electronics

MDPI

Article

A Tracking-Based Two-Stage Framework for Spatio-Temporal Action Detection

Jing Luo [1,†], Yulin Yang [1,2,†], Rongkai Liu [1], Li Chen [1], Hongxiao Fei [1], Chao Hu [3,4,*], Ronghua Shi [3] and You Zou [5]

[1] School of Computer, Central South University, Changsha 410000, China; 194702040@csu.edu.cn (L.C.); hxfei@csu.edu.cn (H.F.)
[2] Hunan Hanma Technology Co., Ltd., Changsha 410083, China
[3] School of Electronic Information, Central South University, Changsha 410000, China; shirh@csu.edu.cn
[4] Hunan "the 14th Five-Year Plan" Research Base of Education Sciences (Research on Educational Informatization), Central South University, Changsha 410083, China
[5] Information and Networking Center, Central South University, Changsha 410083, China
* Correspondence: huchao@csu.edu.cn
† These authors contributed equally to this work.

Abstract: Spatio-temporal action detection (STAD) is a task receiving widespread attention and has numerous application scenarios, such as video surveillance and smart education. Current studies follow a localization-based two-stage detection paradigm, which exploits a person detector for action localization and a feature processing model with a classifier for action classification. However, many issues occur due to the imbalance between task settings and model complexity in STAD. Firstly, the model complexity of heavy offline person detectors adds to the inference overhead. Secondly, the frame-level actor proposals are incompatible with the video-level feature aggregation and Region-of-Interest feature pooling in action classification, which limits the detection performance under diverse action motions and results in low detection accuracy. In this paper, we propose a tracking-based two-stage spatio-temporal action detection framework called TrAD. The key idea of TrAD is to build video-level consistency and reduce model complexity in our STAD framework by generating action track proposals among multiple video frames instead of actor proposals in a single frame. In particular, we utilize tailored tracking to simulate the behavior of human cognitive actions and used the captured motion trajectories as video-level proposals. We then integrate a proposal scaling method and a feature aggregation module into action classification to enhance feature pooling for detected tracks. Evaluations in the AVA dataset demonstrate that TrAD achieves SOTA performance with 29.7 mAP, while also facilitating a 58% reduction in overall computation compared to SlowFast.

Keywords: artificial intelligence; computer vision; action detection; object tracking

Citation: Luo, J.; Yang, Y.; Liu, R.; Chen, L.; Fei, H.; Hu, C.; Shi, R.; Zou, Y. A Tracking-Based Two-Stage Framework for Spatio-Temporal Action Detection. *Electronics* **2024**, *13*, 479. https://doi.org/10.3390/electronics13030479

Academic Editor: Domenico Ursino

Received: 7 December 2023
Revised: 13 January 2024
Accepted: 17 January 2024
Published: 23 January 2024

1. Introduction

Nowadays, Artificial Intelligence (AI) plays an important role in almost every aspect of life. Various AI technologies are used in video, audio, and text data mining applications. In particular, video understanding, as a crucial branch of AI technology, is meeting a growing demand in various fields such as autonomous driving, security surveillance, and transportation. Video understanding has become a popular research area due to its increasing significance in these domains. Spatio-temporal action detection (STAD) is essential in video understanding as it detects multiple person actions spatially and temporally. Sets of bounding boxes are arranged along the temporal sequence to form the detection result, named "action tube" [1]. It can be applied to various fields such as sports analysis [2], video surveillance [3], smart education [4], etc. Current methods for spatio-temporal action detection mainly follow a two-stage consecutive framework. The framework involves action localization and action classification, mostly applying Fast R-CNN [5] architecture. As illustrated in Figure 1a, the localization-based STAD paradigm

takes video frames as input and outputs action tubes. The first part is action localization, which uses an offline person detector to locate actors in the keyframe. The second is action classification, where a model extracts feature from video frames and classifies actions based on actor proposals and extracted features. The actor proposals with predicted action classes are finally linked to constructing an action tube.

