Short-Term Anchor Linking and Long-Term Self-Guided Attention for Video Object Detection

Daniel Cores[a,*], Víctor M. Brea[a], Manuel Mucientes[a]

[a]*Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS) - Universidade de Santiago de Compostela, Santiago de Compostela, Spain*

Abstract

We present a new network architecture able to take advantage of spatio-temporal information available in videos to boost object detection precision. First, box features are associated and aggregated by linking proposals that come from the same *anchor box* in the nearby frames. Then, we design a new attention module that aggregates short-term enhanced box features to exploit long-term spatio-temporal information. This module takes advantage of geometrical features in the long-term for the first time in the video object detection domain. Finally, a spatio-temporal double head is fed with both spatial information from the reference frame and the aggregated information that takes into account the short- and long-term temporal context. We have tested our proposal in five video object detection datasets with very different characteristics, in order to prove its robustness in a wide number of scenarios. Non-parametric statistical tests show that our approach outperforms the state-of-the-art. Our code is available at `https://github.com/daniel-cores/SLTnet`.

Keywords: Video object detection, spatio-temporal features, Convolutional Neural Networks

1. Introduction

The advances in Convolutional Neural Networks (CNNs) have dramatically boosted the precision of single image object detectors. Nevertheless, applying single image methods directly on isolated video frames might produce unsatisfactory results due to challenges such as motion blur, out of focus or occlusions in some specific frames. Videos contain spatio-temporal information that single image object detectors do not exploit, and that can be very valuable to address these issues. Thus, to calculate the detection set for a given frame, spatio-temporal object detectors enrich features in the reference frame by analyzing a set of supporting frames that provide spatio-temporal context.

Spatio-temporal feature aggregation is a fundamental part in almost every state-of-the-art video object detector. Features from a set of supporting frames are aggregated to features in the reference frame, achieving more robust feature maps. Regarding the aggregation strategy, there are two main categories of spatio-temporal object detectors: pixel level aggregation methods [40, 34, 1, 36, 13, 24] and object level aggregation methods [18, 17, 31, 7, 8, 29, 4, 5]. Pixel level based methods aggregate information using per frame full size feature maps, while object level approaches focus on aggregating box features throughout time. Thus, the main goal of object level methods is to improve per proposal feature maps instead of improving the whole frame feature maps, concentrating on regions with high probability of containing an object.

We propose a new box level spatio-temporal object detection framework that exploits both short and long-term spatio-temporal information. First, we aggregate box features throughout nearby frames by applying a new proposal linking algorithm based on anchor boxes. We avoid short object tubelets used in previous work [18, 17, 16, 31, 5] to establish short-term relationships, providing a simpler and more efficient yet effective method. Also, we define a self-guided multi stage attention module that takes the short-term enhanced box features to establish long-term relationships. For the best of our knowledge, this is the first implementation that can handle both geometry and appearance features in long-term attention modules, since previous work such

[*] Corresponding author

Email addresses: `daniel.cores@usc.es` (Daniel Cores), `victor.brea@usc.es` (Víctor M. Brea), `manuel.mucientes@usc.es` (Manuel Mucientes) as [8] only exploits a short-term temporal context, and [4] only takes into account appearance features when it comes to long-term. Our approach shows state-of-the-art results in a wide variety of video object detection datasets.

The main contributions of this work are:

- A new short-term linking method throughout neighboring frames to associate object proposals. This method links each proposal in the reference frame with proposals that come from the same anchor in the supporting frames, relying on the Region Proposal Network (RPN) to adjust the anchor to the object in the corresponding frame.

- A new self-guided multi-stage attention module that can handle both appearance and geometry features in the long-term. Object position becomes meaningless in the long-term when it comes to compare two bounding boxes. To solve this issue, we keep track of the bounding box center, updating the bounding box position. We call this method self-guided because it reuses the attention weights from previous frames to guide the proposal tracking.

- An in-depth experimental study in five object detection video datasets with different characteristics regarding the number of objects per frame, the size of the objects, and the speed at which the position of the object changes between consecutive frames. We have compared our proposal with the state-of-the-art approaches, and we have applied a non-parametric statistical test, which shows that our method ranks first, and that the differences with the other approaches are statistically significant.

2. Related work

Image object detection. State-of-the-art single image object detectors follow two main approaches: two stage and one stage architectures.

Two stage frameworks were first popularized by R-CNN [12]. Then, Fast R-CNN [11], introduced an RoI pooling layer that allows the network to use a per image feature map instead of one for each object proposal. The generation of the object proposals was first integrated in the network by Faster R-CNN [26] defining a Region Proposal Network (RPN). Feature Pyramid Network (FPN) [20] produces feature maps of different resolutions with high-level semantics throughout by adding a top-down architecture with skip connections to Faster R-CNN. That idea has also been improved in current state-of-the-art networks like PANet [22] and EfficientDet [30].

One stage object detectors such as SSD [23] and YOLO [25] directly calculate the final detection set taking a dense grid of bounding boxes as input, instead of proposals targeting objects of interest. Therefore, these architectures must deal with a high imbalance between objects of interest and background examples in the network head. Authors in [21] propose a new cost function to deal with this issue. A more recent work [32] proposes an anchor free approach, avoiding the complicated computation related to anchor boxes.

All previous work considers object instances individually, without exploiting any relationship between them. Attention modules were first introduced in the object detection domain by [14] to model these relations. This work was motivated by the success of attention modules in natural language processing (NLP), modeling dependencies between different elements [33].

Video object detection. The main idea behind most of the state-of-the-art spatio-temporal object detection frameworks is to include feature aggregation throughout several input frames to enhance the per-frame features. Some works such as [40, 34] use optical flow information to find correspondences between features in the reference frame and features in the supporting frames. Recent methods try to avoid the optical flow calculation time, for instance by learning these correspondences based just on deformable convolutions [1]. Alternatively, [36] proposes a Recurrent Neural Network (RNN), defining a module called Spatial-Temporal Memory Module (STMM) that aggregates spatial information throughout time. A more recent work [13] proposes a new module, Progressive Sparse Local Attention (PSLA), based on attention mechanisms but working in a local fashion. All these methods aim to find correspondences and aggregate features at pixel level.

Several approaches have proposed to work at object level instead of pixel level [18, 17, 31, 7, 8, 29, 4, 5], linking objects throughout time. Object level methods aggregate only useful information in areas with high probability of containing an object. We follow this object oriented approach in our architecture.

