uses the aligned adjacent frame image features for fusion. Existing feature alignment methods [33,41–43] often use Deformable Convolutional Networks (DCNs) [44] or optical flow estimation methods [38] to align adjacent frame features with key frame features, which greatly increases the amount of computation or the number of parameters. However, in airborne visual scenes, the motion relationship information between the adjacent frame features and the key frame features can be used to align the features. In this approach, the similarity volume S is used for feature alignment, requiring only a minimal increase in computation without introducing additional parameters.

The alignment process is shown in Figure 5. First, the most similar local feature relative positions are found in a similar volume. The deeper the color, the more similar they are. It can be seen that the red position is the most similar relative position between the adjacent frame features and the key frame features. For example, the $(-1, -1)$ shown in the first rectangle means that the position of the key frame feature (i, j) is most similar to the position of the adjacent frame feature $(i-1, j-1)$. These relative positions can then be combined to form something similar to optical flow, which is the trajectory of all feature positions, called Feature T $\rightarrow$ Feature T$-$1. Then, use Feature T $\rightarrow$ Feature T$-$1 to warp f^{T-1} to obtain the aligned adjacent frame features. The specific operation process of

warping can be seen on the right side of the figure. For the convenience of the display, the feature vectors in each row are marked with different colors, and the warping process of the middle nine feature vectors is illustrated here. The red arrow indicates the displacement from Feature T $\rightarrow$ Feature T$-$1, aligning the adjacent frame features with the key frame features through this warping operation. Then, we use weighted addition to fuse the features. The weight is obtained from the result of the previous local similarity operation, that is, the similarity value at the red position of Feature T $\rightarrow$ Feature T$-$1. The process is formulated as follows:

$$F_{T\rightarrow T-1}, S_{\max} = f_{\max}(S) \tag{3}$$

$$\bar{f}^{T-1} = W(f^{T-1}, F_{T\rightarrow T-1}) \tag{4}$$

$$\hat{f}^{T} = f^{T} + S_{\max} \cdot \bar{f}^{T-1} \tag{5}$$

where $f_{\max}(\cdot)$ represents the Find max indexes step in Figure 2. This operation is performed on S to obtain $F_{T\rightarrow T-1} \in \mathbb{R}^{H\times W\times 2}$, similar to an optical flow graph, denoted as Feature T $\rightarrow$ Feature T$-$1. It obtains the maximum value corresponding to each feature position in the similarity value. $W(\cdot,\cdot)$ represents the process of aligning the adjacent frame features. $\bar{f}^{T-1} \in \mathbb{R}^{H\times W\times D}$ refers to Aligned Feature T$-$1 in Figure 2. Finally, we obtain $\hat{f}^{T} \in \mathbb{R}^{H\times W\times D}$ through weighted fusion.

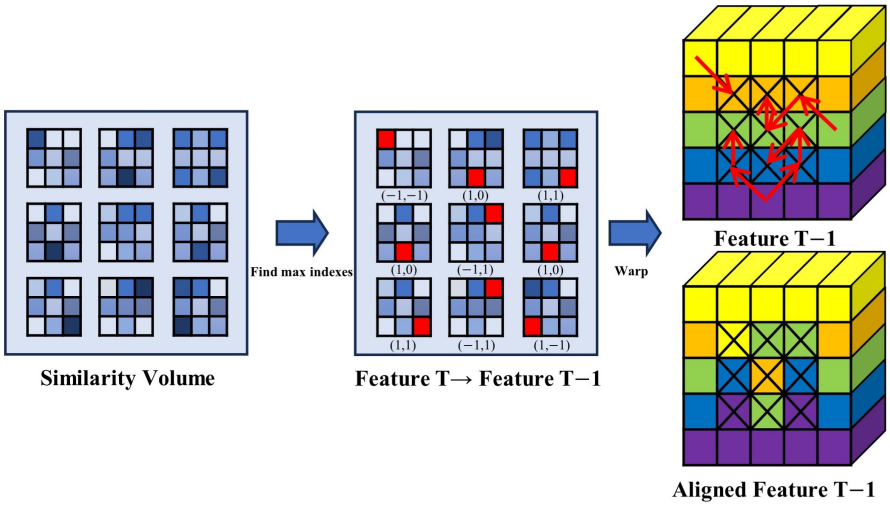

Figure 5. The process of alignment.

