Noname manuscript No.
(will be inserted by the editor)

Spatiotemporal Tubelet Feature Aggregation and Object Linking for Small Object Detection in Videos

Daniel Cores · Víctor M. Brea · Manuel Mucientes

Received: date / Accepted: date

Abstract This paper addresses the problem of exploiting spatiotemporal information to improve small object detection precision in video. We propose a two-stage object detector called FANet based on short-term spatiotemporal feature aggregation and long-term object linking to refine object detections. First, we generate a set of short tubelet proposals. Then, we aggregate RoI pooled deep features throughout the tubelet using a new temporal pooling operator that summarizes the information with a fixed output size independent of the tubelet length. In addition, we define a double head implementation that we feed with spatiotemporal information for spatiotemporal classification and with spatial information for object localization and spatial classification. Finally, a long-term linking method builds long tubes with the previously calculated short tubelets to overcome detection errors. The association strategy addresses the generally low overlap between instances of small objects in consecutive frames by reducing the influence of the overlap in the final linking score. We evaluated our model in three different datasets with small objects, outperforming previous state-of-the-art spatiotemporal object detectors and our spatial baseline.

Keywords Video Object Detection · Small Object Detection · Convolutional Neural Network · Spatiotemporal CNN

1 Introduction

Object detection has been one of the most active research topics in computer vision in recent years. However, the use of temporal information in videos to boost detection precision is still an open problem. Although object detection frameworks can be executed at the frame level, they do not take advantage of temporal information available in videos that can be crucial to address challenges such as motion blur, occlusions or changes in object appearance in some frames. Addressing these issues is fundamental to solving the small object detection problem since the spatial information given by each individual frame is very limited. Therefore, any

Daniel Cores ✉ · Víctor M. Brea · Manuel Mucientes
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) - Universidade de Santiago de Compostela, Santiago de Compostela, Spain
E-mail: {daniel.cores, victor.brea, manuel.mucientes}@usc.es

partial occlusion or subtle image degradation might have a considerable impact on detection precision.

In general, object detection frameworks implement two main tasks: bounding box regression and object classification. We hypothesize that extracting spatiotemporal information from the object appearance in previous frames can significantly improve classification task accuracy. For small targets, limited spatial information makes it difficult to distinguish objects of close categories. This raises the issue of linking and aggregating spatiotemporal features throughout time.

Object detection based on convolutional neural networks (CNNs) follows two main approaches: one-stage and two-stage architectures. One-stage methods [1,2] generate candidate object locations directly from the feature maps in a dense manner. Instead, two-stage frameworks [3,4,5] use an additional network, called the region proposal network (RPN), to generate the proposals that are later refined by the network head. Our approach follows the two-stage architecture, and our spatiotemporal network uses the RPN proposals to propagate information from previous frames.

Small objects are present in many real applications. Typical videos including small targets, are those recorded from on-board cameras on unmanned aerial vehicles (UAV) or outdoors video surveillance cameras. The main challenge in these scenarios comes from a high density of small moving objects. This highly degrades the performance of state-of-the-art object detectors, especially when working with spatiotemporal object detectors that are designed to deal with few large objects per image.

Additionally, for small objects, the generally low overlap of an object between close frames and the lack of spatial information due to the small size —which is essential to distinguish between objects of different categories— hinder the feature matching and aggregation process across neighboring frames. This not only limits the benefits of exploiting spatiotemporal information of small objects with traditional video object detectors, but might also spoil the final feature map by aggregating nonrelated features. An incorrect aggregation blurs the already limited visual differences among small object categories, decreasing the final detection precision.

Our proposal follows the Faster R-CNN model but extends it to generate both temporal and spatial information to improve small object detection precision. The novelties of this work are as follows:

- A new tubelet proposal method that calculates object proposals at different levels in a feature pyramid network (FPN) model. Object proposals in the same tubelet are mapped to different pyramid levels in each frame according to their corresponding size. This makes our approach robust against size changes in consecutive frames, allowing to extract box features at different resolution levels for the same object.
- A temporal pooling method capable of summarizing information from the previous N frames, that calculates a feature map with the same size regardless of the number of input frames. Thus, it works with a fully connected network head with the same number of parameters and a constant execution time independent of N.
- A spatiotemporal double head. This component exploits both spatial information from the current frame and spatiotemporal information from many input frames. Spatial information is used to solve the object localization problem, while both spatial and spatiotemporal are combined to improve classification accuracy.
- A long-term linking algorithm that creates long tubes associating object instances throughout video frames. Then, confidence scores are updated for every detection in each tube, considering long-term spatiotemporal consistency. This method reuses short-term infor-

mation to improve the long tube creation process, overcoming network errors in certain frames such as missing detections that can otherwise break the tubes. Addressing missing detections is one of the main challenges of small object detection.

- Our framework outperforms state-of-the-art video object detectors in the USC-GRAD-STDdb [6], UAVDT [7] and VisDrone [8] datasets for the very small object subset — $\leq 256px$— defined in [6].

2 Related Work

The object detection problem with CNNs was first defined in the single image domain following both one-stage and two-stage architectures [9]. Recently, spatiotemporal frameworks were proposed based on these methods but considering the temporal information available in videos to improve detection precision.

2.1 Single image object detection

The first milestone for two-stage detectors was R-CNN [3]. R-CNN needs to apply feature extraction with a CNN on each region of a precalculated proposal set, resulting in a very slow approach. This issue was addressed in Fast R-CNN [4], by adding an RoI pooling layer on top of the CNN. Instead of executing the CNN over each proposal, Fast R-CNN extracts the features of the whole image, generating a global deep feature map. Then, RoI pooling generates a per proposal feature map extracting the corresponding features. This significantly improves both the training and test times by sharing all the CNN backbone calculations.

All these methods rely on an external region proposal method. The Faster R-CNN framework [5] defines an RPN to generate the proposal set in a fully convolutional fashion reusing the backbone calculations. This makes it possible to perform end-to-end training of the whole system without any precomputed information. The feature pyramid network [10] proposes a change in the definition of the CNN backbone, extracting feature maps at different depth levels instead of taking just the deepest level. Therefore, the RPN and the network head must calculate object proposals and the final detections at different feature map levels. FPN implements a top-down pathway and lateral connections to combine low-resolution semantically strong features with high-resolution semantically weak features to work at different levels without losing semantic meaning. Working with high-resolution feature maps enables the network to improve the small object detection precision. mSODANet [11] extends this idea by adding contextual features at multiple levels, improving the detector robustness to scale variations.