Object tracking techniques have been applied to link detections calculated at frame level in [18, 17]. As an alternative to object tracking, a Tubelet Proposal Network (TPN) was first introduced in [16]. This network has two main steps: propagation of static proposals across time and calculation of the corresponding displacement in each frame. It takes advantage of the generally large receptive field of CNNs and the spatio-temporal redundancy between consecutive frames to be able to handle moving objects using static proposals throughout neighboring frames. A similar idea is present in [31] with the definition of a Cuboid Proposal Network (CPN) as the first step for short object tubelet detection. This CPN works with *anchor cuboids*, a spatio-temporal generalization of *anchor boxes* from the single image domain, to generate short tubelets that link objects in the short-term. Using *anchor cuboids* to link object proposals is also included in [5], and was also proposed to solve the action recognition problem in [15]. In our implementation, we avoid tubelet proposals, working directly with box proposals. This reduces the overhead of adding spatio-temporal context to single image object detectors.

As in pixel level methods, attention mechanisms have also grown in popularity among object level methods [7, 8, 29, 4]. The method described in [14] for single image object detection was extended to videos in [8], modeling relationships between object proposals across nearby frames with a multi stage attention module. This module takes into account both appearance (RoI pooling output) and geometry features (bounding box definition) to establish the relation weights. These relation weights among proposals of different frames are used to enhance box features in the current frame. The solution in [29] also searches for similar proposals in the supporting frames, focusing on long term relationships.

Previous work only considers short- or long-term information to implement attention mechanisms, but not the combination of both to take full advantage of the whole spatio-temporal context. This issue is tackled in [4] by integrating both information from nearby frames and randomly selected key frames from the entire video. Nevertheless, since original bounding box positions are not meaningful to compare proposals in distant frames, they just get rid of geometric features, proposing a location free implementation. As a step forward, we propose a new method to integrate these geometric features in the long-term aggregation method. Our approach updates bounding box positions throughout time, making possible to use previous locations to establish proposal relationships. This way, object trajectories guide the attention process, associating proposals corresponding to the same object in the past.

3. Method

We propose a new spatio-temporal framework able to improve object detection precision in videos by exploiting short- and long-term temporal context. Although our implementation is based on the two-stage object detector Faster R-CNN [26] with a Feature Pyramid Network (FPN) structure [20], the core components of our approach can be directly applied to any two-stage object detector architecture.

Two-stage single image object detector architectures take a predefined set of *anchor boxes* to initialize object proposals. Then, a Region Proposal Network (RPN) calculates the final object proposals set by modifying these *anchor boxes* to better fit the objects in the image. In addition, the RPN also gives the probability of containing an object of interest for each proposal box. Finally, spatially redundant proposals with lower confidence are removed, typically applying Non-Maximum Suppression (NMS). Our spatio-temporal framework keeps this same pipeline to initialize the per frame object proposal set. Thus, we do not add an extra overhead in the proposal generation in comparison with the single image counterpart.

Once the per-frame object proposals are calculated, they are linked throughout the nearby frames to exploit short-term information. We define two modes of operation: (i) an approach working with N input previous frames —$f_{t-N-1}, ..., f_{t-1}, f_t$— for each reference frame f_t; (ii) a symmetric approach using frames in advance taking into account $f_{t-N}, ..., f_{t-1}, f_t, f_{t+1}, ..., f_{t+N}$ for each reference frame. For the sake of simplicity we only consider the symmetric approach in further explanations. We report the precision for both approaches in the results section.

Box features in the reference frame f_t are enhanced with features in the nearby supporting frames, performing an adaptive weight feature aggregation (Fig. 1a). More details about both the linking and aggregation process are given in Section 3.1.

We also exploit long-term relations among proposals to further enrich box features. Our long-term method works with short-term aggregated features and attention mechanisms to define the long-term relationships (Fig. 1b). Section 3.2 describes this component.

Finally, the spatio-temporal double head classifies the objects of interest using both the enhanced box features and spatial features, while the bounding box regression is performed by just taking features from the reference frame (Fig. 1c). This differs from the extended trend in the state-of-the-art of applying both the bounding box regression and object classification heads over the spatio-temporal aggregated features. Even if objects in the current frame are not well defined, we argue that the most relevant information to localize the object must come from this frame. For instance, although

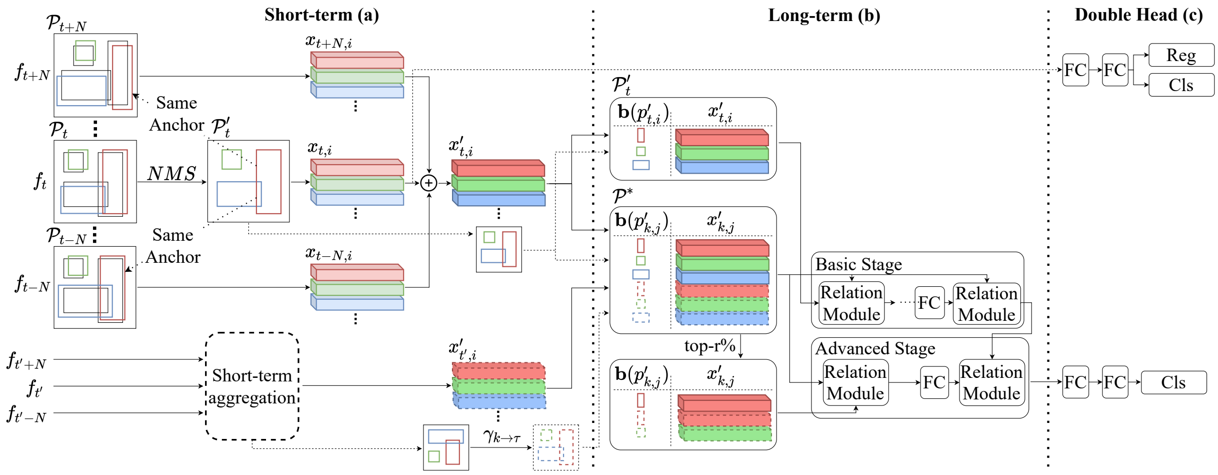
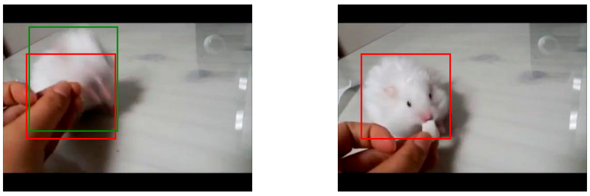


Figure 1: Our approach has three main components: (a) short-term object linking based on anchor boxes, and box aggregation features throughout the nearby frames; (b) long-term self-guided attention module that enhances short-term aggregated features with key frame information ($kf_{t'}$); (c) spatio-temporal double head.



(a) Reference frame.

(b) Supporting frame.

Figure 2: Images from ImageNet VID validation set. Red boxes represent the bounding box in the supporting frame. The green box represents the object location in the reference frame. Location information from the supporting frame (red box) is not accurate in the reference frame (green box).