3.3. Background Subtraction Module

Inspired by the background subtraction method [45] in moving object detection, we use a similar method to eliminate background information and highlight the foreground. The core idea of background subtraction is to subtract the determined or real-time updated background model from the current frame in the image sequence. Although the principle of background subtraction is simple, this method is mainly applied to fixed camera scenes. In airborne vision, background changes that are too fast or too complex can easily lead to poor results. How to obtain the background information of the target is very important.

The background subtraction module refers to the subtraction between the foreground information of the key frame and the background information of the adjacent frame to further strengthen the target feature. In this module, we use a method similar to that in the feature alignment fusion module to obtain the target background information. Unlike the feature alignment fusion module, where the goal is to find the most similar position to the key frame, this module aims to find the most dissimilar position to the key frame, which can be regarded as the background information of the target. For the position containing the target semantics, the most dissimilar position represents the background, whereas for the background, the most dissimilar position may contain the target semantics. We adopt the idea of background subtraction to subtract the background information of adjacent

frames from the key frame features, thereby separating the background information from the target semantic information to further enhance features. Ablation experiments were carried out to verify the feasibility of the method. The process is formulated as follows:

$$F_B = f_{\min}(S) \tag{6}$$

$$f^B = W(f^{T-1}, F_B) \tag{7}$$

$$f^F = Conv_{1\times1}(\hat{f}^T - f^B) \tag{8}$$

where $f_{\min}(\cdot)$ denotes the Find min indexes step in Figure 2, which finds the relative position of the adjacent frame feature vector that is most dissimilar to the key frame feature vector. $F_B \in \mathbb{R}^{H\times W\times 2}$ can be regarded as the optical flow for acquiring the background position, known as the background flow. $f^B \in \mathbb{R}^{H\times W\times D}$ represents the background feature obtained after using the adjacent frame warp, referred to as Background in Figure 2. Then, f^B is subtracted from $\hat{f}^T$ obtained by the feature alignment fusion module.

Finally, the result from subtraction is input into a 1×1 convolution to obtain $f^F \in \mathbb{R}^{H\times W\times D}$ fused with the adjacent frame. Throughout the process, it is evident that in the proposed mechanism, only the last 1×1 convolution increases the number of parameters. This is a negligible increase compared to the network parameters.

4. Experiments

4.1. Experimental Setup

Datasets. We used the FL [7] and NPS [18] datasets. The FL dataset contains 14 videos with a total of 38,948 frames of grayscale images. These images were captured by airborne vision and include indoor and outdoor scenes. The resolution is 640×480 or 752×480, and the target size ranges from 9×9 to 259×197, with an average size of 25.5×16.4. The NPS dataset contains 50 videos with a total of 70,250 frames of color images. These images were also captured by airborne vision. The resolution is 1920×1280 or 1280×760, and the target size ranges from 10×8 to 65×21, with an average size of 16.2×11.6. We used the clean version annotations released by Ashraf [15], and the datasets were divided as described in [17]. For the FL dataset, we used half of the frames in each video as the training set and the other half as the test set. For the NPS dataset, we allocated videos with video IDs #01–#36 as the training set, videos with video IDs #37–#40 as the validation set, and videos with video IDs #41–#50 as the test set.

Implementation details and metrics. In order to facilitate comparison with existing SOTA methods, we used YOLOv5l [35] as the basic framework, consistent with the latest video-object detection [17] and small object detection [28] methods. The training and testing configurations followed those outlined in [17]. Training only used frames with provided annotations. For the evaluation, we evaluated every fourth frame. The experiments were performed on an Intel Xeon W-2245 CPU and NVIDIA RTX 3090 24G GPU. In the experiments, we inserted the proposed efficient adjacent frame fusion module into the P3 layer of YOLOv5, that is, after the second C3 module. The evaluation metrics included average precision at IoU = 0.5 (AP), precision (P), recall (R), F1-score (F1), parameters (Param.), Giga Floating-point Operations Per Second (GFLOPs), and frames per second (FPS) to highlight the efficiency of the proposed mechanism.

4.2. Comparison with State of the Art

We compared our method with state-of-the-art methods on the FL and NPS datasets. All compared methods were implemented using their official code or MMdetection. As shown in Table 1, performance-wise, our method outperformed the other methods by 0.6% in terms of the AP metric on the FL dataset. Additionally, when using only two frames of information, our method achieved the same AP value as the Transvisdrone method. The reason why the AP metric achieved by our method was not as good as that of Transvisdrone (f = 5) on the NPS dataset is that the algorithms use different amounts of information. Transvisdrone (f = 5) uses information from four adjacent frames, whereas our method

only uses information from one adjacent frame. Moreover, the FL and NPS datasets are significantly different. Table 1 shows that the FL dataset is more challenging to learn compared to the NPS dataset, which suggests that our method may perform better on challenging datasets. In terms of throughput, which was measured in frames per second on the RTX3090 24G, the proposed method outperformed the state-of-the-art methods, demonstrating the great advantage of our algorithm. Specifically, our approach was 1.5 times faster than the other methods. In summary, it can be concluded that the advantages of our method are evident in terms of both accuracy and speed.