As an alternative, STDnet [6] proposes a specific architecture to address the small object detection problem. First, it selects promising areas of the image with a high probability of containing small objects. Then, a two-stage approach generates object proposals in these promising regions to calculate the final object detection set. By focusing on small portions of the image, a small stride of 4 from shallow layers can be kept without dramatically affecting the computation time. Hence, it enables to work with semantically strong high-resolution feature maps.

Many other solutions propose new header implementations based on the original Faster R-CNN framework, such as the Cascade R-CNN [12]. This method defines a multistage head that iteratively refines the object proposal set. One-stage object detectors [13,1,2] directly regress and classify *anchor boxes* without object proposals.

2.2 Video object detection

The video object detection problem has drawn the attention of the research community with new architectures specifically designed to exploit spatiotemporal information. Even so, improving the detection performance, including the temporal information available in videos, remains an open problem. The same issue also remains unsolved in related fields such as action recognition.

Some video object detection approaches rely on optical flow. For example, [14] proposed aggregating deep spatial features throughout time to improve the per-frame feature maps. To do so, the authors resort to movement information given by the optical flow to find the correspondences between the current features and the nearby feature maps. As an alternative to deep feature fusion, SVM-based spatiotemporal feature fusion [15] has been successfully applied to address the small object detection problem in low-contrast aerial environments. This approach focuses on analyzing pixel variations over time rather than the usually limited visual representations of small objects. We propose a novel method to find these correspondences working with two-stage frameworks by linking object proposals throughout time.

Several approaches have been proposed to link object detections throughout neighboring frames, making up short object tubelets. In reference [16], the authors introduced a method to link detections generated by a frame-level object detector through tracking techniques. T-CNN [17] also applied tracking to link detections of two single-frame detectors over time. The authors in [18] defined a tubelet proposal network (TPN) with two main components. First, it propagates static proposals at the frame level across time. Then, the second network estimates the bounding box displacement in each frame to build the tubelet proposal. Although this second component works with pooled features extracted from the same bounding box over time, the network can handle moving objects due to the generally large receptive field of CNNs. Instead of applying frame-level RoI pooling methods, a temporal RoI align operator was proposed in [19]. This operator performs feature aggregation between RoI features in the current frame and features from the entire feature map in the support frames. Therefore, it is not bounded by object proposals in support frames, extending the search area to the entire feature map.

Another idea is to extend the *anchor boxes* of single-frame object detectors to the spatiotemporal domain. The ACtion Tubelet detector (ACT-detector) [20] utilizes *anchor cuboids* to initialize the action tubelets. The work in [21] proposed a cuboid proposal network (CPN) for short object tubelet detection. Unlike these previous methods, in our proposal the regression of each of the *anchor boxes* in the *anchor cuboid* is performed by the corresponding RPN with information from the corresponding frame, allowing us to reuse part of the computations from previous iterations while preserving the proposals linked throughout consecutive frames.

The aim of the described methods is to link objects in the short-term. Therefore, they only take into account the nearby frames wasting long-term information. To overcome this, the approach described in reference [22] solves object linking with tracking information to build long tubes, and aggregates detection scores throughout the tube. To do that, tracking and object detection are performed and learned simultaneously with a multitask objective. Sequence level semantics aggregation (SELSA) [23] links object proposals extracted at the

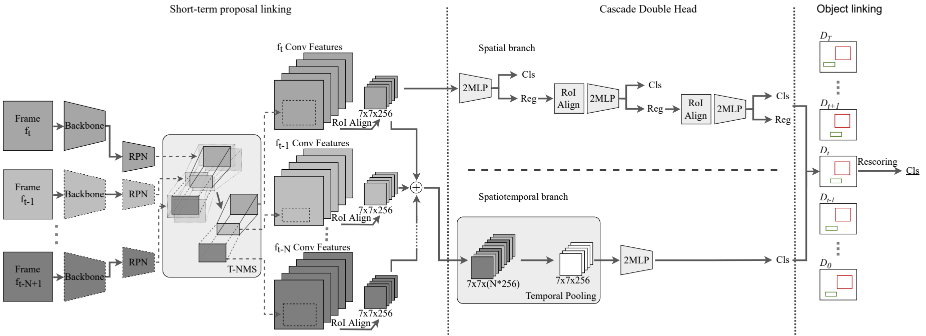

Fig. 1 FANet architecture with a single level backbone. The dotted backbone and RPN boxes represent components that are reused without new computations. Blocks labeled as 2MLP in the network head implement a multilayer perceptron with two fully connected layers.

frame level based on semantic similarities. Then, it performs feature aggregation according to these similarities, obtaining more robust and discriminative object features.

The authors in [21] also performed long-term object linking by concatenating small tubelets. First, they calculated short temporally overlapping tubelets, so one single frame could have detections associated with more than one tubelet. Then, tubelets were joined by analyzing the overlap in the shared frames. Memory enhanced global-local aggregation (MEGA) [24] extends relation distillation networks (RDN) [25] to take advantage of long-term spatiotemporal information. Both methods are based on attention mechanisms to establish the proposal relationships between different frames. STDnet-ST [26] improves STDnet [6] by adding spatiotemporal information, linking objects throughout time. Although this is a specific architecture for small object detection in videos, it is a class agnostic detector. We address this issue in our spatiotemporal double head by providing both object localization and classification.

All these methods only consider the final detection set to perform long-term object linking. In this work, we include short-term information calculated by the RPN to overcome missed detections, and build larger tubes. It is important to note that our approach only uses object proposals and detections given by the network, without any external tracking method to aid the object linking process.

3 FANet Architecture

The proposed framework (Fig.1) generalizes a single-frame two-stage object detector, adding spatiotemporal information from the nearby frames $f_{t-N+1}..., f_{t-1}, f_t$ to improve the detection precision in each frame f_t. Even though we build our system on Faster R-CNN [5] with a feature pyramid network (FPN) backbone [10] to illustrate the network architecture, the same ideas can be applied to other models. Indeed, since the spatiotemporal tubelet proposal is a core concept in our architecture, the multiscale level approach imposes higher complexity than single scale models due to the multiple RPNs. These object tubelets link proposals throughout time allowing the network to improve the per-frame features by aggregating box features from different frames. The proposal linking strategy relies on the RPN receptive field instead of overlap-based metrics or visual appearance similarities, making the method robust against moving small objects. Then, a long-term object linking leverages temporal consistency, increasing the confidence of detections that maintain spatiotemporal coherence throughout the video. Reusing short-term linking information allows this long-term method to overcome network errors such as false negatives, which are more frequent in small object detection.