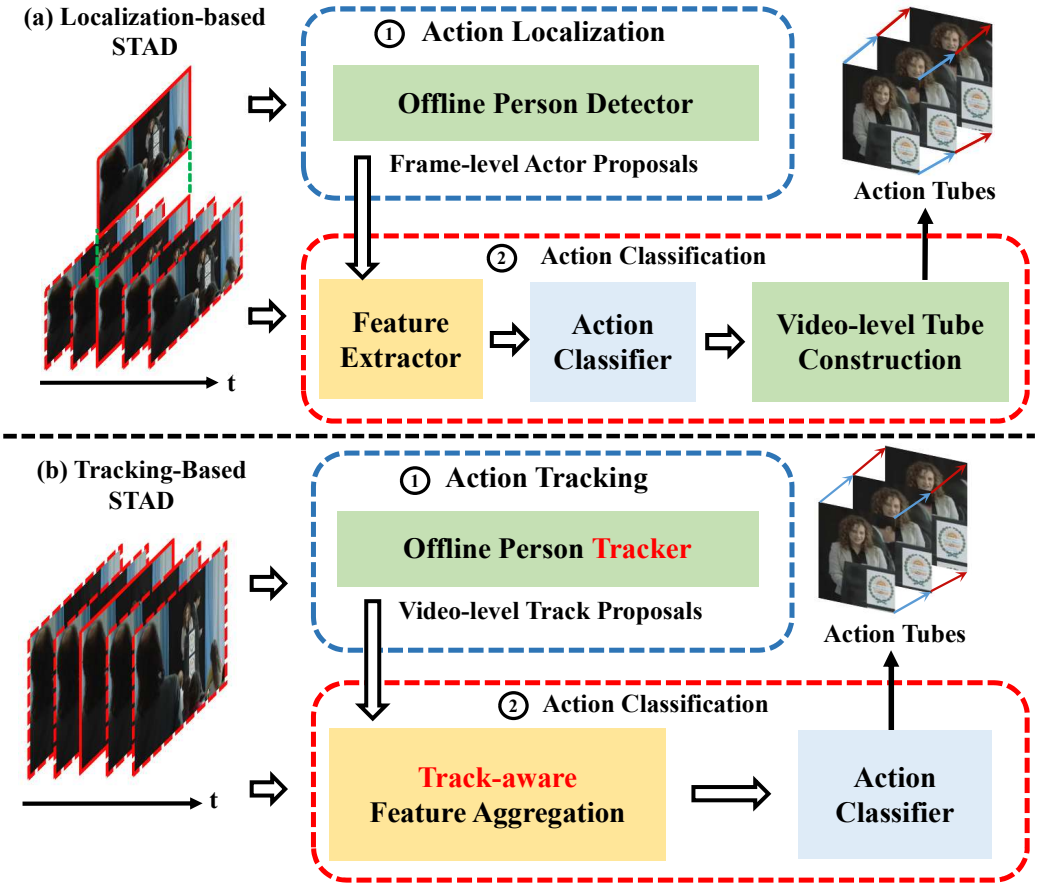

Figure 1. Motivation of TrAD. (**a**) In the localization-based STAD, action localization employs a heavy offline person detector to provide frame-level actor proposals, while action classification is conducted at video level. (**b**) Our proposed tracking-based STAD replaces the heavy detector with an efficient tracker to give video-level track proposals and design a track-aware feature aggregation module to support action classification. We use three colors to indicate components that have the same function in both paradigms.

However, when employing the localization-based paradigm in STAD, the mismatch between task settings and the complexity of the model leads to various issues, hindering its performance.

Issue 1: the offline person detectors increase the computational burden of the detection process. The model complexity of the employed person detectors, such as Faster R-CNN [6] and Deformable DETR [7], is enormous, thus leading to significant inference overhead. For example, SlowOnly [8] employs an offline person detector that has approximately nine times more computing cost than the classifier network (from 406.5 to 41.8 GFLOPs).

Issue 2: the frame-level actor proposals cause incompatibilities between action localization and action classification on feature processing. The actor proposals are inferred by a person detector on single frames but lack spatial and temporal consistency. The same actor may not remain in the same position or may appear only in a few frames. Most models for classification aggregate and pool features based on the RoIs built by merely duplicating proposals around keyframes [9], which cannot fully capture motion or

contextual information. Consequently, the detection performance is limited under diverse action motions and the detection accuracy is reduced.

Drawing inspiration from the intuition that humans focus on an individual in action and then distinguish the action, we propose a similar STAD framework. We name our framework *TrAD*, which is illustrated in Figure 1b. At the heart of TrAD is a principle that not only emphasizes video-level consistency and computational efficiency but also captures the essence of human action perception in an intuitive manner. The framework consists of action tracking and action classification. We extended action localization to action tracking, achieved through a meticulously designed tracking algorithm combined with an efficient detector. The action tracking generates video-level action track proposals spanning multiple video frames, rather than just actor proposals within a solitary keyframe. Furthermore, by employing action tracking, we restructured the tube construction part within the action classification of the localization-based paradigm. This restructure ensures more coherent and efficient task settings within the two-stage framework. Subsequently, the action classification is fine-tuned to better resonate with the tracking-based STAD. We introduce a track-aware feature aggregation module that incorporates a proposal scaling method, a Track-of-Interest Align technique, and a feature aggregation module. This module accentuates feature processing for detected tracks, leading to an enhanced STAD performance.

The contributions of our work are as follows:

- We propose a tracking-based two-stage spatio-temporal action detection framework aimed at achieving better action detection.
- We extend action localization to action tracking by utilizing a simple and robust tracker and propose a track-aware feature aggregation module with an algorithm.
- Experiments on AVA demonstrate that the performance of TrAD excels compared to localization-based state of the art while reducing overall computation significantly.

2. Related Work

We introduce works related to our research, including spatio-temporal action detection, object detection, and multi-object tracking.

2.1. Spatio-Temporal Action Detection

Differing from action recognition [10–12], spatio-temporal action detection provides spatial and temporal localization of multiple action instances in the input video clip. Recently, STAD is drawing more attention. Many related works are proposed, and large datasets such as UCF101-24 [13], AVA [14], and JHMDB [15] are introduced. Most existing works conduct STAD in a two-stage manner while a few works design a unified one-stage STAD model.

Two-stage STAD methods follow the localization-based paradigm of combining action localization and action classification in the Fast R-CNN [5] architecture. The action localization employs a pretrained object detector to locate actors. The next step receives the actor proposals and performs feature processing to further classify the action. However, localization-based methods are trapped in a dilemma of poor detection performance due to the imbalance between task settings and model complexity. Many modules are proposed to perform better STAD such as relation networks [16–19] and feature bank modules [16,20]. We do not consider these as our idea focuses on improving the localization-based paradigm while reducing overall model complexity.

One-stage STAD methods perform action localization and classification in a unified model. YOWO [21] builds one-stage architecture through a unified pipeline with two backbones. Taking a step further, WOO [22] proposes a one-stage end-to-end framework for both action localization and action classification with one union backbone. SE-STAD [23] proposes a simple and efficient pipeline to build an end-to-end detector and a semi-supervised learning strategy. These studies inspire our work to redesign the

localization-based paradigm from the perspective of task settings, making it balanced and reasonable.

2.2. Object Detection

Proposal generation in STAD is very important as it represents where the action instance takes place. The localization-based paradigm usually employs a fine-tuned heavy object detector to locate actors in video frames. Object detection methods mainly fall into three categories: two-stage, one-stage, and transformer-based detection [24].

Two-stage detectors focus on accuracy while one-stage detectors focus on efficiency. Similar to STAD, two-stage detectors [5,6,25] consist of a region proposal network and a model to extract features on RoI and classify them. In contrast, one-stage methods employ the anchor-free strategy and complete detection through one unified model. As one of the best known one-stage detectors, the YOLO detector series is known for its simplicity, speed, and accuracy. YOLOv7 [26] is one of the most recent YOLO detectors, and outperforms most of the one-stage and two-stage object detectors by proposing optimized architectures and employing several trainable bag-of-freebies methods. Recently, transformer-based end-to-end object detection methods are proposed as a novel paradigm for object detection. DEtection TRansformer (DETR) [27] treats object detection as a direct set prediction problem and exploits a transformer encoder–decoder architecture. However, transformer-based detection stills suffer from model complexity and training cost due to the nature of transformer networks. In this paper, we select YOLOv7 as our person detector in TrAD.