Table 1. Comparison of different methods.

Method	AP-FL	AP-NPS	FPS
FCOS [46]	62.4	83.7	28
Mask-RCNN [47]	68.9	89.5	29
YOLOv5-tph [28]	67.2	92.5	27
Dogfight [15]	72.0	89.1	2
Transvisdrone (f = 2) [17]	71.7	94.0	30
Transvisdrone (f = 5) [17]	72.6	**94.9**	30
Ours	**73.2**	94.0	**45**

The optimal results for each metric are shown in **bold** in the table. All tables below do the same.

4.3. Ablation Experiments and Analysis

In order to verify the efficiency of the proposed module, we conducted ablation experiments on the FL dataset.

Ablation study of frame resolution. Using different resolution inputs allows for a trade-off between performance and throughput. We used four different resolutions (1280, 800, 640, and 480) to compare the performance of the proposed method. The baseline (YOLOv5l) was also used to compare the advantages of the proposed method. Table 2 shows that our method achieved a 4.7% higher AP compared to the baseline. However, the number of parameters only increased by 0.07 M, the computation only increased by 3.3 GFLOPs, and the speed was slower than the baseline by 6 FPS. These results indicate that the computational load introduced by our mechanism was acceptable. When comparing the first and third rows in the table, it can be observed that our method achieved the same AP value as the baseline but with a 75% reduction in computation and more than twice the speed of the baseline. Similarly, when comparing the fourth and fifth rows, it can be seen that the FPS was twice as fast even with a reduction of only 0.3% in the AP.

Table 2. Ablation study of frame resolution.

Method	Resolution	AP	P	R	F1	Param. (M)	GFLOPs	FPS
Baseline	1280	68.5	73.5	66.6	69.9	**46.10**	430.6	51
Ours	480	67.8	66.5	68.3	67.4	46.17	**61.0**	**133**
	640	68.5	69.5	67.6	68.5	46.17	108.5	113
	800	72.9	73.4	71.0	72.2	46.17	169.5	90
	1280	**73.2**	**73.5**	**72.3**	**72.9**	46.17	433.9	45

Effect of different modules. We explored the influence of different modules on the 480-resolution experiment, as shown in Table 3. It can be seen that the weighted addition of values calculated through local similarity yielded better results compared to direct addition. Additionally, it can be seen that both the feature alignment fusion module and the background subtraction module improved the performance of the model with only a small increase in computation. Therefore, it can be concluded that it is feasible to use local similarity calculation to align features and extract the background.

Table 3. Ablation study of different modules. Align and weighted add, subtract, and both refer to the feature alignment fusion module, background subtraction module, and efficient adjacent frame fusion mechanism, respectively.

Method	AP	P	R	F1	Param. (M)	GFLOPs	FPS
Baseline	59.5	64.9	62.0	63.4	**46.10**	**60.5**	**147**
Align and directly add	63.0	65.6	61.7	63.6	46.17	61.0	139
Align and weighted add	65.0	65.3	64.8	65.0	46.17	61.0	137
Subtraction	61.7	64.0	57.1	60.3	46.17	61.0	138
Both	**67.8**	**66.5**	**68.3**	**67.4**	46.17	61.0	133

Effect of insertion position. In the previous experiment, we inserted the proposed mechanism into the P3 layer of YOLOv5l, that is, after the second C3 module. We also compared the effects of different layers within the network, as shown in Table 4. P4 refers to inserting the mechanism after the third C3 module, and P5 refers to inserting it after the fourth C3 module. It can be seen that the insertion of the mechanism at P3 worked the best.

Table 4. Results of the mechanism at different insertion positions.

Method	AP	P	R	F1	Param. (M)	GFLOPs	FPS
P3	**67.8**	66.5	**68.3**	**67.4**	**46.17**	61.0	**133**
P4	58.9	**66.7**	58.6	62.4	46.37	61.0	132
P5	61.4	67.4	63.9	65.6	46.37	61.0	131

Effect of neighborhood size. We investigated the influence of different neighborhood sizes K on the generation of the similarity volume, as shown in Table 5. The experimental results indicate that a larger K does not necessarily lead to better performance. This also verifies two observations: Firstly, in airborne vision, targets do not move significantly in adjacent frames. Secondly, the existence of downsampling layers in the neural network reduces the size of the image, resulting in small movement of targets in the feature map.