We initialize object tubelets as *anchor cuboids*. Each *anchor cuboid* is a sequence of N *anchor boxes*, one per frame, with the same area and aspect ratio, in the same position. Then, the RPN modifies each *anchor box* independently in the corresponding frame using features calculated by the corresponding backbone. This, together with the fact that both the network backbone and the RPN share the convolutional weights among all input frames, allows us to reuse the backbone and RPN computations to reduce the overhead associated with the proposed spatiotemporal approach with respect to the single-frame method. Thus, feature maps and object proposals associated with frames $f_{t-N+2}, ..., f_{t-1}, f_t$ are reused to process the next frame. As in single image object detectors, the resulting proposal set has spatially redundant proposals. In our implementation, this issue is addressed by adding a tubelet non-maximum suppression (T-NMS) [21] algorithm that filters redundant tubelet proposals. The tubelet generation process is described in Sec. 3.1.

An RoI align method [27] is fed with per-frame feature maps and tubelet proposals, extracting RoI features centered on objects with a fixed size. Fig. 1 shows the simplest case in which there is a single level backbone rather than the more complex FPN. After RoI align, we concatenate all feature maps associated with proposals belonging to the same tubelet. Then, we shuffle the channels, so channels in the same position in the original feature maps are consecutive in the concatenated feature map (Fig.1). The resulting feature map has a joined dimension N times the original RoI align size, making it dependent on the number of input frames. The temporal pooling method reduces this dimension to a fixed size independent of N. The joining and pooling processes are described in Sec. 3.2.

We implement a spatiotemporal double head with spatial and spatiotemporal branches specifically designed for object classification and localization (Sec. 3.3). Spatial information from the current frame is processed in the spatial branch while spatiotemporal information is used in the spatiotemporal branch. As one of the goals of the spatial branch is to localize the object in the current frame, we follow a multistage object architecture [12] in which each stage output is the input to the next stage. Consequently, it gradually refines object proposals to maximize the overlap with the actual object. Our framework can be trained end-to-end as it does not need any precomputed or heavily engineered object proposals.

Last, per-frame object detections are linked, making up long tubes. Short-term object tubelets provide helpful information about whether two detections in different neighboring frames are the same object. The proposed long-term object linking algorithm takes this information as input to grow the final tubes. Then, the detection confidence score is updated taking into account long-term spatiotemporal consistency (Sec. 3.4).

3.1 Short-term proposal linking

In general, the starting point for most object detectors is a set of predefined *anchor boxes*. Then, they adjust these *anchor boxes* to better fit the objects and assign an object category, removing those classified as background. Instead, we propose to use *anchor cuboids* generated as N consecutive *anchor boxes*. Therefore, for a given input frame, the number of *anchor cuboids* is the same as the number of *anchor boxes* in the single image counterpart calculated as k *anchor boxes* for each sliding position $W \times H$. Moreover, every *anchor box* in an *anchor cuboid* has the same size and aspect ratio and is in the same position for all short-term input frames N.

As our framework is designed to work with an FPN (feature pyramid network) [10], object proposals are mapped to different pyramid levels according to their area. In our implementation, every proposal box $(b^{t-N+1}, ..., b^{t-1}, b^t)$ in the *anchor cuboid* is independently associated with the corresponding FPN level in each input frame following the association strategy defined in [10]. Consequently, RPN outputs for previous frames can be reused and only the new proposal b^t must be calculated. Adapting this strategy to single-level models implies that *anchor boxes* belonging to all *anchor cuboids* are mapped to the same level.

Regressing every *anchor cuboid* leads to spatially redundant tubelets in the proposal set. In two-stage single-frame detectors, this problem is generally solved executing the nonmaximum suppression (NMS) method over the proposal set. In our case, we perform a generalization that removes spatiotemporal redundant tubelet proposals instead of per image box proposals. Otherwise, applying a traditional NMS method will remove tubelet fragments breaking the short-term links. We implement an extension of the tubelet nonmaximum suppression (T-NMS), first described in [21], but with different metrics to calculate the tubelet score $ts(\tau_i)$ and to determine the overlap between two tubelets, τ_i and τ_j, making it more suitable for small objects. The goal of our T-NMS is to remove redundant tubelet proposals to support the RoI feature aggregation process. Instead, in [21], T-NMS is used to remove final detections after a per-frame refinement of *cuboid proposals*.

The score of a given tubelet τ_i is calculated taking into account the confidence of proposals that belong to τ_i:

$$ts(\tau_i) = mean(bs_i^{t-N+1}, bs_i^{t-1}, ..., bs_i^t). \tag{1}$$

where bs^t is the score of proposal b at frame t.

To calculate the overlap between two tubelets τ_i and τ_j, we use:

$$overlap(\tau_i, \tau_j) = \underset{k=t-N+1,...,t}{\text{mean}} \; IoU(b_i^k, b_j^k). \tag{2}$$

The original tubelet overlap definition is based on the *min* function, which is too demanding in the small object detection context. Instead, we use the *mean* function, reaching higher overlap values.

Tubelet scores (Eq. 1) and overlaps between a pair of tubelets (Eq. 2) are used in the T-NMS to remove tubelets with a high overlap with higher scoring tubelets. Unlike the original FPN strategy, which performs a per-level NMS, our T-NMS implementation globally removes the spatially redundant proposals, taking as input the whole set of tubelets. The resultant subset $\mathcal{T}$ represents the final collection of proposals.