2.3. Multi-Object Tracking

The goal of multiple object tracking (MOT) is to detect and track diverse objects in videos [28]. Current MOT methods mostly adopt the standard tracking by detection. This paradigm obtains detection results through object detection in video frames, then builds tracks by associating and assigning the bounding boxes. The tracking process can also guide object detection to achieve better results by utilizing techniques such as the Kalman filter [29]. To associate tracks and detection results, several similarity metrics, such as Intersection over Union (IoU) and ReID features, are leveraged. DeepSORT [30] enhances tracking by incorporating a CNN model for feature extraction and further conducting a ReID task based on the appearance features. ByteTrack [31] tracks objects by associating every detection box rather than the boxes with high detection scores only. Therefore, the performance of ByteTrack partly depends on the detector. However, ByteTrack does not add additional ReID models, thus restraining overall model complexity.

The idea of tracking appears in STAD as the tube construction which has a much simpler implementation [3,9,32,33]. We aim to extract this process and merge it with action localization to build the tracking-based paradigm and improve the performance of STAD.

3. Methods

In the following sections, we will explain the design and mechanism behind our action tracking, covering both the design of the tracker and the specifics of our tracking algorithm. Subsequently, we will dive into the refined action classification, discussing the principles of the Flexible Actor Proposal Scaling and detailing the feature processing strategies for tracks.

3.1. Overview

In this section, we first give an overview of our proposed framework. The framework involves action tracking and action classification as Figure 2 illustrates. TrAD starts with input video frames being fed to the feature extractor and tracker. The tracker produces actor track proposals and sends track proposals to track-aware action classification. Meanwhile, the overall features are extracted from the backbone network and sent to action classification. Then, actor track proposals are processed by Flexible Actor Proposal Scaling and combined with the overall feature for feature pooling. The ToI Align is exploited to pool per-track

features. Finally, the pooled track features are aggregated to generate final vectors and classified into certain action classes.

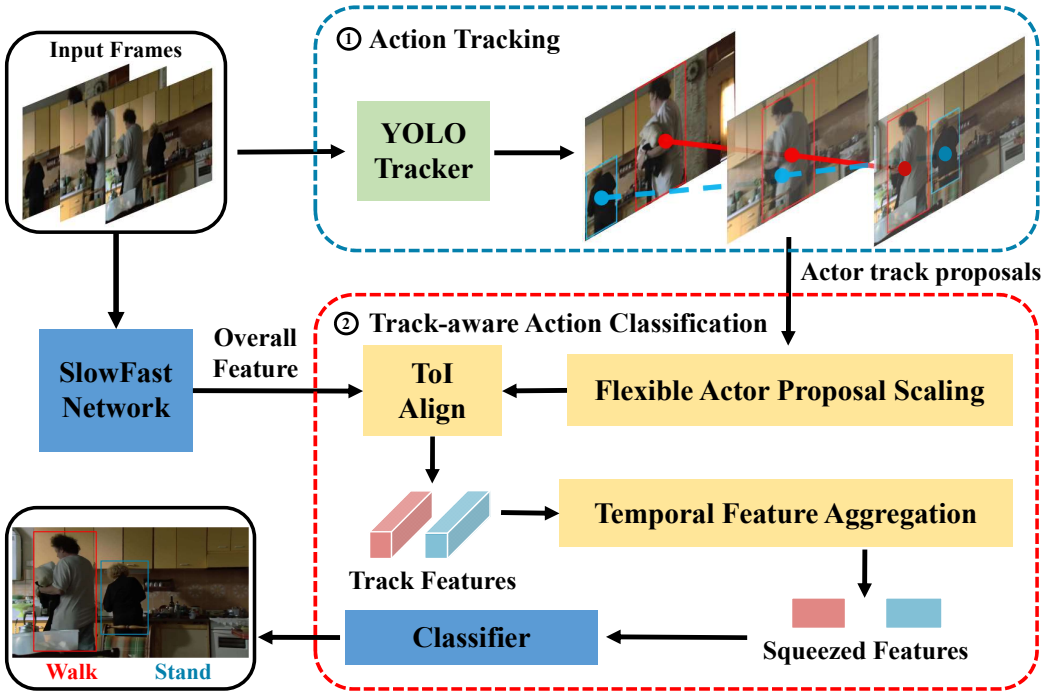

Figure 2. Overview of TrAD. TrAD conducts two-stage STAD with action tracking and track-aware action classification. 1. Action tracking employs a YOLO tracker to process input frames and generates actor track proposals. 2. Track-aware action classification includes Flexible Actor Proposal Scaling, ToI Align pooling method, and temporal feature aggregation.

In detail, we employ SlowFast Network [8] as our backbone network implementation. We extend the action localization to action tracking by utilizing a tracker to perform action localization and proposal linking simultaneously. This allows TrAD to generate video-level track proposals. We optimize the detector in the tracker by implementing YOLOv7 detection and integrating a simple and robust tracking algorithm. The proposed action tracking enables TrAD to better handle various movements of actors and also focus on detecting precise tracks instead of potential false positive actor proposals. Moreover, the generated tracks can better cope with the video-level action classification task.

Then, the track-aware feature aggregation module is proposed to fit action tracking. The module consists of Flexible Actor Proposal Scaling, ToI Align, and temporal feature aggregation. Firstly, we apply an algorithm to adjust actor bounding boxes in the generated tracks. The scaled regions can lead to better feature pooling. Secondly, instead of utilizing RoIs, we propose Tracks of Interest (ToIs) to pool actor track features. Thirdly, we propose a temporal feature aggregation strategy geared toward processing track features.

Inspired by related works [34–38], we expect our framework to be applied in spatio-temporal action detection and security surveillance.