Fig. 2a suffers from motion blur, making the classification task really challenging, the localization task can still be done. In fact, even though Fig. 2b gives relevant information to address the classification issue, the most relevant information to localize the object regarding its position and shape is still in Fig. 2a. The final classification score is calculated as:

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

being p_{tmp} the classification score calculated with spatio-temporal features and p_{spt} the score of the classification in the reference frame without spatio-temporal information.

3.1. Short-term anchor linking and aggregation

Our short-term module first links the proposal boxes in the nearby frames and, then, aggregates the corresponding box features. Each object proposal p_i has an associated confidence score $\mathbf{s}(p_i)$ and a bounding box $\mathbf{b}(p_i)$ used to extract box features with an RoI pooling layer —RoI Align in our implementation. These box features have the same shape regardless of the object size. Thus, the output of this short-term module keeps the same dimension as the original RoI pooling layer, independently of the number of input frames.

To link proposals in the short-term, we exploit the spatial redundancy between close frames. For every object in the image, it is very likely that the same object appears in a similar position in the nearby frames. This is a core concept in our implementation: we link proposals that come from the same *anchor box* for every frame in $\{f_{t-N}, ..., f_{t-1}, f_t, f_{t+1}, ..., f_{t+N}\}$. This method also relies on the generally large field of view of deep CNNs that allows the RPN to fit each anchor to the actual object even when the object is not very close to the predefined *anchor box*.

For every frame f_t in the input video, the RPN generates a proposal set $\mathcal{P}_t = \{p_{t,i}\}_{i=1}^{A}$, being A the total number of *anchor boxes*, calculated by multiplying the number of *anchor boxes* per position by the number of grid positions in one image. Hence, A is also the initial number of proposals per image. The final proposal set used by the network head is calculated as $\mathcal{P}'_t = top_s(NMS(\mathcal{P}_t))$. Thus, $\mathcal{P}'_t$ only contains the top s proposals ordered by confidence after removing the spatially redundant ones by means of Non-Maximum Suppression, resulting in $|\mathcal{P}'_t| \leq |\mathcal{P}_t|$. Therefore, it is very likely that there are no proposals associated with the same *anchor box* for every input frame f_j in $\{f_{t-N}, ..., f_{t-1}, f_t, f_{t+1}, ..., f_{t+N}\}$, making impossible to link proposals directly using the $\mathcal{P}'_j$ proposal sets. Instead, we keep the original proposal set $\mathcal{P}_j$ for every supporting frame $\{f_{t-N}, ..., f_{t-1}\} \cup \{f_{t+1}, ..., f_{t+N}\}$, and we link each proposal in $\mathcal{P}'_t$ with proposals from $\{\mathcal{P}_{t-N}, ..., \mathcal{P}_{t-1}\} \cup \{\mathcal{P}_{t+1}, ..., \mathcal{P}_{t+N}\}$ that came from the same *anchor box*. This process is shown in Fig. 1a: boxes with the same colors for nearby frames come from the same *anchor box*.

Then, we aggregate the box features extracted with the RoI Align method at the corresponding bounding box of each frame:

$$x'_{t,i} = \sum_{l=-N}^{N} \omega^s_{t+l,i}\, x_{t+l,i} \qquad (2)$$

being $x'_{t,i}$ the aggregated feature map, $\omega^s_{t+l,i}$ the short-term weight for the feature map that came from the proposal associated with anchor i in frame f_{t+l}, and $x_{t+l,i}$ the feature map associated with proposal i in frame f_{t+l}. The short-term weight is based on the cosine similarity metric between supporting proposals and proposals in the reference frame:

$$\omega^s_{t+l,i} = exp\left(\frac{x_{t,i}\, x_{t+l,i}}{|x_{t,i}||x_{t+l,i}|}\right) \qquad (3)$$

Then, the weights are normalized using a Softmax function to ensure that $\sum_{l=-N}^{N} \omega^s_{t+l,i} = 1$.

This goes beyond the highly effective pixel level method reported in [40] and [1], by designing a new box level method. In contrast with previous methods that use a subnetwork to calculate an intermediate feature representation of the full frame feature maps, we work directly with the box features calculated by the RoI Align method. Hence, we focus just on promising regions instead of aggregating the complete frame information, which simplifies the process.

3.2. Long-term self-guided attention module

In the long-term scenario, we cannot rely on spatial redundancy to link the proposals as in the short-term case. Therefore, we follow a more flexible approach in which every reference proposal is compared to every supporting proposal. This idea is based on attention methods proposed in [33] applied to NLP, and lately in [14] applied to the single image object detection problem. In spatio-temporal long-term aggregation, this technique allows to establish the relationship between each proposal in the reference frame and every proposal in a set of supporting key frames. In this case, supporting key frames are selected at a fixed interval I rather than consecutively as in the short-term phase. We are also considering the reference frame as a supporting frame, so that we can use proposals from the reference frame in the aggregation process.

Formally, given a proposal in the reference frame $p'_{t,i}$ and a set of supporting proposals $\mathcal{P}^{KF}$, the goal of the relation module is to calculate M relation features $\mathbf{f}^m_R$:

$$\mathbf{f}^m_R(p'_{t,i}, \mathcal{P}^{KF}) = \sum_{k=1}^{K} \sum_{j=1}^{|\mathcal{P}'_k|} w^m_{t(i),k(j)}\, (W_V\, x'_{k,j}), \quad m = 1, ..., M \qquad (4)$$

where $\mathcal{P}^{KF} = \{\mathcal{P}'_1, \mathcal{P}'_2, ..., \mathcal{P}'_K\}$, being K the number of long-term supporting key frames, and where each proposal $p'_{t,i}$ is defined by its appearance features $x'_{t,i}$ and geometry features $\mathbf{b}(p'_{t,i})$. The linear transformation matrix W_V is optimised through backpropagation in an end-to-end fashion. Previous attempts to adapt the relational module to the spatio-temporal domain [8, 4] use as appearance features the RoI pooled object proposals directly. In contrast, the appearance features that feed the relational module in our implementation are the output of the short-term aggregation process $x'_{k,j}$. This way, we can work with a more robust representation of the object.

The relational weight $w^m_{t(i),k(j)}$ is calculated as:

$$w^m_{t(i),k(j)} = \frac{g^m_{t(i),k(j)}\, exp(a^m_{t(i),k(j)})}{\sum_q g^m_{t(i),q}\, exp(a^m_{t(i),q})} \qquad (5)$$

being $a^m_{t(i),k(j)}$ the appearance weight and $g^m_{t(i),k(j)}$ the geometry weight. The appearance weight is calculated as a normalized dot product:

$$a^m_{t(i),k(j)} = \frac{\langle W_H\, x'_{t,i}, W_Q\, x'_{k,j}\rangle}{\sqrt{d_h}} \qquad (6)$$

where W_H and W_Q of Eq. 6, as well as W_G in Eq. 7, are also learnt in the training process as W_V (Eq. 4). W_H and W_Q project the appearance features in the reference frame and supporting key frames, respectively, being d_H the projected dimension.