3.2 RoI feature aggregation

In Faster R-CNN-based object detectors, an RoI feature pooling method takes the proposal set to produce a per proposal fixed-size feature map. Working with FPN, object proposals are distributed among the pyramid levels according to their size to perform RoI pooling over the corresponding feature map. We employ the RoI align [27] method to perform the feature pooling operation taking each bounding box b^j belonging to each tubelet $\tau_i = (b_i^{t-N+1} ..., b_i^{t-1}, b_i^t)$ to extract features from the corresponding pyramid level in frame f_j. Performing this mapping process independently in each frame rather than per tubelet enables us to map each box b^j within the same tubelet τ_i to a different pyramid level, making

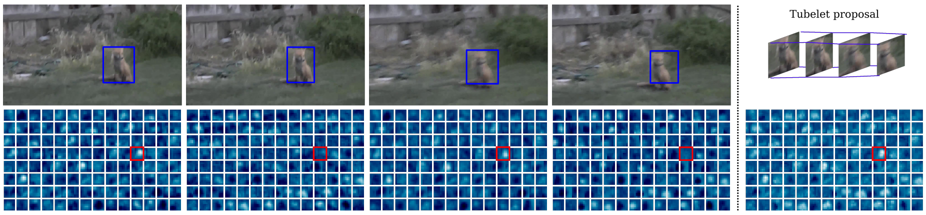

Fig. 2 Temporal pooling example with a number of input frames $N = 4$. From left to right, there are the four input frames with an object proposal (the blue bounding box) in each one, all belonging to the same tubelet. Below each input frame, a subset of channels from the RoI align output is represented. On the top right, we show the tubelet proposal linking all box proposals. At the bottom right, we also show the aggregated feature map calculated by thetemporal pooling method. We highlight one channel (framed in red) as an example of how the highest activations (lighter colors) in each frame contribute to the aggregated feature map.

the system robust against scale variations in the tubelet sequence. This allows us to integrate high-resolution feature maps from every input frame, even when objects are very small at some time instant. The results are a fixed-size feature map (in our case of $7 \times 7 \times 256$, Fig. 1) associated with each box in the tubelet.

Spatiotemporal information associated with each tubelet is summarized by a new operator called *temporal pooling* that calculates a feature map with a fixed-size independent of N. Thus, the *temporal pooling* output has the same size as the RoI align output for one single frame. This method requires N to be small enough to allow the RPN to adapt the corresponding *anchor box* in the *anchor cuboid* sequence to fit the object in each frame. Working with a large N, the object movement might exceed the RPN receptive field, making it impossible to adjust the same *anchor box* in every frame to the target object [1]. Since all RoI Align outputs that belong to the same tubelet have a fixed size and are centered in occurrences of the same object over time, we can link features in the same position of the RoI pooled feature map in every frame.

RPN errors might result in misaligned proposals throughout the tubelet that can damage the final output. However, as RoI pooled feature maps are coarse representations of the object, small variations in consecutive frames have minor effects on the *temporal pooling* inputs. This makes the short-term spatiotemporal aggregation process robust against localization errors in the RPN. These localization issues are of great importance when working with small objects. Thus, slight localization errors can dramatically reduce the overlap with the ground-truth.

The first step of the *temporal pooling* is to concatenate the N input RoI feature maps of size $W \times H \times C$, resulting in a feature map of size $W \times H \times N \cdot C$ (Fig. 1). Then, the output is reordered so that channels at the same position in the input feature maps are placed consecutively (*temporal pooling* input in Fig. 1). Finally, the output feature map is calculated as:

$$x_{ijk} = \max_{t=0\ldots,N-1} (y_{ij(N(k-1)+t)}) \tag{3}$$

where $y_{ij(N(k-1)+t)}$ is an element in the position $i \times j$ in channel $(N(k-1)+t)$ in the input feature map of size $W \times H \times N \cdot C$, and x_{ijk} is an element in position $i \times j$ in channel k of the output feature map. This process accumulates the highest activation values in the nearby frames, as Fig. 2 shows.

[1] As Section 4.3 shows, the network achieves the best result with $N = 4$.

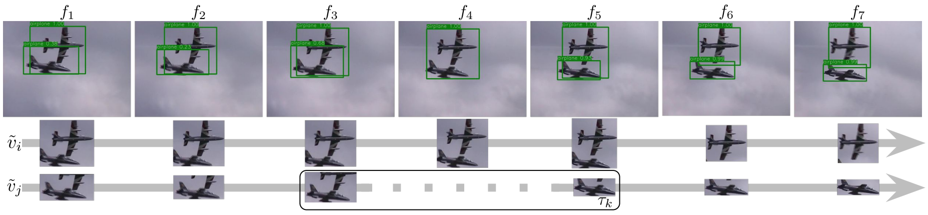

Fig. 3 Long-term object linking. The green boxes are actual network detections. The network does not detect the two objects in f_4, breaking $\tilde{v}_j$ in two fragments. The last detection of the first fragment and the first detection of the second fragment belong to the same RPN tubelet (τ_k).

3.3 Cascade double head

The double head architecture is based on the idea that aggregated spatiotemporal information is valuable to improve object classification, while spatial information extracted from the current frame is crucial for bounding box regression. Consequently, we design each head branch to perform better in its respective task taking into account the input data.

The spatial head (Fig. 1, top right) takes as input the RoI align output at the current frame f_t to calculate a spatial object classification and a class agnostic bounding box regression. Since the main goal of this branch is the object localization, we implement a cascade head [12] to iteratively refine the object proposal set until the final bounding box regression is performed.

Following the Cascade R-CNN training strategy, we perform proposal resampling after each stage, applying an increasing IoU threshold to assign each proposal to a ground truth object. Hence, the requirements to consider one proposal as a positive example to train the corresponding stage are harder as we advance in the cascade. In general, an IoU threshold that is too high might assign all proposals to background, removing the positive examples. Nevertheless, as each stage takes as input the refined proposal set from the previous stage, we can increase the IoU threshold achieving more accurate boxes in each stage. At test time, we use the average of the classification scores calculated by every stage detector over the final proposal set [12].

The spatiotemporal head (Fig. 1, bottom right) is fed with RoI features generated by our temporal pooling method. Thus, spatial information from the previous N frames is considered to classify the object in this branch. As a result, this strategy produces a bounding box regression and two object classification vectors, one based on features in the current frame and another based on the aggregated spatiotemporal features. The final classification is calculated as follows [28]:

$$p = p_{tmp} + p_{spt}(1 - p_{tmp}) \tag{4}$$

where p_{spt} and p_{tmp} are the score vectors from the spatial and temporal heads, respectively. Thus, we are considering spatial classification in the current frame and spatiotemporal classification at the same level. This makes the current frame f_t to have a greater influence on the final classification result than any previous frame f_i in $f_{t-N+1}, ..., f_{t-1}$.