3.2. Action Tracking

The proposed action tracking is illustrated in the blue frame in Figure 2. TrAD extends action localization to efficient and robust action tracking. Prior methods for action localization often use pretrained object detectors, mostly Faster-RCNN with a ResNeXt-101 [39] backbone. However, this approach leads to high computational overheads, and misalignment between the frame-level proposals and video-level detection adds an unnecessary tube construction process to action classification. To address these issues, TrAD combines the ideas behind action localization and tube construction to propose video-level action tracking based on multi-object tracking (MOT).

For STAD, we tailor the tracker to focus on tracking humans while being able to handle occlusion and blur due to action motion changes and camera movements during track generation. We follow the tracking-by-detection paradigm and select ByteTrack [31] as our tracking method for its simplicity. A few modifications are made to our ByteTrack to fit STAD. In our action tracking, which primarily focuses on humans, the subject of all actions, we employ a class-specific YOLOv7 as our tracker's detector to harness its efficient detection capabilities and consequently reduce the architecture's computational overhead.

In our ByteTrack, given a video sequence V and our detector YoloDET, the tracker outputs video-level actor track $\mathbf{T}_{track}$ in which the actor proposal in each frame is arranged temporally. For each frame in V, the YoloDET is employed to bring up actor proposals with detection scores. The proposals are diverted to two sets, $\mathbf{D}_h$ and $\mathbf{D}_l$, based on two preset thresholds, τ_1 and τ_2. Proposals with detection score higher than τ_1 are split into $\mathbf{D}_h$ while those with detection score lower than τ_2 are eliminated; the remainder are sent to $\mathbf{D}_l$. Then, we apply the Kalman filter for each track in $\mathbf{T}_{track}$ to predict the bounding box in the current frame. The ByteTrack performs two associations to link proposals to tracks. Instead of using IoU or ReID feature, we utilize the IoU sums of actor proposals with predicted boxes and the last proposal of the track as a matching similarity metric. In the first association, proposals in $\mathbf{D}_h$ and tracks in $\mathbf{T}_{track}$ are matched based on IoU sums through the Hungarian [40] Algorithm [41]. Unmatched proposals are assigned to $\mathbf{D}_{remain}$ to build new tracks while unmatched tracks are assigned to $\mathbf{T}_{remain}$ to perform the second association with $\mathbf{D}_l$ following the same similarity calculation. We delete unmatched proposals in $\mathbf{D}_l$ and reserve unmatched tracks with their predicted boxes to $\mathbf{T}_{lost}$ for the next 30 frames to continue matching. The proposals in $\mathbf{D}_{remain}$ are added to $\mathbf{T}$ to initialize new tracks. As shown in Figure 2, the blue line indicates a track through 3 frames. Despite the actor's occlusion in the middle frame, our tracker maintains the track and completes the tracking in the next frame.

Our YOLO tracker can make full use of detected actor proposals and handle occlusion and blur situations. After tracking, video-level track proposals are provided for track-aware action classification. This also helps filter some false positive high detection score actor proposals. The action tracking extends the action localization in the localization-based paradigm and explores the potential of actor proposals with low detection scores.

3.3. Track-Aware Action Classification

As in the red frame shown in Figure 2, the procedure of the proposed track-aware action classification can be described as follows: (1) tracks are preprocessed by the tracker; (2) track features are obtained by applying ToI Align on the overall feature with scaled tracks; (3) features are aggregated in temporal dimensions to form one-dimension vectors; (4) vectors are sent to the classifier for final action prediction. This subsection focuses on the algorithm and feature processing methods.

Flexible Actor Proposal Scaling (FAPS): We propose a simple algorithm called Flexible Actor Proposal Scaling (FAPS) to transform detected tracks for feature pooling, rather than utilizing the tracks directly. We contend that, due to the distinct motions of various actions, there will be differing aspect ratios for actor proposals. By adjusting the aspect ratio, we can obtain improved regions which leads to better feature pooling. To accomplish this, we conduct K-Means cluster analysis over annotations of three major STAD datasets, AVA [14] (https://research.google.com/ava/ accessed on 23 March 2022), UCF101-24 [13] (https://www.crcv.ucf.edu/data/UCF101.php accessed on 12 April 2022), and MultiSports [2] (https://deeperaction.github.io/datasets/multisports.html accessed on 15 April 2022), to discover the pattern of aspect ratios of ground-truth actor bounding boxes and obtain cluster centers $\{c_1 = 0.46, c_2 = 0.73, c_3 = 1.19, c_4 = 2.64\}$ for four clusters $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4$ that represent the predominant aspect ratios of actor bounding boxes, and compute three boundary points $\{\beta_1 = 0.60, \beta_2 = 0.96, \beta_3 = 1.92\}$ in between these clusters.

We give the pseudo-algorithm of FAPS in Algorithm 1. FAPS takes the input of detected tracks $\mathbf{T}$ and outputs scaled tracks $\mathbf{T}^*$. For each actor proposal α in $\mathbf{T}$, we first compute its aspect ratio γ by Equation (1), where θ_1 and θ_2 are set to 1.

$$\gamma = \frac{width \times \theta_1}{height \times \theta_2} \tag{1}$$

Algorithm 1 Flexible Actor Proposal Scaling (FAPS)

Require: Detected track $\mathbf{T}$
Ensure: Scaled track $\mathbf{T}^*$
Initialize $\mathbf{T}^* \leftarrow \varnothing$
for each actor proposal $\alpha \in \mathbf{D}$ **do**
  $\gamma \leftarrow \text{Equ1}(\alpha, \theta_1 = 1, \theta_2 = 1)$
  if min(distance of γ to boundary) < dis_{min} **then**
    $\alpha \in \beta$ // Assign the proposal to the corresponding boundary point.
  else
    distance(γ, $\mathbf{C}$). // Calculate distance of γ to each cluster.
    $\mathbf{C}_{min} \leftarrow$ min(distance) // Find the nearest cluster.
    $\alpha \in \mathbf{C}_{min}$ // Assign the proposal to the nearest cluster.
  end if
  $\theta_1, \theta_2 \leftarrow$ Value // Assign values to θ_1, θ_2.
  calculate new height and width
  scale α
  $\mathbf{T}^* \leftarrow T^* \cup \alpha^*$
end for
return $\mathbf{T}^*$