Geometry weights are computed as:

$$g^m_{t(i),k(j)} = max\{0, W_G\, \mathcal{E}(\mathbf{b}(p'_{t,i}), \gamma_{k\to\tau}(\mathbf{b}(p'_{k,j}))))\} \qquad (7)$$

where each geometric feature $\mathbf{b}(p')$ is a 4-d vector representing the bounding box parameters (x, y, w, h) associated with proposal p'. Function $\mathcal{E}$ embeds the vector $\left(log\left(\frac{|x_i-x_j|}{w_i}\right), log\left(\frac{|y_i-y_j|}{h_i}\right), log\left(\frac{w_j}{w_i}\right), log\left(\frac{h_j}{h_i}\right)\right)$ in a high dimensional representation following the method outlined in [33]. We introduce a new function $\gamma_{k\to\tau}$ to transform geometric features from supporting key frames, so that

```
Algorithm 1: $\gamma_{k\to\tau}$
   Input  : Previous frame proposals:
            $\mathcal{P}'_\tau = \{p'_{\tau,i}\}_{i=1}^{\eta}$
   Input  : Key frame proposals: $\mathcal{P}'_k = \{p'_{k,j}\}_{j=1}^{\eta}$
   Input  : Attention weights:
            $\{w^m_{\tau(i),k(j)}\} \mid \forall p'_{\tau,i} \in \mathcal{P}'_\tau, \forall p'_{k,j} \in \mathcal{P}'_k$
1  $\overline{w}_{\tau(i),k(j)} = mean_m(w^m_{\tau(i),k(j)})$
2  for $k$ in $1, ..., K$ do
3      $\mathcal{S}_{i,j} \leftarrow \mathbf{s}(p'_{\tau,i})\, exp(\overline{w}_{\tau(i),k(j)}) \mid \forall p'_{\tau,i} \in$
        $\mathcal{P}'_\tau, \forall p'_{k,j} \in \mathcal{P}'_k$
4      $\mathcal{H} \leftarrow Hungarian(\mathcal{S})$
5      for $(i, j)$ in $\mathcal{H}$ do
6          $\mathbf{b}_{xy}(p'_{k,j}) \leftarrow \mathbf{b}_{xy}(p'_{\tau,i}) | p'_{\tau,i} \in \mathcal{P}'_\tau, p'_{k,j} \in \mathcal{P}'_k$
7  return updated $\mathcal{P}'_k$
```

they can be compared with geometric features from proposals in the reference frame. Otherwise, comparing box positions in distant frames would not be meaningful to calculate strong attention weights. This new method allows to exploit geometric features in long-term attention mechanisms for the first time in video object detection.

The core idea behind $\gamma_{k\to\tau}$ is to update proposal box positions using the attention weights to predict the object movement throughout the video by matching object proposals (Alg. 1). In doing that, this function considers the previous frame proposal set $\mathcal{P}'_\tau$ and the supporting key frame proposal set $\mathcal{P}_k$. At a certain reference frame f_t we have already calculated the relation weights that associate every proposal in $\mathcal{P}'_\tau$ with proposals in $\mathcal{P}'_k$: $w^m_{\tau(i),k(j)}$ (Eq. 5). As there are M relation weights for every pair of proposals, we aggregate them calculating the average relation weight $\overline{w}_{\tau(i),k(j)}$ (Alg. 1:1). The output of $\gamma_{k\to\tau}$ are the per key frame proposal sets $\mathcal{P}'_k$ with the proposal positions updated to the predicted position of the objects in the previous frame f_τ.

Then, to link proposals in key frames with proposals in f_τ, a score matrix $\mathcal{S}$ is populated (Alg. 1:3), taking into account proposals score $\mathbf{s}(p'_{\tau,i})$ and relational weights $\overline{w}_{\tau(i),k(j)}$. By considering the RPN confidence in f_τ, we avoid to link with low confidence proposals in the previous frame. The association problem can be solved with the Hungarian method (Alg. 1:4) [19]. Then, each bounding box position $\mathbf{b}_{xy}(p'_{k,j})$ in the supporting key frame k is updated to the corresponding position in f_τ (Alg. 1:6). $\mathbf{b}_{xy}(p')$ represents the bounding box center coordinates of proposal p'. We experimentally found that it is better to keep the original bounding box size $(w; h)$, and just updating the center coordinates. Now, Eq. 7 compares bounding box positions in consecutive frames rather than in arbitrary distant frames.

The final feature map for each proposal used by the network head is calculated as:

$$\mathbf{f}_R(p'_{t,i}, \mathcal{P}^{KF}) = \mathbf{f}_R(p'_{t,i}, \mathcal{P}^{KF}) + concat[\{\mathbf{f}^m_R(p'_{t,i}, \mathcal{P}^{KF})\}_{m=1}^{M}] \tag{8}$$

This is the concatenation of the M relational features $\mathbf{f}^m_R(p'_{t,i}, \mathcal{P}^{KF})$ (Eq. 4), and adding the result to the original proposal appearance feature $\mathbf{f}_R(p'_{t,i}, \mathcal{P}^{KF})$.

We follow a multi stage implementation similar to [8] with a set of stacked relation modules. The aim of this architecture is to iteratively refine object proposals defining two main stages, a basic stage and an advanced stage (Fig. 1b).The basic stage inputs are the top λ proposals of every key frame and the reference frame proposals $\mathcal{P}'_t$. The advanced stage has two steps. First, the top r% proposals in $\mathcal{P}^{KF}$ are enhanced by an attention module with $\mathcal{P}^{KF}$ as supporting proposals —first relation module in the advanced stage in Fig. 1b. Then, these enhanced proposals are used as supporting proposals to further improve proposal features calculated in the basic stage —second relation module in the advanced stage in Fig. 1b.

3.3. Training and inference

Both nearby frame and long-term key frame selections are implemented in a different way in the training and inference phases. This is mainly because of the ground truth availability in the training stage and the lack of constraints on which frames can be used in each moment.

As explained in Sec. 3.1, we resort to a set of neighboring frames $\{f_{t-N}, ..., f_{t-1}, f_t, f_{t+1}, ..., f_{t+N}\}$ to enhance the reference frame box features. In the inference stage, all video frames are sequentially processed by the network. Therefore, we can reuse all the backbone and RPN calculations from the nearby frames, drastically reducing the impact of enlarging the reference frame neighborhood. In contrast, in the training stage, instead of all video frames, we select a fixed size subsample of evenly spaced frames. This way, we prevent from large videos to bias the training process. In consequence, the idea of reusing computations in training cannot be applied since close frames are not selected as reference frames. Thus, the training approach is slightly different, taking just three input frames: the reference frame and two supporting frames. The two supporting frames f_{s1} and f_{s2} are randomly selected from $\{f_{t-N}, ..., f_{t-1}\}$ and $\{f_{t+1}, ..., f_{t+N}\}$ respectively.