3.4 Linking object detection

We propose a two-step long-term object linking algorithm that takes network detections and produces long object tubes. Linking network detections to build long tubes is a widely adopted approach in both action recognition [29,30] and object detection [22,17,21]. The main goal of these methods is to rescore all the detections in each tube, increasing the confidence to those detections that can be linked throughout time and maintaining spatiotemporal consistency in the long-term.

In the first stage of our method, linking detections in consecutive frames is addressed by maximizing the accumulated linking score in each tubelet. Network errors such as false negatives or misclassified detections might break large tubes, since it is not possible to find a detection to link in some frames. The second step of the long-term linking method utilizes the short-term tubelet information to overcome some of these problems, allowing the algorithm to produce larger tubes.

In this implementation, each detection $d_t^i = \{x_t^i, y_t^i, w_t^i, h_t^i, p_t^i\}$ in the set D_t at frame t has an associated bounding box with center (x_t^i, y_t^i), width (w_t^i), height (h_t^i), and an associated classification confidence (p_t^i) for the object class. Detections with a confidence score lower than a given threshold β are removed, reducing the probability of poor quality detections being part of the final object tubes. The linking score $ls(d^i, d^j)$ between two detections d^i and d^j at different frames t and t' is defined as:

$$ls(d_t^i, d_{t'}^j) = p_t^i + p_{t'}^j + GIoU(d_t^i, d_{t'}^j). \tag{5}$$

where GIoU is the generalized intersection over union proposed in [31]. For small objects, detections associated with the same object in nearby frames might have no overlap. GIoU allows us to measure the similarity of two bounding boxes even when they do not overlap.

Then, object tubes $\hat{v}$ can be calculated maximizing the following expression:

$$\hat{v} = \arg\max_{\mathcal{V}} \sum_{t=2}^{T} ls(D_{t-1}, D_t) \tag{6}$$

where $\mathcal{V}$ is the set of all possible tubes. This optimization problem is solved by applying the Viterbi algorithm.

Alg. 1 describes how we create long-term object tubes in detail. First, object tubes ending at frame $i = T$ are calculated by applying Equation 6 (Alg. 1:4). The selected detections are removed from $\mathcal{D}$, as they cannot be used to build new object tubes (Alg. 1:5). The process iteratively creates all the tubes ending at frame i until there are no more remaining detections at that frame. Then, the same method is applied to build tubes ending at frame $i - 1$ (Alg. 1:2).

This method creates long tubes linking consecutive object detections without considering missing detections in some frames due to network errors or occlusions. Thus, one missing detection in one specific frame would break a long tube in two parts. This naive approach is widely used in the literature in both object detection and action recognition. We propose a more robust method specifically designed to deal with these issues. Thus, in the second step of our linking algorithm, the information given by RPN tubelets links tube fragments, increasing the final size. In Fig. 3, both the last detection of the first fragment and the first detection of the second fragment of $\tilde{v}_j$ belong to the same RPN tubelet. Taking this information into account, our algorithm links the two fragments making one larger tube.

Alg. 2 describes how short-term information is added to the long-term linking process. First, we check for any pairs of tubes if they are candidates to be joined (Alg. 2:6). The

Algorithm 1: Long-term tube creation

```
Input  : Per frame detection set $\mathcal{D} = \{D_t = (d_t^1, ..., d_t^{n_t})\}_{t=1}^{T}$
Input  : All possible tubes: $\mathcal{V}$
Output: Object tubes $\hat{\mathcal{V}}$
$\hat{\mathcal{V}} \leftarrow \emptyset$
for $i$ in $T, ..., 2$ do
    while $D_i \neq \emptyset$ do
        $\hat{v} \leftarrow \arg\max_{\mathcal{V}} \sum_{t=2}^{i} ls(D_{t-1}, D_t)$
        $\mathcal{D} \leftarrow \mathcal{D} \setminus \hat{v}$
        $\hat{\mathcal{V}} \leftarrow \hat{\mathcal{V}} \cup \hat{v}$
```

Algorithm 2: Long-term object tube linking

```
   Input  : Per frame detection set $\mathcal{D} = \{D_t = (d_t^1, ..., d_t^{n_t})\}_{t=1}^{T}$
   Input  : Tubelet set $\mathcal{T} = \{\tau_i = (b_i^1, ..., b_i^N)\}_{i=1}^{\theta}$
   Input  : Object tubes $\hat{\mathcal{V}} = \{\hat{v}_i = (d^{i,1}, ..., d^{i,m_i}\}_{i=1}^{\delta}$
   Output: Joined object tubes $\tilde{\mathcal{V}}$
1  $\tilde{\mathcal{V}} \leftarrow \hat{\mathcal{V}}$
2  for $\hat{v}_i$ in $\hat{\mathcal{V}}$ do
3      for $\hat{v}_j$ in $\hat{\mathcal{V}}$ do
4          $ts_{max} = 0$
5          for $\tau_l$ in $\mathcal{T}$ do
6              if $\exists b_l^k \in \tau_l \mid \gamma(b_l^k, d^{i,m_i})$ and $\exists b_l^{k'} \in \tau_l \mid \gamma(b_l^{k'}, d^{j,1})$ and
                $time(d^{i,m_i}) > time(d^{j,1})$ then
7                  if $ts(\tau_l) > ts_{max}$ then
8                      $ts_{max} = ts(\tau_l)$
9          $\mathcal{C}_{ij} = ts_{max}$
10 $\mathcal{H} \leftarrow Hungarian(\mathcal{C})$
11 for $h_i$ in $\mathcal{H}$ do
12     $\tilde{\mathcal{V}} \leftarrow \tilde{\mathcal{V}} \setminus \hat{v}_{h_i}$
13     $\tilde{v}_i \leftarrow \tilde{v}_i \cup \hat{v}_{h_i}$
14 for $\tilde{v}_i$ in $\tilde{\mathcal{V}}$ do
15     $s = \text{mean}_{h=1,...,m_i-1}\, ls(d^{i,h}, d^{i,h+1})$
16     if $s > \lambda$ then
17         $updateScores(\tilde{v}_i)$
```

first condition they have to fulfill is that the last detection of the first tube d^{i,m_i} and the first detection of the second tube have to belong to the same short-term RPN tubelet τ_l. This is done with function $\gamma(b_l^k, d^{i,m_i})$, which checks whether a detection is associated with a box proposal b_l^k of a tubelet τ_l. This is necessary, as we apply nonmaximum suppression (NMS) and bounding box voting [32] to remove spatially redundant detections and refine the final detection set, so a detection d can be associated with several object proposals b. The second condition that has to be fulfilled is that the last detection of the first tube is previous to the first detection of the second tube.