Next, we compare the aspect ratio γ to the boundary points by calculating the Euclidean distance. If the minimum distance is less than dis_{min}, we assign the actor proposal to that corresponding boundary point. The dis_{min} is set to 0.2 by default. Otherwise, we calculate the Euclidean distance between γ and the four cluster centers to assign the actor proposal to the nearest cluster. Then, the values of θ_1 and θ_2 are set according to the assignment. Specifically, if γ is assigned to boundary points or cluster $\mathbf{C}_1$, $\mathbf{C}_2$, we set θ_1 and θ_2 to 1.3 and 1, respectively. If γ is assigned to cluster $\mathbf{C}_3$, $\mathbf{C}_4$, we set θ_1 and θ_2 to 1 and 0.6, respectively. Then, we compute new height and width according to the changing of θ_1 and θ_2 in Equation (1), and scale the proposal. Note that, while scaling, the center of the actor proposal is kept unchanged, and the proposal is restricted to remain within the video frame. The scaled actor proposal α^* is added to the new track $\mathbf{T}^*$ to build our FAPS output.

Track-of-Interest Align (ToI Align): We propose Track-of-Interest Align to fit feature pooling for track-aware action classification. In TrAD, the backbone processes the input video clip and obtains an overall feature map with a shape of $T \times H \times W$. In TrAD, the backbone processes the input video clip, resulting in a feature map with a shape of $T \times H \times W$, where T denotes the temporal length of the video clip, H signifies the height and W stands for the width. Then, we pool features upon the overall feature for the scaled track whose shape is $D_n \times T \times 4$. Here, D_n represents the number of detection proposals in the track and the value 4 corresponds to the spatial coordinates of each proposal. Action classification in localization-based STAD simply extends the area around keyframes to pool features as the blue cuboid illustrates in Figure 3a.

The cuboid covers actor proposals in the first frame but misses more than half the region of the actor proposal in the last frame. Therefore, we propose ToI Align [33] to pool features for each actor proposal in the track as illustrated in Figure 3. Then, the pooled features are arranged along the temporal axis to form track features. However, the temporal length of the track and the overall feature may not be equal which further causes misalignment. To mitigate such misalignment, we propose repeating the first and

last proposals of the track forward and backward to align the track and the overall feature temporally. After ToI Align, we can get track features with the shape of $D_n \times T \times S$.

Figure 3. Illustration of RoI Align and ToI Align for STAD. The red frames and masks are used to identify the detected actor. (**a**) In the localization-based STAD, the actor proposal in the keyframe is duplicated to form the feature pooling region as the blue cuboid shows. (**b**) In TrAD, each actor proposal in detected tracks is pooled. The pooled features are then stacked as the red lines show.

Temporal Feature Aggregation: The track features pooled by ToI Align require aggregation to refine information and fit the action classifier. First, we aggregate track features spatially by applying average pooling. After spatial aggregation, we apply a layer of temporal convolution for temporal feature aggregation. This procedure helps capture the action's temporal dynamics and provides representations of the overall actor tracks. Specifically, the temporal convolution allows TrAD to extract temporal features that capture the motion and evolution of the action over time, making the representation more robust and informative for action classification.

4. Results

Before showing details of the experiment settings and the analysis of results, we will raise some research questions.

- **RQ1.** Compared to two-stage methods, does our method achieve better detection with less computation?
- **RQ2.** Does tracking improve spatio-temporal action detection?
- **RQ3.** Does the FAPS algorithm affect the spatio-temporal action detection?

First, we will explain our experimental setup, including the datasets we used and how we measured results. Next, we will compare our approach to other two-stage and one-stage methods. We will then look at how action tracking and different proposal methods affect results. Finally, we will wrap up by showing some of our findings.

4.1. Experimental Setup

4.1.1. Datasets

Due to restrictions on the devices we can access, we conduct all STAD experiments only on AVA v2.1 [14]. We select the AVA dataset as it is one of the most commonly used benchmark datasets in STAD research and it is closer to real-life scenarios. There are 215k training, 57k validation, and 120k test segments that are spread over 437 15-min. video clips. The dataset densely annotates 80 atomic visual actions over each clip at 1 Hz. The frame-level detections are linked to form action tubes and can have multiple action labels.

4.1.2. Metrics

We evaluate our TrAD's performance at the frame level in the AVA val set. We choose mean Average Precision (mAP) and giga floating point operations per second (GFLOPs) as evaluation metrics. We consider a detection as true when its IoU with a ground-truth bounding box or tube exceeds the predefined threshold of 0.5 and the detection label matches the ground-truth action label. The Average Precision (AP) for each action class is computed and we further calculate the mean among all classes to obtain the mean Average Precision (mAP). We use mAP for detection evaluation and tracking performance evaluation. The GFLOPs are used to measure the computation of action localization/tracking models and action classification models.

4.1.3. Implementation Details

We employ the YOLOv7 detector that uses YOLOv7-L as the backbone and a pretrained model on the MS-COCO dataset as initial weights to cooperate with our BYTE. We follow [31] to implement our tracker. The default detection thresholds τ_1 and τ_2 are set to 0.6 and 0.1, respectively. Following recent STAD methods [23,32], we employ the commonly used SlowFast network [8] as our backbone which allows us to better compare the performance among the TrAD and SOTA methods. The inputs are video clips with 64 frames. We use pretrained SlowFast ResNet50 8×8 as initial weights to build our TrAD. The SGD is employed for optimization while the learning rate and batch size are set to 0.01 and 32, respectively. We train the TrAD for 10 epochs, where the first epoch is the warm-up epoch and the rest of the epochs follow the cosine learning rate schedule [42]. We implement our TrAD on a platform with an NVIDIA Tesla V100 NVlink GPU, an Intel Xeon Gold 6248 CPU, and 384 GB memory.