In the long-term method we also face the same issue. Instead of several key frames, we randomly select two frames from the whole video for each reference frame. In both long- and short-term cases the number of input frames does not change the number of parameters in the network, so training and testing with different number of input frames does not need any modification in the architecture.

Moreover, the implementation in the training stage of $\gamma_{k\rightarrow\tau}$ (used in Eq. 7) also differs from the inference version described in Sec 3.2. As the network does not process frames sequentially, relational weights for previous frames are not available. However, most video object detection datasets include object identity annotations that link appearances of the same object throughout the whole video. We exploit these annotations during training to update the position of proposals in the key frame following the actual object movement. First, each object proposal is linked to the ground truth box with higher Intersection-Over-Union (IoU). Then, we apply the ground truth object translation to proposal boxes. These updates allow to use Eq. 7 during training.

4. Experimental Results

4.1. Datasets

ImageNet VID dataset [28] has become the standard benchmark to evaluate spatio-temporal object detection frameworks. In fact, most recent solutions report their metrics only on it [10, 36, 1, 34, 31, 8, 29, 13, 4]. Nevertheless, we believe that a complete and reliable evaluation requires tests in several datasets with different characteristics to assess the quality of the detector in a wide number of scenarios. In so doing, we have selected 5 different video datasets to evaluate the performance of both our proposal and state-of-the-art approaches: ImageNet VID [28], UAVDT [9], VisDrone [38], USC-GRAD-STDdb [2] and MOTChallenge [6]. As we will show in the results section (Sec. 4.4), the performance of some of these approaches highly changes with the dataset in comparison with the baseline.

There are many characteristics of the datasets that influence the detection precision. In this paper, we focus the analysis on three of them:

- Number of objects per frame (Fig. 3), which influences both spatial and spatio-temporal object detectors. Most spatio-temporal detectors exploit object relations between different frames and, therefore, a greater number of objects per frame would be preferable to assess how these methods can establish robust relationships working with many objects simultaneously.

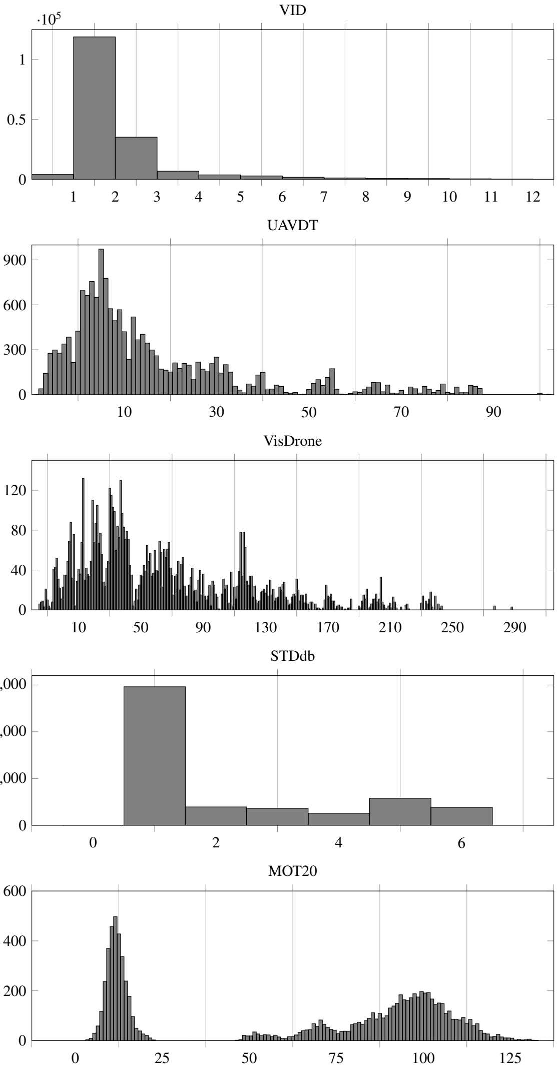


Figure 3: Average number of objects per frame for each test set.

- Size of the objects (Fig. 4), which affects both spatial and spatio-temporal detectors. In fact, small object detection is a challenge itself [3], specially for objects with areas smaller than 256 pixels ($\approx 16 \times 16$).

- The speed at which the position of the objects changes due to the own objects motion or the camera motion. This influences the performance of the spatio-temporal detectors, as most of them make feature aggregation throughout nearby frames. We measure this speed with the Intersection over

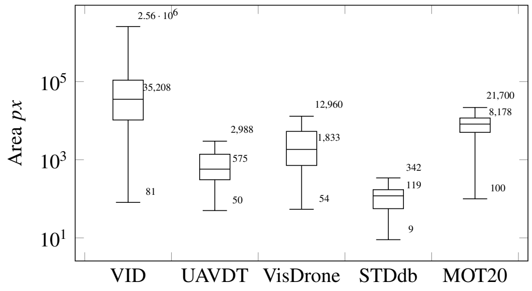


Figure 4: Box plot for the object size of each test set.

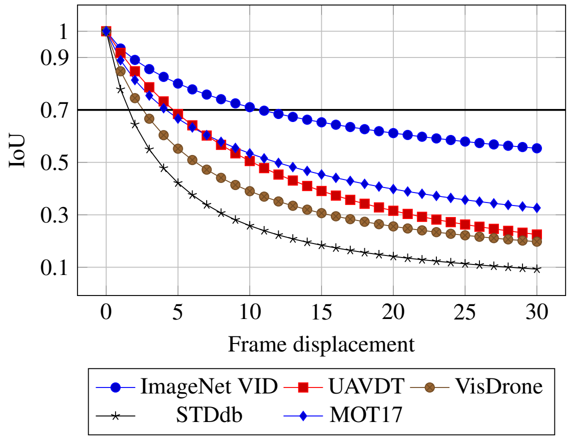


Figure 5: Mean IoU for the same object bounding boxes separated by different number of frames, and for all the training sets.

Union (IoU) of the bounding boxes of the same object in two frames (Fig. 5).