We define a cost matrix $\mathcal{C}$ with as many rows as ending fragments, and as many columns as starting fragments. The linking score is the confidence of the tubelet proposal that links both detections and is calculated by applying Equation 1. As several tubelets might contain d^{i,m_i} and $d^{j,1}$, we select the maximum confidence among all of them (Alg. 2:7-9). Then, we solve the assignment problem by applying the Hungarian method (Alg. 2:10). The second fragment is removed from $\tilde{\mathcal{V}}$ (Alg. 2:12), and its detections are added to the corresponding first fragment, building the final long tube (Alg. 2:13). Finally, the linking score (Eq. 5) is used to calculate the average linking score for the long tube $\tilde{v}_i$ (Alg. 2:15). If the average linking score is higher than a threshold (λ), the confidence for all the detections that belong to the tube is updated to the mean confidence of the top-$\alpha\%$ detections with the highest confidence score in the tube (Alg. 2:17).

4 Experiments

4.1 Datasets

We evaluate our models on the very small object subset of three publicly available datasets, defining objects belonging to this subset as those that have an area smaller than 256 pixels:

- *USC-GRAD-STDdb dataset [33]:* this dataset contains 115 videos —92 for training and 23 for testing— with over 25,000 frames in total and 56,000 annotated small objects. Videos in this dataset are recorded in three different environments —air, sea and land— targeting 5 different classes: bird and drone (air), boat (sea), vehicle and person (land).
- *Unmanned Aerial Vehicle Benchmark (UAVDT) [7]:* this dataset is focused on videos recorded with onboard cameras on UAVs. It contains approximately 40,000 annotated frames in 50 different videos, 30 for training and 20 for testing, with objects of the vehicle class.
- *VisDrone dataset [8]:* this dataset also contains UAV recorded images in 56 training and 17 testing videos with approximately 24,000 frames in total. It has 10 different categories: pedestrian (9,255 annotated small objects), people (8,037 annotated small objects), bicycle (75 annotated small objects), car (3,639 annotated small objects), van (122 annotated small objects), truck (0 annotated small objects), tricycle (650 annotated small objects), awning-tricycle (68 annotated small objects), bus (0 annotated small objects) and motor (5,181 annotated small objects). We excluded from the test set those categories with a very low number of small objects —bicycle, van, truck, tricycle, awning-tricycle and bus—, and we fused pedestrian and people categories.

4.2 Implementation details

We use a ResNeXt-101 [34] with FPN as the backbone in all the experiments. Network weights are initialized with a pretrained model on the ImageNet classification dataset. We add to both the single frame and the spatiotemporal approach a 3-stage cascade of detectors [12] as described in Fig. 1. To train our spatiotemporal framework, we first train the single-frame model, and then we initialize the spatiotemporal network with the same weights, keeping all learned layers frozen. Thus, we only have to train the spatiotemporal head if we have the equivalent single frame model already trained.

Input images are resized by setting the smallest dimension to 720 pixels, keeping the original aspect ratio. If the largest dimension exceeds 1,280 pixels, the image is scaled down again without modifying the aspect ratio.

For the spatial baseline training, we use the SGD learning algorithm with an initial learning rate of 1.25×10^{-4}, reducing it to 1.25×10^{-5} after the first 30K iterations, and

Table 1 Contribution of each component of the network to the precision for the USC-GRAD-STDdb dataset.

Spatial head Cls	Spatiotemporal head Cls	Cascade head	Long-term object linking	$\text{AP}_{xs}^{@.5}$
✓				38.8
	✓			38.0
✓	✓			40.8
✓		✓		43.8
	✓	✓		47.0
✓	✓	✓		48.5
✓	✓	✓	✓	49.6

to 1.25×10^{-6} after the next 40K iterations. We remove redundant RPN proposals and final object detections by applying NMS, setting the IoU thresholds to 0.7 and 0.5, respectively.

As the spatiotemporal network only requires learning the spatiotemporal double head weights, it requires considerably fewer iterations. Thus, we set the number of iterations to 15K with an initial learning rate of 1.25×10^{-3} with two reductions at 10K and 14K iterations by a factor of ten. The network needs N input frames for each example: the previous $N-1$ and the current frame. For this reason, we replicate the first frame $N-1$ times to be able to process the first $N-1$ frames of each video.

Finally, we apply a bounding box voting transformation [32] and a confidence threshold $\beta = 0.05$ over the output detection set. We keep the same configuration for every dataset.

4.3 Ablation studies

We conducted a series of experiments to assess the influence of the number of input frames on the precision of the network. Moreover, we also performed a collection of ablation studies to analyze the contribution of each component of the network to the final result.

Fig. 4 shows the influence of the number of input frames N on the precision of the network on the three considered datasets. Our spatiotemporal method significantly improves the single-frame baseline, in which no temporal information is available, even when only considering one extra frame ($N = 2$). The AP generally stabilizes for a higher number of frames, being robust to small variations in N. For large values of N, the AP decreases, as the tubelet initialization based on *anchor cuboids* assumes that an object is always associated with the same *anchor box* in the same position for every input frame. When the object moves outside that scope, this assumption is not true, and this is more frequent in the case of long tubelets.

Tab. 1 shows the contribution of each network component to the final mean average precision (AP) on the USC-GRAD-STDdb dataset. In all these experiments, we obtain the bounding boxes from the spatial header, choosing the classification scores from the spatial branch, the spatiotemporal branch or the combination of both (Eq. 4). Regarding the bounding boxes, we also compare the network precision with and without refining the object proposals using a cascade of detectors. In these experiments, the architecture with only the spatial head differs from the spatial network baseline in the T-NMS method that filters the proposal set differently from the conventional NMS. Our proposal with only the spatial head reaches 38.8% AP, the cascade head improves AP by 5.0%, the short-term object linking adds 4.7% AP and, finally, the long-term object linking improves AP by 1.1%. Thus, the combination of both short- and long-term components improves the network AP