4.2. Comparisons with State-of-the-Art Methods

We compare our proposed detector TrAD with other state-of-the-art methods in the AVA dataset in Table 1. The selected two-stage methods are the baseline in AVA [14], Actor-Centric Relation Network [17], Long-term Feature Bank [20], Context-aware RCNN [32], SlowOnly, SlowFast [8], and Actor–Relation–Actor Relation Network [16]. We implement the two most popular methods, SlowOnly and SlowFast, with different sizes of ResNet backbones.

From the results, we observe that TrAD outperforms the other two-stage methods, achieving 29.7 mAP. TrAD ranks second among the methods using the SlowFast ResNet50 as the backbone network when comparing the mAP and only trails 0.3 mAP compared to ACAR which uses an additional relation network. TrAD also exceeds SlowOnly and SlowFast with a larger backbone of ResNet101 in terms of mAP. When we replace the action classification from track-aware to frame-level in TrAD, the mAP drops approximately 5%, indicating the significance of correlation across actor tracks in feature extraction and action prediction.

In terms of computational overhead, the action tracking and action classification consume 99.7 and 109.7 GFLOPs, respectively, showing the task–model balance of our framework design. The overall model complexity reduces by 58% compared to SlowFast with ResNet50 backbone (from 209.4 to 504 GFLOPs). We also consider some recent one-stage methods such as YOWO and WOO for comparison. The TrAD raises mAP by 73% and 18% while only adding 70.1 and 67.8 more GFLOPs compared to YOWO and WOO, respectively.

Table 1 shows the comparable performance of our framework. Our method performs better compared to other SOTA two-stage and one-stage methods while using the same backbone. TrAD even achieves comparable performance with two-stage methods that use a larger backbone.

Table 1. Comparison of STAD performance on AVA dataset. The $T \times \tau$ represents the frame sampling strategy where T is the frame number and τ is the sample rate. We report GFLOPs of both models for action localization/tracking and action classification. 'CA-RCNN' is short for Context-aware RCNN. 'w.o. TAC' means track-aware action classification is replaced by frame-level action classification. The performance and overhead of TrAD are bolded.

Method	Backbone	T×τ	mAP	GFLOPs
AVA Baseline [14]	I3D-VGG	32 × 2	14.5	N/A
ACRN [17]	S3D-G	-	17.4	81.4 + 406.5
LFB [20]	R50-I3D-NL	4 × 16	25.8	N/A
CA-RCNN [32]	SFR50-NL	32 × 2	28.0	N/A
SlowOnly [8]	R50	8 × 8	19.7	42.8 + 406.5
SlowFast [8]	R50	8 × 8	25.2	97.5 + 406.5
ACAR [16]	SFR101	8 × 8	30.0	137.5 + 406.5
TrAD	SFR50	8 × 8	**29.7**	**99.7 + 109.7**
TrAD w.o. TAC	SFR50	8 × 8	28.1	97.5 + 109.7
SlowOnly [8]	R101	8 × 8	24.8	117.1 + 406.5
SlowFast [8]	R101	8 × 8	29.3	147.3 + 406.5
YOWO [21]	SFR50	8 × 8	17.1	139.3
WOO [22]	SFR50	8 × 8	25.2	141.6

4.3. Impact of the Action Tracking

We boost STAD performance by replacing action localization with action tracking. Our tracker focuses on tracking efficiency and robust track generation which further leads to better action detection, unlike tracking in [3,43]. To evaluate the effectiveness of our approach, we compared the mAP and GFLOPs of SlowFast ResNet50 and TrAD, as shown in Table 2. For the SlowFast model, we used the Faster-RCNN ResNeXt-101 to perform action localization and two trackers (YOLOv5-based DeepSort and YOLOv7-based Bytetrack) for action tracking. YOLOv7-based Bytetrack is used as the default for TrAD. As seen in Table 2, action tracking improves SlowFast by 12% and 17% in mAP by using DeepSort and Bytetrack, respectively. This performance improvement is obtained by our proposed action tracking process. Moreover, our TrAD tracker is approximately four times smaller compared to the heavy person detector (from 109.7 to 406.5 GFLOPs), which demonstrates the simplicity and effectiveness of action tracking.

We also conducted an experiment using only the proposed TrAD without action tracking and used YOLOv7 as the person detector, resulting in an approximately 11% decline in performance. The result validates our idea of task-setting alignment and the design of action tracking, which can provide high-quality actor tracks and help improve detection performance.

Table 2. Impact of proposed action tracking. We obtain mAP of STAD methods on AVA. 'w.' means using action tracking to provide actor proposal instead of action localization. 'w.o. AT' means using action localization to provide actor proposal without tracking.

Method	mAP	GFLOPs
SlowFast	25.2	97.5 + 406.5
SlowFast w. DeepSort	27.3	97.5 + 90.7
SlowFast w. Bytetrack	28.7	97.5 + 109.7
TrAD	29.7	99.5 + 109.7
TrAD w.o. AT	26.3	99.5 + 106.4

4.4. Impact of Different Proposal Strategies

In this section, we compare the impact of different proposal strategies on STAD. We conduct experiments on SlowOnly and TrAD both using ResNet50 8 × 8 as the backbone with and without different θ settings. The results are shown in Table 3.

Table 3. Comparison of different proposal strategies used in STAD methods. We try different strategies in SlowOnly R50 8 × 8 backbone [8]. ✓ and ✗ are used to clarify whether the FAPS is employed or not respectively. 'Fixed' means assigning default values to θ_1 and θ_2. 'Random' means the θ_1 and θ_2 are randomly assigned from 0.6 to 1.3.

Method	FAPS	θ_1	θ_2	mAP
SlowOnly	✓	Fixed	Fixed	20.4
	✗	-	-	19.7
TrAD	✓	Fixed	Fixed	29.7
	✓	Random	Random	27.1
	✗	-	-	28.4

We can see that the introduction of FAPS with fixed θ improves STAD performance by 3.5% and 5% on SlowOnly and TrAD, respectively. The impact of different θ settings on STAD is further investigated. We define the value setting in Algorithm 1 as the 'Fixed' assignment and the strategy of randomly assigning values from 0.6 to 1.3 to θ_1 and θ_2 as the 'Random' assignment. From the table, we can see that, compared to fixed values where θ_1 and θ_2 are set to 1.25 and 0.75, random value assignment even cut off around 3% on mAP. We believe this is caused by the unstable region transformation that leads to the intervention of irrelevant or missing information.