Fig. 6 shows some examples of frames extracted from different datasets. The datasets have been selected to cover a wide variety of scenarios, including low/high number of objects per frame, small/large objects, and low/fast position changing

ImageNet VID dataset [28]. It contains 30 different objects categories in 3,862 training and 555 validation videos. Following the training procedure proposed by [10], we also include data from ImageNet DET, a single image object detection dataset. This dataset contains 456,567 training and 20,121 validation images with annotated objects of 200 different categories that include the 30 classes considered in ImageNet VID. We select at most 2,000 images per VID object class from ImageNet DET to prevent from biasing the training set by including categories with a large number of images in ImageNet DET. To be able to train our spatio-temporal framework with still images, each image is repeated to create short input videos. The dataset has a very low number of objects per frame, the objects are large, and the position of the objects changes very slowly —IoUs of 0.7 on average for an object 10 frames apart.

Unmanned Aerial Vehicle Benchmark (UAVDT) [9]. It is focused on videos recorded by cameras mounted on Unmanned Aerial Vehicles (UAVs) with about 40,000 annotated frames belonging to 30 training videos and 20 testing videos, with just one category. The number of objects per image is higher than Imagenet VID, the size of the objects is medium/small, and the position of the objects changes slowly.

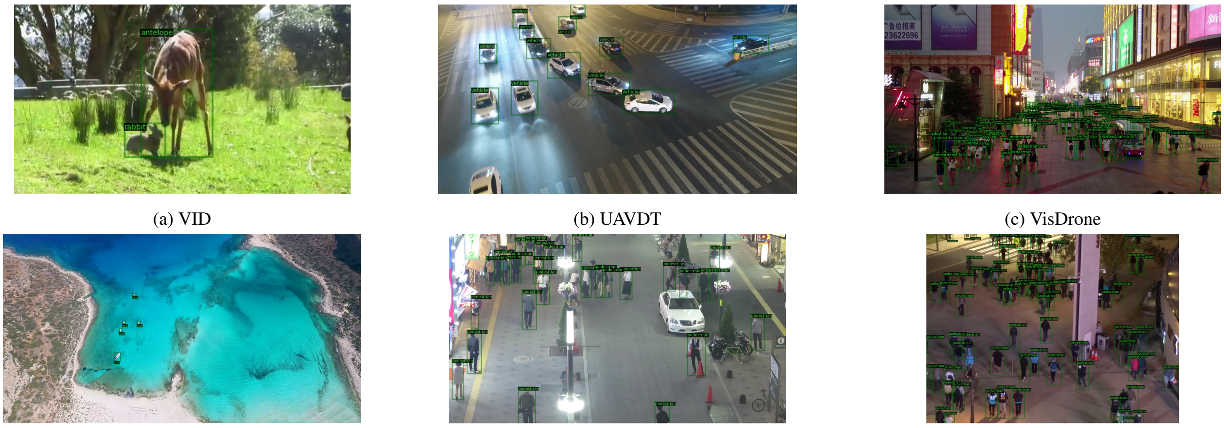
VisDrone dataset [38]. It is also focused on UAV recorded images, with 56 training videos and 17 videos for testing, containing 11 object categories. Nevertheless, it has much more objects per frame than UAVDT, and the size of the objects is medium/large. Also, the position of the objects changes fast.

USC-GRAD-STDdb dataset [2]. It is specifically designed for small object detection. It is composed of 92 training videos and 23 testing videos with over 56,000 annotated small objects of 5 different categories. The number of objects per frame is very low, and their size ranges 256 ($\approx 16 \times 16$) to as small as 16 ($\approx 4 \times 4$) pixels. Moreover, the object position changes very fast — average IoUs below 0.7 for an object in two consecutive frames— due to the small object sizes and the camera movement.

MOTChallenge [6]. It proposes pedestrian focused annotated video sequences. In this paper we train all the object detectors with the 7 training sequences from MOT17 and evaluate with the 4 sequences in the MOT20 training set. The number of objects per frame in MOT20 is considerably higher than in MOT17, changing the training and testing conditions. Following the same strategy that we use with ImageNet VID and ImageNet DET, we also add single images from CUHK-SYSU dataset [37] to the training set.

4.2. Implementation details

Our proposal has as per frame feature extractor a Feature Pyramid Network (FPN) [20] with ResNeXt-101 backbone and deformable convolutions [39] on *conv3*, *conv4* and *conv5*. We initialize the backbone with pretrain ImageNet classification weights to train the single frame baseline. To train the spatio-temporal network, we reuse the baseline weights keeping them frozen. This way, we only have to train the attention module and the temporal classification head if we have the single image counterpart trained, speeding up the training



(a) VID

(b) UAVDT

(c) VisDrone

(d) STDdb

(e) MOT17

(f) MOT20

Figure 6: Examples from the different video datasets evaluated with state-of-the-art solutions and our approach.

process dramatically. In the ablation studies we replace the ResNeXt-101 backbone by the smaller ResNet-50 due to the high number of different experiments needed.

The results of the state-of-the-art proposals in the different datasets have been obtained from: (i) the results reported by authors in their original work —ImageNet VID dataset—; and (ii) training and testing them with the implementations provided in [4] —UAVDT, VisDrone, STDdb, MOT.

For experimentation on ImageNet VID, images are scaled, so the smallest size is at most 600px. To train the single image baseline we set the base learning rate to 2.5×10^{-4} for 360K iterations, reducing by $*0.1$ at 280K and 250K iterations. For the spatio-temporal network the initial learning rate is set to 1.25×10^{-3} for 270K iterations with learning rate reductions at 210K and 250K iterations. For UAVDT, VisDrone, SDTdb and MOT we run 45K training iterations for the baseline network with an initial learning rate of 1.25×10^{-3} and learning rate reduction steps at 30K and 45K iterations. For the spatio-temporal network the number of iterations is set to 15K with a learning rate of 1.25×10^{-3} and just one reduction step at 12K iterations. We set the shortest image dimension to 720px in VisDrone, STDdb and MOT, and 540px for UAVDT, keeping the largest dimension below 1280px and 1024px, respectively.

We define a heuristic rule to set the number of short-term supporting frames N in the different datasets. The rule takes into account the object movement average in the training set to determine this hyperparameter. Based on data from Fig. 5, we select the value of N that keeps the IoU for the same object higher than 0.7 for every displacement lower than N frames. Therefore, N is set to 10 for ImageNet VID, 4 for UAVDT and MOT, 2 for VisDrone, and 1 for STDdb. We keep all the other parameters unchanged, irrespective of the dataset.