Table 5. Results of the mechanism with different neighborhood sizes.

Method	AP	P	R	F1	Param. (M)	GFLOPs	FPS
K = 3	67.8	66.5	68.3	67.4	46.17	61.0	133
K = 5	66.4	64.7	69.8	67.2	46.17	61.0	124

4.4. Visualization

We evaluated the changes in the feature heatmaps observed when using the proposed mechanism in YOLOv5l, as shown in Figure 6. The first and second rows show the original images with feature heatmaps before and after applying the mechanism, respectively. In order to better show the effect of the mechanism, we show feature heatmaps in the third and fourth rows, corresponding to before and after applying the mechanism, respectively. The last row shows the detection results, where the red box indicates the ground truth and the green box indicates the prediction box. Observing the images, it can be seen that the activation value of the target center feature increases after applying the mechanism, and the target features are enhanced. The comparison between the images in the third and fourth rows in the fourth column shows that the left target features are clearly enhanced.

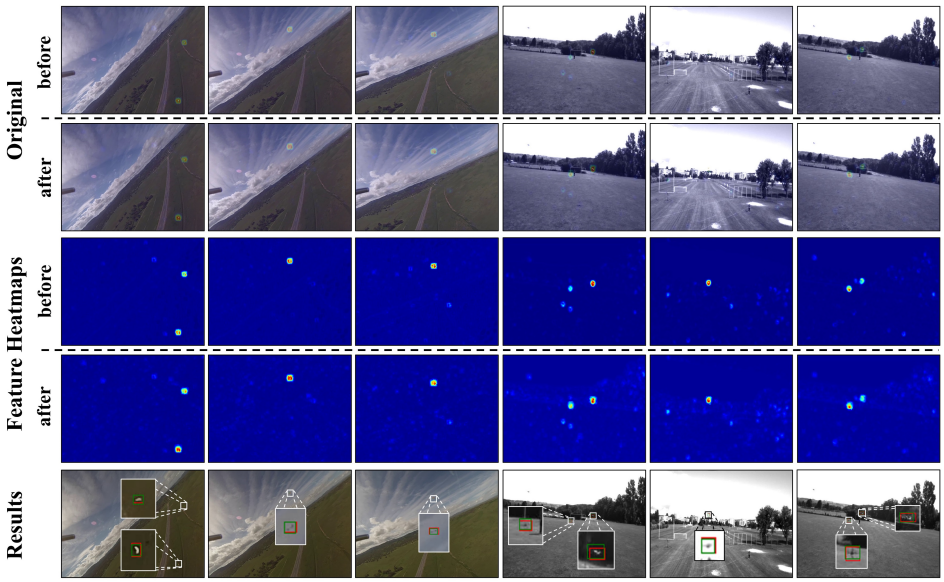

Figure 6. Visualization of feature heatmaps before and after applying the mechanism. The first to the third columns display NPS dataset images, whereas the fourth to sixth columns display FL dataset images (best viewed at 300% zoom).

5. Conclusions

In this paper, we propose an efficient adjacent frame fusion mechanism for airborne visual scenes. The experimental results show that the proposed mechanism can significantly improve the performance of the model by using only the features of an adjacent frame. Compared to state-of-the-art methods, our proposed mechanism is faster and better with fewer parameters. From this, we can conclude that it is not necessary to use a mechanism with high computational complexity to obtain pixel motion in airborne visual scenes, as local similarity calculation can achieve good results. Therefore, our mechanism, along with the TAD algorithm, is effective. Additionally, using only one adjacent frame to improve the detection performance of the key frame is more suitable for time-sensitive scenes in drone-to-drone detection.

At the same time, this perspective also provides alternative approaches to airborne visual object detection. To ensure accuracy and accelerate model operation, a small model combining adjacent and key frames can be used instead of a large model focusing solely on a single frame. Another option is to use low-resolution multi-frame images as inputs to reduce computation while maintaining accuracy.

In subsequent research, we will conduct more experiments to explore the effect of the mechanism at different levels in the network. Additionally, we will explore more efficient fusion design when using multi-frame adjacent frames and how to mitigate any reduction in the speed of detecting results for the last frame.