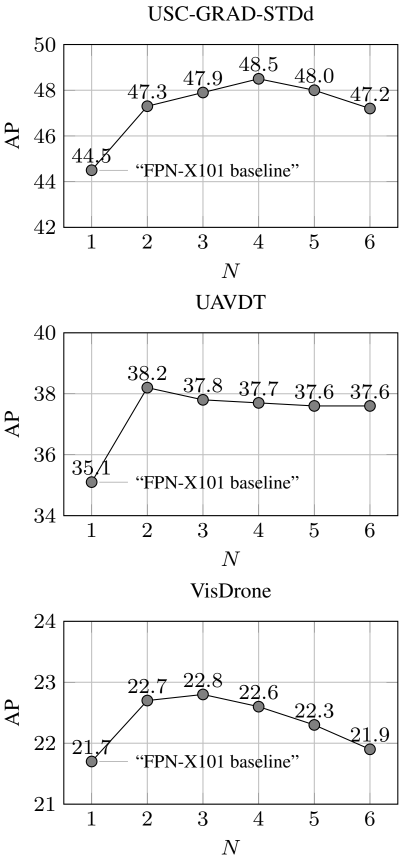

Fig. 4 Detection AP when setting the tubelet length to N without long-term information.

by 5.8%. These results prove that the proposed long-term object linking method can recover low-confidence detections that maintain spatiotemporal coherence, assigning them higher confidence than the network output. Therefore, the tube creation method —described in Alg. 1 and Alg. 2— effectively links related detections throughout the video, and the rescoring strategy specified in Alg. 2:14 - Alg. 2:17 provides more accurate detection confidence than the network output.

4.4 Results

We compare our method with state-of-the-art spatiotemporal object detectors and our single-frame baseline (Cascade R-CNN FPN-X101) on three datasets. We also modified FGFA [14], RDN [25] and MEGA [24] to work with an FPN architecture[2]. Our single-frame baseline is also based on FPN, setting a strong baseline. Following the same strategy as other video object detectors in the comparison, we also set a symmetric approach in which we

[2] Source code available at `https://github.com/daniel-cores/mega_FPN`

Table 2 USC-GRAD-STDdb results.

Method	Drone	Boat	Vehicle	Person	Bird	$\text{AP}_{xs}^{@.5}$
FGFA X101 [14]	44.1	50.6	15.4	39.8	6.6	31.3
SELSA X101 [23]	51.4	32.9	12.7	23.6	4.6	25.4
RDN X101 [25]	67.6	52.8	34.6	41.6	15.8	42.5
MEGA X101 [24]	66.0	39.2	19.9	40.3	22.1	37.5
Temporal RoI Align X101 [19]	46.1	28.0	10.0	19.5	2.6	21.2
FGFA FPN-X101	56.1	72.6	23.9	49.8	17.3	43.9
RDN FPN-X101	69.0	47.0	34.9	62.2	21.1	46.8
MEGA FPN-X101	67.5	52.6	30.7	59.3	10.7	44.1
Baseline: Cascade R-CNN FPN-X101	65.2	44.9	36.1	62.8	13.4	44.5
FANet FPN-X101 short-term (ours)	68.0	48.8	38.8	66.5	19.2	48.2
FANet FPN-X101 long-term (ours)	66.5	50.3	39.4	66.8	25.1	49.6

Table 3 UAVDT results. This dataset only contains one category (vehicle).

Method	$\text{AP}_{xs}^{@.5}$
FGFA X101 [14]	20.0
SELSA X101 [23]	19.8
RDN X101 [25]	21.5
MEGA X101 [24]	20.8
Temporal RoI Align X101 [19]	16.9
FGFA FPN-X101	26.7
RDN FPN-X101	32.4
MEGA FPN-X101	32.2
Baseline: CascadeR-CNN FPN-X101	35.1
FANet FPN-X101 short-term (ours)	37.8
FANet FPN-X101 long-term (ours)	38.2

select frames in advance. Hence, for instance, if $N = 3$, instead of selecting the 2 previous frames, we select the previous and the next one to the current frame.

Tab. 2 shows the results on the USC-GRAD-STDdb dataset. Unlike [33], [6] and [26], which give a class agnostic AP, we report the results taking into account object categories. It can be seen how our modified versions of state-of-the-art spatiotemporal frameworks significantly outperform the original versions in the small object detection domain. However, our framework achieves the best results in comparison with previous spatiotemporal work, including the FPN versions. Thus, adding both long- and short-term information leads to 49.5% AP with an IoU threshold of 0.5, the highest result in this dataset. The best modified spatiotemporal method achieves 46.8% AP, while the spatial baseline scores 44.5% AP, resulting in a difference from our approach of 2.8% and 5.1%, respectively.

Tab. 3 shows the results in the extra small subset (objects smaller than 256 pixels in area) of the UAVDT dataset. Our method achieves 37.8% AP with only short-term information and 38.2% AP considering both short- and long-term information. This result improves the single-frame baseline by 3.1% and the best spatiotemporal method by 5.8%.

The results for the VisDrone dataset are shown in Tab 4. Our approach again achieves the best results with a 22.7% AP in the extra small subset. Our method outperforms the best spatiotemporal framework by 2.1%, while the difference from the spatial baseline is 1.0%. This dataset is particularly challenging for spatiotemporal approaches due to the high object density that makes it difficult to link the same object throughout time. This hinders long-

Table 4 VisDrone results for categories with a significant number of small objects.

Method	People	Car	Motor	$AP_{xs}^{@.5}$
FGFA X101 [14]	8.2	36.0	7.2	16.8
SELSA X101 [23]	8.3	32.9	6.0	15.7
RDN X101 [25]	7.4	33.8	9.7	15.6
MEGA X101 [24]	8.5	35.9	7.4	15.5
Temporal RoI Align X101 [19]	6.5	33.1	3.8	14.5
FGFA FPN-X101	8,8	40,3	10.0	19.7
RDN FPN-X101	7.8	40.7	13.0	20.5
MEGA FPN-X101	7.8	40.6	9.3	19.2
Baseline: Cascade R-CNN FPN-X101	10.5	46.2	8.6	21.7
FANet FPN-X101 short-term (ours)	11.6	48.1	8.4	22.6
FANet FPN-X101 long-term (ours)	11.6	47.3	9.2	22.7

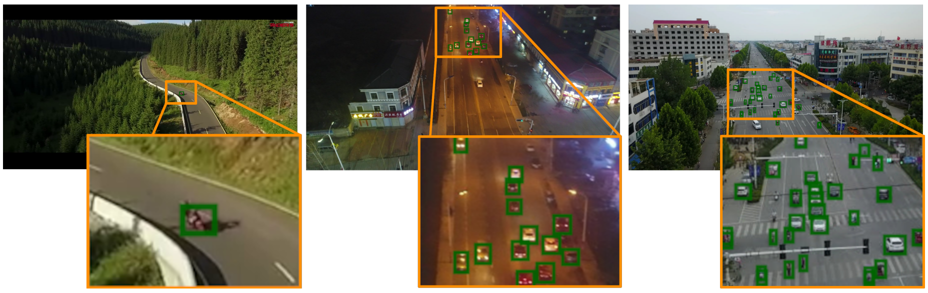

Fig. 5 Visual analysis of the results of FANet for images from USC-GRAD-STDdb, UAVDT and VisDrone (from left to right).

term linking, limiting its effect in this dataset. Fig. 5 shows detection examples of FANet for images from every dataset.