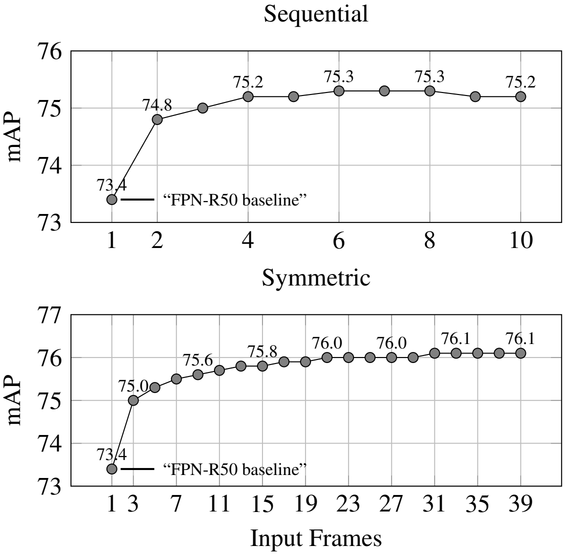


Figure 7: mAP varing the number of short-term input frames without considering long-term information.

4.3. Ablation studies

We conducted a series of ablation studies to prove that each component in our architecture is contributing to the network precision. For the sake of simplicity, we use ImageNet VID as the reference dataset for these experiments. Fig 7 shows how the number of short-term input frames affects the mAP in both sequential and symmetric modes. Moreover, Fig 7 shows how the symmetric setting yields a higher mAP than the sequential

Table 1: Long-term strategy, without short-term information.

Method	Appearance	Geometry	Mean AP
location free	✓		77.1
self-guided	✓	✓	77.3

Table 2: Influence of each component on the framework precision on ImageNet VID dataset.

Short-term	Long-term	Mean AP
		73.4
✓		76.1
	✓	77.3
✓	✓	77.8

approach. This was expected since the network can aggregate more information representing the same object in close frames than in the sequential approach. In both cases, our spatio-temporal approach shows a significant improvement over the single frame baseline: 0.6 by just using one supporting frame in the sequential case, and 1.4 with two supporting frames in the symmetric case, which is the minimum possible value in this approach.

Regarding the long-term strategy, Table 1 proves the effectiveness of considering geometry features in the long-term aggregation process. Our approach outperforms the location free version —first proposed in [4]— in which geometry features are just not taken into account. In order to evaluate the long-term method isolated, we do not include short-term aggregation in these experiments.

Finally, Table 2 shows how each component contributes to the final mAP. The results prove that short- and long-term spatio-temporal information are complementary and, thus, both are valuable to boost the object detection precision.

4.4. Results

In this section we compare our framework with the state-of-the-art spatio-temporal object detectors in the 5 selected datasets. Table 3 shows the results for the ImageNet VID dataset. We compare our model in both sequential and symmetric configurations, as well as with box level post-processing techniques. In our case, the post-processing is just box rescoring over long tubes calculated with the Viterbi algorithm over the per frame final detection set. Post-procesing methods are a special case of a symmetric approach, as they use frames in advance. However, these methods need the detection set for every frame in the video in order to be executed. Our method ranks second in the sequential mode and third in both symmetric and post-processing modes, with mAPs close to MEGA. It is important to notice that in the symmetric mode MEGA selects random frames from the whole video, while the other methods —including our proposal— just use a small number of consecutive frames in advance

For the rest of datasets, the comparison has been done with those proposals provided in [4] —FGFA [40], RDN [8], MEGA [4]; all of them in the symmetric mode, which is the best available one—, and the baseline FPN-X101. We report $mAP_{@0.5}$ and, also, $mAP_{@0.5-0.95}$, which is much more exhaustive.

Table 4 shows the results in the UAVDT dataset. Our method outperforms the best spatio-temporal object detector by 1.4 points in $mAP_{@0.5}$ and 2.5 points in $mAP_{@0.5-0.95}$. For the Visdrone dataset (Table 5) our method also achieves the best results with higher difference in $mAP_{@0.5-0.95}$. In the challenging small object detection problem (Table 6), the other spatio-temporal object detectors degrade their performance with results below our single frame baseline. Our approach achieves again the best results, overcoming the single frame baseline by 2.6 $mAP_{@0.5}$, and the best spatio-temporal framework by 4.8 $mAP_{@0.5}$. Finally, Table 7 shows the results for the MOT dataset, where our method ranks second, 0.9 $mAP_{@0.5}$ below FGFA, which is the best.

In summary, our method achieves the best results in UAVDT, Visdrone and STDdb, while MEGA [4] provides the best results in ImageNet VID —our method ranks third—, and FGFA [40] in the MOT20 dataset — our method ranks second. All in all, it can be stated that our method ranks better than the state-of-the-art in the collection of tested datasets, showing an excellent performance under very different conditions of number of objects per frame, size of the objects, and the speed the objects move in consecutive frames. In order to assess if the differences between our method and the state-of-the-art are statistically significant, we conducted a series of non-parametric tests with the STAC platform [27]. As data is non symmetric and paired, we run the Binomial Sign test comparing our proposal with each of the state-of-the-art approaches (Table 8). In this comparison we use the more robust $mAP_{@0.5-0.95}$ for every dataset except for ImageNet VID, in which we use the $mAP_{@0.5}$ originally reported by the authors. The test shows that the probability of having statistically significant differences with FGFA, RDN and MEGA is 93.75%, and of 100% with our single frame baseline. Therefore, we can conclude that, overall, our proposal outperforms the state-of-the-art, and that the differences are statistically significant.

Table 3: Results on ImageNet VID dataset.

Method	Mode	$mAP_{@0.5}$
FPN-X101 baseline	*Sequential*	*78.6*
D&T [10]	Sequential	78.7
PSLA [13]	Sequential	80.0
OGEMN [7]	Sequential	80.0
MEGA [4]	Sequential	81.9
ours	Sequential	**81.3**
FGFA [40]	Symmetric	77.8
STSN [1]	Symmetric	78.9
MANet [34]	Symmetric	78.1
SELSA [35]	Symmetric	80.3
RDN [8]	Symmetric	83.2
MEGA [4]	*Symmetric*	*84.1**
ours	Symmetric	**81.9**
D&T [10]	Post-processing	79.8
FGFA [40]	Post-processing	80.1
STSN [1]	Post-processing	80.4
MANet [34]	Post-processing	80.3
STMN [36]	Post-processing	80.5
SELSA [35]	Post-processing	80.5
PSLA[13]	Post-processing	81.4
OGEMN [7]	Post-processing	81.6
RDN [8]	Post-processing	84.7
MEGA [4]	Post-processing	85.4
ours	Post-processing	**82.4**

5. Conclusions

We have proposed a new framework for spatio-temporal object detection that takes into account both short- and long-term information. First, short-term information is linked and aggregated based on anchor association. Then, long-term information is taken into account by means of our self-guided attention module. This component allows to consider geometrical features in the long-term for the first time in the video object detection domain.

We have tested our proposal with 5 video object detection datasets, in order to analyze the performance in very different scenarios. Moreover, we have compared our approach with the state-of-the-art. Results show that our proposal ranks first on average in the collection of datasets, and non-parametric statistical tests indicate that the differences are statistically significant.