4.5. Qualitative Results

Lastly, we qualitatively give several examples of predictions in the AVA dataset in Figure 4. We put the keyframes in the middle with their surrounding two frames. The predicted results and ground-truth boxes are shown in yellow and red frames, respectively. We use blue frames to present tracking boxes in the surrounding frames. From Figure 4, we can see that the action instances in AVA mostly include the actors' body parts and objects in interaction, while the rapid or large shift of actors appear less as AVA is built on movie clips where actors do not need to move in many scenes. In general, our method delivers excellent action detection results as most predictions fit the ground truth.

Figure 4. Qualitative visualization example in the AVA dataset. Here we visualize a few examples of predictions made by TrAD. The keyframe is put in the middle together with the surrounding two frames. We demonstrate ground-truth boxes (red) and predicted results (yellow) in the keyframe and show the tracking boxes (blue) in the surrounding frames.

5. Conclusions

In this paper, we explore the potential of the two-stage spatio-temporal action detection paradigm. Our method balances the two tasks in the original paradigm and builds a promising paradigm for action detection. We replace action localization with action tracking to bring up track proposals instead of independent frame-level actor proposals. We further

refine the action classification by building track-level feature extraction and aggregation to adapt action tracking. By using these advantages, the proposed framework provides a new paradigm for spatio-temporal action detection. Experiments on spatio-temporal action detection in the AVA dataset show the outstanding performance of TrAD and its competitive results compared to state-of-the-art methods. We also conduct ablation experiments to investigate the effectiveness of the proposed modules which prove to be effective. We hope our upgraded two-stage paradigm can inspire more studies on the action detection problem. In the future, we plan to study how to build relations among tracks and across frames to further improve the performance of spatio-temporal action detection.

Author Contributions: Conceptualization, J.L. and Y.Y.; Formal analysis, L.C.; Funding acquisition, C.H.; Investigation, L.C.; Methodology, J.L. and Y.Y.; Project administration, C.H. and R.S.; Resources, Y.Z.; Software, J.L., Y.Y. and R.L.; Supervision, C.H. and R.S.; Validation, J.L., Y.Y. and R.L.; Writing—original draft, J.L. and Y.Y.; Writing—review and editing, C.H., H.F. and R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the High Performance Computing Center of Central South University, National Key R&D Program of China titled with "The Large-Scale Longitudinal and Cross-Sectional Study of Student Development", Project No: 2021YFC3340800, a National Natural Science Foundation (Project No: 62177046), a Hunan Educational Science "Fourteen Five-Year Plan" Project (Project No: XJK23AJD021), a Hunan Social Science Foundation project (Project No: 22YBA012), and a Central South University Graduate Education Teaching Reform Project (Project No. 2022JGB033), a Hunan Educational Science "Thirteen Five-Year Plan" Project (Project No: XJK19BXX001), and a Hunan Provincial Archives Technology Project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are openly available at https://research.google.com/ava/, accessed on 23 March 2022.

Conflicts of Interest: Author Yulin Yang was employed by the company Hunan Hanma Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Notation	Meaning
V	Video sequence
YoloDET	The YOLO object detector
$\mathbf{T}$	Actor tracks sets
$\mathbf{D}$	Actor proposals
$\tau_{1,2}$	Thresholds for track diverting
$\mathbf{C}_{1,2,3,4}$	Bounding box ratio clusters
$c_{1,2,3,4}$	Center values of the bounding box ratio clusters
$\beta_{1,2,3}$	Boundary points
γ	Bounding box ratio
T	Temporal length of a video clip
H	Height of a feature map
W	Width of a feature map