5 Conclusion

We presented a novel CNN-based framework that exploits spatiotemporal information to improve small object detection in videos. The proposal implements a feature aggregation method throughout short tubelet proposals that does not require neither tracking algorithms or optical flow. We redesigned the network head to take advantage of this aggregated spatiotemporal data with a double head implementation. The experimentation proved that this short-term information is complementary to the long-term information calculated by the proposed linking method. The overall framework outperformed the single-frame baseline and previous state-of-the-art spatiotemporal object detectors in the very small object subset of three different datasets. Therefore, it is a suitable solution for applications in which the average object size tends to be very small.

Although our tubelet initialization based on anchor cuboids provides a light computational method to link objects throughout neighboring frames, it imposes a limitation on the maximum tubelet length. In the future, we will further develop this component making it more flexible.

Acknowledgments

This research was partially funded by the Spanish Ministry of Science, Innovation and Universities under grants PID2020-112623GB-I00 and RTI2018-097088-B-C32, and the Galician Ministry of Education, Culture and Universities under grants ED431C 2018/29, ED431C 2017/69 and accreditation 2016-2019, ED431G/08. These grants are co-funded by the European Regional Development Fund (ERDF/FEDER program).

References

1. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single shot multibox detector, in: European Conference on Computer Vision (ECCV), 2016.
2. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
3. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
4. R. Girshick, Fast R-CNN, in: IEEE International Conference on Computer Vision (ICCV), 2015.
5. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems (NIPS), 2015.
6. B. Bosquet, M. Mucientes, V. M. Brea, STDnet: Exploiting high resolution feature maps for small object detection, Engineering Applications of Artificial Intelligence 91 (2020) 103615.
7. D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, Q. Tian, The unmanned aerial vehicle benchmark: Object detection and tracking, in: European Conference on Computer Vision (ECCV), 2018, pp. 370–386.
8. P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu, et al., Visdrone-det2018: The vision meets drone object detection in image challenge results, in: European Conference on Computer Vision (ECCV), 2018.
9. S. K. Pal, A. Pramanik, J. Maiti, P. Mitra, Deep learning in multi-object detection and tracking: state of the art, Applied Intelligence (2021) 1–30.
10. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
11. V. Chalavadi, P. Jeripothula, R. Datla, S. B. Ch, et al., msodanet: A network for multi-scale object detection in aerial images using hierarchical dilated convolutions, Pattern Recognition (2022) 108548.
12. Z. Cai, N. Vasconcelos, Cascade R-CNN: Delving into high quality object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
13. T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
14. X. Zhu, Y. Wang, J. Dai, L. Yuan, Y. Wei, Flow-guided feature aggregation for video object detection, in: IEEE International Conference on Computer Vision (ICCV), 2017.
15. J. Xie, C. Gao, J. Wu, Z. Shi, J. Chen, Small low-contrast target detection: Data-driven spatiotemporal feature fusion and implementation, IEEE Transactions on Cybernetics (2021).
16. K. Kang, W. Ouyang, H. Li, X. Wang, Object detection from video tubelets with convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
17. K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, et al., T-CNN: Tubelets with convolutional neural networks for object detection from videos, IEEE Transactions on Circuits and Systems for Video Technology 28 (10) (2017) 2896–2907.
18. K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, X. Wang, Object detection in videos with tubelet proposal networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
19. T. Gong, K. Chen, X. Wang, Q. Chu, F. Zhu, D. Lin, N. Yu, H. Feng, Temporal ROI Align for Video Object Recognition, in: Conference on Artificial Intelligence (AAAI), Vol. 35, 2021, pp. 1442–1450.
20. V. Kalogeiton, P. Weinzaepfel, V. Ferrari, C. Schmid, Action tubelet detector for spatio-temporal action localization, in: IEEE International Conference on Computer Vision (ICCV), 2017.
21. P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, J. Wang, Object detection in videos by high quality object linking, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019).
22. C. Feichtenhofer, A. Pinz, A. Zisserman, Detect to track and track to detect, in: IEEE International Conference on Computer Vision (ICCV), 2017.

23. H. Wu, Y. Chen, N. Wang, Z. Zhang, Sequence level semantics aggregation for video object detection, in: IEEE International Conference on Computer Vision (ICCV), 2019, pp. 9217–9225.
24. Y. Chen, Y. Cao, H. Hu, L. Wang, Memory enhanced global-local aggregation for video object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10337–10346.
25. J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, T. Mei, Relation distillation networks for video object detection, in: IEEE International Conference on Computer Vision (ICCV), 2019, pp. 7023–7032.
26. B. Bosquet, M. Mucientes, V. M. Brea, STDnet-ST: Spatio-temporal convnet for small object detection, Pattern Recognition (2021) 107929.
27. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: IEEE International Conference on Computer Vision (ICCV), 2017.
28. Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, Y. Fu, Rethinking classification and localization for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
29. G. Gkioxari, J. Malik, Finding action tubes, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
30. S. Saha, G. Singh, M. Sapienza, P. Torr, F. Cuzzolin, Deep learning for detecting multiple space-time action tubes in videos, in: British Machine Vision Conference (BMVC), 2016.
31. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 658–666.
32. S. Gidaris, N. Komodakis, Object detection via a multi-region and semantic segmentation-aware CNN model, in: IEEE International Conference on Computer Vision (ICCV), 2015.
33. B. Bosquet, M. Mucientes, V. M. Brea, STDnet: A convnet for small target detection., in: British Machine Vision Conference (BMVC), 2018.
34. S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.