Acknowledgements

This research was partially funded by the Spanish Ministry of Science, Innovation and Universities

Table 4: Results on UAVDT dataset.

Method	$mAP_{@0.5}$	$mAP_{@0.5-0.95}$
FGFA [40]	57.6	28.9
RDN [8]	60.4	32.5
MEGA [4]	59.4	31.7
FPN-X101	59.6	33.6
ours	61.8	35.0

Table 5: Results on Visdrone dataset.

Method	$mAP_{@0.5}$	$mAP_{@0.5-0.95}$
FGFA [40]	30.7	14.1
RDN [8]	31.5	14.4
MEGA [4]	31.8	14.5
FPN-X101	31.4	15.2
ours	31.9	15.3

Table 6: Results on STDdb dataset.

Method	$mAP_{@0.5}$	$mAP_{@0.5-0.95}$
FGFA [40]	29.3	9.0
RDN [8]	40.1	13.2
MEGA [4]	37.5	12.6
FPN-X101	42.3	15.6
ours	44.9	16.6

Table 7: Results on MOT20 dataset.

Method	$mAP_{@0.5}$	$mAP_{@0.5-0.95}$
FGFA [40]	67.0	29.4
RDN [8]	66.1	26.6
MEGA [4]	46.5	17.4
FPN-X101	65.6	28.2
ours	66.1	28.4

Table 8: Binomial Sign test. p is the calculated p-value comparing each method with our proposal.

	FPN	FGFA	RDN	MEGA
$1 - p$	1.000	0.9375	0.9375	0.9375

under grants TIN2017-84796-C2-1-R 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] Bertasius, G., Torresani, L., Shi, J., 2018. Object detection in video with spatiotemporal sampling networks, in: IEEE International Conference on Computer Vision (ICCV).

[2] Bosquet, B., Mucientes, M., Brea, V.M., 2018. STDnet: A convnet for small target detection., in: British Machine Vision Conference (BMVC).

[3] Bosquet, B., Mucientes, M., Brea, V.M., 2020. STDnet: Exploiting high resolution feature maps for small object detection. Engineering Applications of Artificial Intelligence 91, 103615.

[4] Chen, Y., Cao, Y., Hu, H., Wang, L., 2020. Memory enhanced global-local aggregation for video object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10337–10346.

[5] Cores, D., Mucientes, M., Brea, V.M., 2020. RoI feature propagation for video object detection., in: European Conference on Artificial Intelligence (ECAI).

[6] Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L., 2020. Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 .

[7] Deng, H., Hua, Y., Song, T., Zhang, Z., Xue, Z., Ma, R., Robertson, N., Guan, H., 2019a. Object guided external memory network for video object detection, in: IEEE International Conference on Computer Vision (ICCV), pp. 6678–6687.

[8] Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T., 2019b. Relation distillation networks for video object detection, in: IEEE International Conference on Computer Vision (ICCV), pp. 7023–7032.

[9] Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., Tian, Q., 2018. The unmanned aerial vehicle benchmark: Object detection and tracking, in: European Conference on Computer Vision (ECCV), pp. 370–386.

[10] Feichtenhofer, C., Pinz, A., Zisserman, A., 2017. Detect to track and track to detect, in: IEEE International Conference on Computer Vision (ICCV).

[11] Girshick, R., 2015. Fast R-CNN, in: IEEE International Conference on Computer Vision (ICCV).

[12] Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[13] Guo, C., Fan, B., Gu, J., Zhang, Q., Xiang, S., Prinet, V., Pan, C., 2019. Progressive sparse local attention for video object detection, in: IEEE International Conference on Computer Vision (ICCV), pp. 3909–3918.

[14] Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y., 2018. Relation networks for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3588–3597.

[15] Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C., 2017. Action tubelet detector for spatio-temporal action localization, in: IEEE International Conference on Computer Vision (ICCV).

[16] Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., Wang, X., 2017a. Object detection in videos with tubelet proposal networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[17] Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Zhang, C., Wang, Z., Wang, R., Wang, X., et al., 2017b. T-CNN: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology 28, 2896–2907.

[18] Kang, K., Ouyang, W., Li, H., Wang, X., 2016. Object detection from video tubelets with convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[19] Kuhn, H.W., 1955. The hungarian method for the assignment problem. Naval research logistics quarterly 2, 83–97.

[20] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[21] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[22] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8759–8768.

[23] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. SSD: Single shot multibox detector, in: European Conference on Computer Vision (ECCV).

[24] Mhalla, A., Chateau, T., Amara, N.E.B., 2019. Spatio-temporal object detection by deep learning: Video-interlacing to improve multi-object tracking. Image and Vision Computing 88, 120–131.

[25] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[26] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems (NIPS).

[27] Rodríguez-Fdez, I., Canosa, A., Mucientes, M., Bugarín, A., 2015. STAC: a web platform for the comparison of algorithms using statistical tests, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

[28] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115, 211–252. doi:10.1007/s11263-015-0816-y.

[29] Shvets, M., Liu, W., Berg, A.C., 2019. Leveraging long-range temporal relationships between proposals for video object detection, in: IEEE International Conference on Computer Vision (ICCV), pp. 9756–9764.

[30] Tan, M., Pang, R., Le, Q.V., 2020. Efficientdet: Scalable and efficient object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10781–10790.

[31] Tang, P., Wang, C., Wang, X., Liu, W., Zeng, W., Wang, J., 2019. Object detection in videos by high quality object linking. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[32] Tian, Z., Shen, C., Chen, H., He, T., 2019. FCOS: Fully convolutional one-stage object detection, in: IEEE International Conference on Computer Vision (ICCV), pp. 9627–9636.

[33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, in: Advances in neural information processing systems, pp. 5998–6008.

[34] Wang, S., Zhou, Y., Yan, J., Deng, Z., 2018. Fully motion-

aware network for video object detection, in: IEEE International Conference on Computer Vision (ICCV).

[35] Wu, H., Chen, Y., Wang, N., Zhang, Z., 2019. Sequence level semantics aggregation for video object detection, in: IEEE International Conference on Computer Vision (ICCV), pp. 9217–9225.

[36] Xiao, F., Jae Lee, Y., 2018. Video object detection with an aligned spatial-temporal memory, in: European Conference on Computer Vision (ECCV).

[37] Xiao, T., Li, S., Wang, B., Lin, L., Wang, X., 2017. Joint detection and identification feature learning for person search, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3415–3424.

[38] Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q., 2018. Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437 .

[39] Zhu, X., Hu, H., Lin, S., Dai, J., 2019. Deformable convnets v2: More deformable, better results, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9308–9316.

[40] Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y., 2017. Flow-guided feature aggregation for video object detection, in: IEEE International Conference on Computer Vision (ICCV).