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

Funding: This work was supported by the Comprehensive Research Project on Equipment [grant no. WJ20211A030131]; the Independent Propositional Project of PAP [grant no. ZZKY20223105]; the Basic Frontier Innovation Project at the Engineering University of PAP [grant no. WJY202209]; the Applied Research Advancement Project in Engineering University of PAP [grant no. WYY202304]; and the Graduate Student Sponsored Project [grant no. JYWJ2023B003].

Data Availability Statement: Data will be made available on request.

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

References

1. Quamar, M.M.; Al-Ramadan, B.; Khan, K.; Shafiullah, M.; El Ferik, S. Advancements and Applications of Drone-Integrated Geographic Information System Technology & mdash: A Review. *Remote Sens.* **2023**, *15*, 5039. [CrossRef]
2. Yin, W.; Peng, Y.; Ye, Z.; Liu, W. A Novel Dual Mixing Attention Network for UAV-Based Vehicle Re-Identification. *Appl. Sci.* **2023**, *13*, 11651. [CrossRef]
3. AL-Dosari, K.; Hunaiti, Z.; Balachandran, W. Systematic Review on Civilian Drones in Safety and Security Applications. *Drones* **2023**, *7*, 210. [CrossRef]
4. Ahirwar, S.; Swarnkar, R.; Srinivas, S.; Namwade, G. Application of Drone in Agriculture. *Int. J. Curr. Microbiol. Appl. Sci.* **2019**, *8*, 2500–2505. [CrossRef]
5. Raivi, A.M.; Huda, S.M.A.; Alam, M.M.; Moh, S. Drone Routing for Drone-Based Delivery Systems: A Review of Trajectory Planning, Charging, and Security. *Sensors* **2023**, *23*, 1463. [CrossRef] [PubMed]
6. Hassanalian, M.; Abdelkefi, A. Classifications, applications, and design challenges of drones: A review. *Prog. Aerosp. Sci.* **2017**, *91*, 99–131. [CrossRef]
7. Rozantsev, A.; Lepetit, V.; Fua, P. Detecting Flying Objects Using a Single Moving Camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 879–892. [CrossRef] [PubMed]
8. Jacobsen, R.H.; Marandi, A. Security Threats Analysis of the Unmanned Aerial Vehicle System. In Proceedings of the MILCOM 2021—2021 IEEE Military Communications Conference (MILCOM), San Diego, CA, USA, 29 November–2 December 2021; pp. 316–322. [CrossRef]
9. Hassija, V.; Chamola, V.; Agrawal, A.; Goyal, A.; Luong, N.C.; Niyato, D.; Yu, F.R.; Guizani, M. Fast, Reliable, and Secure Drone Communication: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* **2021**, *23*, 2802–2832. [CrossRef]
10. Liu, B.; Luo, H. An Improved Yolov5 for Multi-Rotor UAV Detection. *Electronics* **2022**, *11*, 2330. [CrossRef]
11. Liu, H.; Fan, K.; Ouyang, Q.; Li, N. Real-Time Small Drones Detection Based on Pruned YOLOv4. *Sensors* **2021**, *21*, 3374. [CrossRef]
12. Kerzel, D. Eye movements and visible persistence explain the mislocalization of the final position of a moving target. *Vis. Res.* **2000**, *40*, 3703–3715. [CrossRef] [PubMed]
13. Nijhawan, R. Visual prediction: Psychophysics and neurophysiology of compensation for time delays. *Behav. Brain Sci.* **2008**, *31*, 179–198. [CrossRef]
14. Lyu, Y.; Liu, Z.; Li, H.; Guo, D.; Fu, Y. A Real-Time and Lightweight Method for Tiny Airborne Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Vancouver, BC, Canada, 17–24 June 2023; pp. 3016–3025.
15. Ashraf, M.W.; Sultani, W.; Shah, M. Dogfight: Detecting Drones From Drones Videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7067–7076.
16. Sun, Y.; Zhi, X.; Han, H.; Jiang, S.; Shi, T.; Gong, J.; Zhang, W. Enhancing UAV Detection in Surveillance Camera Videos through Spatiotemporal Information and Optical Flow. *Sensors* **2023**, *23*, 6037. [CrossRef]
17. Sangam, T.; Dave, I.R.; Sultani, W.; Shah, M. TransVisDrone: Spatio-Temporal Transformer for Vision-based Drone-to-Drone Detection in Aerial Videos. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 6006–6013. [CrossRef]
18. Li, J.; Ye, D.H.; Chung, T.; Kolsch, M.; Wachs, J.; Bouman, C. Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs). In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 4992–4997. [CrossRef]
19. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for small object detection. *arXiv* **2019**, arXiv:1902.07296.
20. Chen, C.; Zhang, Y.; Lv, Q.; Wei, S.; Wang, X.; Sun, X.; Dong, J. RRNet: A Hybrid Detector for Object Detection in Drone-Captured Images. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Seoul, Republic of Korea, 27 October–2 November 2019.
21. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.Y.; Shlens, J.; Le, Q.V. Learning Data Augmentation Strategies for Object Detection. In Proceedings of the Computer Vision–ECCV 2020, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 566–583.
22. Li, Z.; Zhou, F. FSSD: Feature Fusion Single Shot Multibox Detector. *arXiv* **2017**, arXiv:1712.00960.
23. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659.
24. Qi, G.; Zhang, Y.; Wang, K.; Mazur, N.; Liu, Y.; Malaviya, D. Small Object Detection Method Based on Adaptive Spatial Parallel Convolution and Fast Multi-Scale Fusion. *Remote Sens.* **2022**, *14*, 420. [CrossRef]
25. Wang, H.; Wang, J.; Bai, K.; Sun, Y. Centered Multi-Task Generative Adversarial Network for Small Object Detection. *Sensors* **2021**, *21*, 5194. [CrossRef] [PubMed]
26. Courtrai, L.; Pham, M.T.; Lefèvre, S. Small Object Detection in Remote Sensing Images Based on Super-Resolution with Auxiliary Generative Adversarial Networks. *Remote Sens.* **2020**, *12*, 3152. [CrossRef]

27. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B. SOD-MTGAN: Small Object Detection via Multi-Task Generative Adversarial Network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
28. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Montreal, QC, Canada, 10–27 October 2021; pp. 2778–2788.
29. Zhou, W.; Min, X.; Hu, R.; Long, Y.; Luo, H.; Yi, J. FasterX: Real-Time Object Detection Based on Edge GPUs for UAV Applications. *arXiv* **2022**, arXiv:2209.03157.
30. Han, W.; Khorrami, P.; Paine, T.L.; Ramachandran, P.; Babaeizadeh, M.; Shi, H.; Li, J.; Yan, S.; Huang, T.S. Seq-NMS for Video Object Detection. *arXiv* **2016**, arXiv:1602.08465.
31. Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; et al. T-CNN: Tubelets with Convolutional Neural Networks for Object Detection From Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 2896–2907. [CrossRef]
32. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Detect to Track and Track to Detect. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
33. Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; Wei, Y. Flow-Guided Feature Aggregation for Video Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
34. Chen, Y.; Cao, Y.; Hu, H.; Wang, L. Memory Enhanced Global-Local Aggregation for Video Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
35. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012; Kwon, Y.; Tao, X.; Fang, J.; imyhxy; Michael, K.; et al. ultralytics/yolov5: v6.1-TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. Available online: https://zenodo.org/records/6222936 (accessed on 1 December 2023).
36. Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; Hu, H. Video Swin Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 3202–3211.
37. Guo, C.; Fan, B.; Gu, J.; Zhang, Q.; Xiang, S.; Prinet, V.; Pan, C. Progressive Sparse Local Attention for Video Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
38. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; van der Smagt, P.; Cremers, D.; Brox, T. FlowNet: Learning Optical Flow with Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
39. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
40. Teed, Z.; Deng, J. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In Proceedings of the Computer Vision–ECCV 2020, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 402–419.
41. Xiao, J.; Wu, Y.; Chen, Y.; Wang, S.; Wang, Z.; Ma, J. LSTFE-Net: Long Short-Term Feature Enhancement Network for Video Small Object Detection. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 14613–14622. [CrossRef]
42. Wang, H.; Su, D.; Liu, C.; Jin, L.; Sun, X.; Peng, X. Deformable Non-Local Network for Video Super-Resolution. *IEEE Access* **2019**, *7*, 177734–177744. [CrossRef]
43. Chan, K.C.; Zhou, S.; Xu, X.; Loy, C.C. BasicVSR++: Improving Video Super-Resolution with Enhanced Propagation and Alignment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5972–5981.
44. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
45. Piccardi, M. Background subtraction techniques: A review. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), Hague, The Netherlands, 10–13 October 2004; Volume 4, pp. 3099–3104. [CrossRef]
46. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: A simple and strong anchor-free object detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1922–1933. [CrossRef] [PubMed]
47. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.