References

1. Gkioxari, G.; Malik, J. Finding action tubes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 759–768. [CrossRef]
2. Li, Y.; Chen, L.; He, R.; Wang, Z.; Wu, G.; Wang, L. MultiSports: A Multi-Person Video Dataset of Spatio-Temporally Localized Sports Actions. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, 10–17 October 2021; IEEE: Washington, DC, USA, 2021; pp. 13516–13525. [CrossRef]
3. Dave, I.R.; Scheffer, Z.; Kumar, A.; Shiraz, S.; Rawat, Y.S.; Shah, M. GabriellaV2: Towards better generalization in surveillance videos for Action Detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACV—Workshops, Waikoloa, HI, USA, 4–8 January 2022; IEEE: Washington, DC, USA, 2022; pp. 122–132. [CrossRef]
4. Sun, B.; Wu, Y.; Zhao, K.; He, J.; Yu, L.; Yan, H.; Luo, A. Student Class Behavior Dataset: A video dataset for recognizing, detecting, and captioning students' behaviors in classroom scenes. *Neural Comput. Appl.* **2021**, *33*, 8335–8354. [CrossRef]
5. Girshick, R.B. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 1440–1448. [CrossRef]
6. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; MIT Press: Cambridge, MA, USA, 2015; pp. 91–99.
7. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021.
8. Feichtenhofer, C.; Fan, H.; Malik, J.; He, K. SlowFast Networks for Video Recognition. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Washington, DC, USA, 2019; pp. 6201–6210. [CrossRef]
9. Girdhar, R.; Carreira, J.; Doersch, C.; Zisserman, A. A Better Baseline for AVA. *CoRR* **2018**, abs/1807.10066. Available online: http://xxx.lanl.gov/abs/1807.10066 (accessed on 30 March 2022).
10. Bertasius, G.; Wang, H.; Torresani, L. Is Space-Time Attention All You Need for Video Understanding? In Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Virtual Event, 18–24 July 2021; Volume 139, pp. 813–824.
11. Li, Y.; Wu, C.Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. MViTv2: Improved multiscale vision transformers for classification and detection. In Proceedings of the CVPR, New Orleans, LA, USA, 19–24 June 2022.
12. Li, K.; Wang, Y.; Peng, G.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. UniFormer: Unified Transformer for Efficient Spatial-Temporal Representation Learning. In Proceedings of the International Conference on Learning Representations, virtual, 25–29 April 2022.
13. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* **2012**, abs/1212.0402. Available online: http://xxx.lanl.gov/abs/1212.0402 (accessed on 12 April 2022).
14. Gu, C.; Sun, C.; Ross, D.A.; Vondrick, C.; Pantofaru, C.; Li, Y.; Vijayanarasimhan, S.; Toderici, G.; Ricco, S.; Sukthankar, R.; et al. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; Computer Vision Foundation/IEEE Computer Society: Washington, DC, USA, 2018; pp. 6047–6056. [CrossRef]
15. Jhuang, H.; Gall, J.; Zuffi, S.; Schmid, C.; Black, M.J. Towards understanding action recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; pp. 3192–3199.
16. Pan, J.; Chen, S.; Shou, M.Z.; Liu, Y.; Shao, J.; Li, H. Actor-context-actor relation network for spatio-temporal action localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 464–474.
17. Sun, C.; Shrivastava, A.; Vondrick, C.; Murphy, K.; Sukthankar, R.; Schmid, C. Actor-Centric Relation Network. In Proceedings of the Computer Vision—ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018; Proceedings, Part XI; Lecture Notes in Computer Science; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11215, pp. 335–351. [CrossRef]
18. Tang, J.; Xia, J.; Mu, X.; Pang, B.; Lu, C. Asynchronous interaction aggregation for action detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 71–87.
19. Zhang, Y.; Tokmakov, P.; Hebert, M.; Schmid, C. A structured model for action detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9975–9984.
20. Wu, C.Y.; Feichtenhofer, C.; Fan, H.; He, K.; Krahenbuhl, P.; Girshick, R. Long-term feature banks for detailed video understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 284–293.
21. Köpüklü, O.; Wei, X.; Rigoll, G. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv* **2019**, arXiv:1911.06644.
22. Chen, S.; Sun, P.; Xie, E.; Ge, C.; Wu, J.; Ma, L.; Shen, J.; Luo, P. Watch Only Once: An End-to-End Video Action Detection Framework. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, 10–17 October 2021; IEEE: Washington, DC, USA, 2021; pp. 8158–8167. [CrossRef]

23. Sui, L.; Zhang, C.; Gu, L.; Han, F. A Simple and Efficient Pipeline to Build an End-to-End Spatial-Temporal Action Detector. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, 2–7 January 2023; IEEE: Washington, DC, USA, 2023; pp. 5988–5997. [CrossRef]
24. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.N.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [CrossRef]
25. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. Sparse R-CNN: End-to-End Object Detection With Learnable Proposals. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, 19–25 June 2021; Computer Vision Foundation/IEEE: Washington, DC, USA, 2021; pp. 14454–14463. [CrossRef]
26. Wang, C.; Bochkovskiy, A.; Liao, H.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *CoRR* **2022**, abs/2207.02696. Available online: http://xxx.lanl.gov/abs/2207.02696 (accessed on 8 May 2022).
27. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the Computer Vision - ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part I; Lecture Notes in Computer Science; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12346, pp. 213–229. [CrossRef]
28. Marvasti-Zadeh, S.M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep Learning for Visual Tracking: A Comprehensive Survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 3943–3968. [CrossRef]
29. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
30. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: Washington, DC, USA, 2017; pp. 3645–3649.
31. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-object tracking by associating every detection box. In Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part XXII; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–21.
32. Wu, J.; Kuang, Z.; Wang, L.; Zhang, W.; Wu, G. Context-Aware RCNN: A Baseline for Action Detection in Videos. In Proceedings of the Computer Vision—ECCV 2020—16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXV; Lecture Notes in Computer Science; Vedaldi, A., Bischof, H., Brox, T., Frahm, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12370, pp. 440–456. [CrossRef]
33. Singh, G.; Choutas, V.; Saha, S.; Yu, F.; Gool, L.V. Spatio-Temporal Action Detection Under Large Motion. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, 2–7 January 2023; IEEE: Washington, DC, USA, 2023; pp. 5998–6007. [CrossRef]
34. Chen, C.; Han, D.; Shen, X. CLVIN: Complete language-vision interaction network for visual question answering. *Knowl.-Based Syst.* **2023**, *275*, 110706. [CrossRef]
35. Chen, C.; Han, D.; Chang, C.C. MPCCT: Multimodal vision-language learning paradigm with context-based compact Transformer. *Pattern Recognit.* **2024**, *147*, 110084. [CrossRef]
36. Han, D.; Zhou, H.; Weng, T.H.; Wu, Z.; Han, B.; Li, K.C.; Pathan, A.S.K. LMCA: A lightweight anomaly network traffic detection model integrating adjusted mobilenet and coordinate attention mechanism for IoT. *Telecommun. Syst.* **2023**, *84*, 549–564. [CrossRef]
37. Shi, S.; Han, D.; Cui, M. A multimodal hybrid parallel network intrusion detection model. *Connect. Sci.* **2023**, *35*, 2227780. [CrossRef]
38. Wang, H.; Han, D.; Cui, M.; Chen, C. NAS-YOLOX: A SAR ship detection using neural architecture search and multi-scale attention. *Connect. Sci.* **2023**, *35*, 1–32. [CrossRef]
39. Xie, S.; Girshick, R.B.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Washington, DC, USA, 2017; pp. 5987–5995. [CrossRef]
40. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
41. Kuhn, H.W. The Hungarian Method for the Assignment Problem. In *50 Years of Integer Programming 1958–2008—From the Early Years to the State-of-the-Art*; Jünger, M., Liebling, T.M., Naddef, D., Nemhauser, G.L., Pulleyblank, W.R., Reinelt, G., Rinaldi, G., Wolsey, L.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 29–47. [CrossRef]
42. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017; Conference Track Proceedings.
43. Weinzaepfel, P.; Harchaoui, Z.; Schmid, C. Learning to Track for Spatio-Temporal Action Localization. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 3164–3172. [CrossRef]