Pattern Recognition 116 (2021) 107929

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

STDnet-ST: Spatio-temporal ConvNet for small object detection

Brais Bosquet*, Manuel Mucientes, Víctor M. Brea

Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain

ARTICLE INFO

Article history:
Received 19 March 2020
Revised 5 October 2020
Accepted 28 February 2021
Available online 10 March 2021

Keywords:
Small object detection
Spatio-temporal convolutional network
Object linking

ABSTRACT

Object detection through convolutional neural networks is reaching unprecedented levels of precision. However, a detailed analysis of the results shows that the accuracy in the detection of small objects is still far from being satisfactory. A recent trend that will likely improve the overall object detection success is to use the spatial information operating alongside temporal video information. This paper introduces STDnet-ST, an end-to-end spatio-temporal convolutional neural network for small object detection in video. We define small as those objects under 16×16 px, where the features become less distinctive. STDnet-ST is an architecture that detects small objects over time and correlates pairs of the top-ranked regions with the highest likelihood of containing those small objects. This permits to link the small objects across the time as tubelets. Furthermore, we propose a procedure to dismiss unprofitable object links in order to provide high quality tubelets, increasing the accuracy. STDnet-ST is evaluated on the publicly accessible USC-GRAD-STDdb, UAVDT and VisDrone2019-VID video datasets, where it achieves state-of-the-art results for small objects.

1. Introduction

Over the last years, the scope of object detection has witnessed significant progress [1]. Most of the state-of-the-art methods share a similar two-stage structure adopting the Faster R-CNN [2] approach, where a deep convolutional neural network (ConvNet) backbone is firstly applied to generate a set of feature maps over the whole input image followed by a detection-specific network [3–5] that provides the detection results from the feature maps.

Small object detection, typically defined as objects with a size below 32×32 pixels in widely adopted image datasets as MS COCO [6], is progressively gaining more interest in the scientific community [7–9]. This permits to tackle practical applications as sense and avoid on board of Unmanned Aerial Vehicles (UAVs), or video surveillance tasks where early actions are required. Small object detection accuracy lags behind that of larger objects [10], which opens the way for more improvement. This is in part due to the lack of specific architectures and datasets, with the exception of face detection, where objects are usually of small size, which makes up a field of interest by itself [9,11].

The lack of specific datasets with small objects has been partially addressed with the rise of UAVs with built-in cameras to record wide areas in the wild with small objects and decent quality. In particular, UAVDT [12], VisDrone2019-VID [13] and, especially, USC-GRAD-STDdb [7] are video datasets with a large percentage of small objects.

Video object detection has had a recent upturn with the advent of ImageNet video object detection challenge (VID) [14], leading to spatio-temporal ConvNets [15]. These networks have been tried to exploit the richer information from several frames when compared to static images. Linking the same objects across video to form sequences, or tubelets, to improve the classification score has proved to be the most efficient technique [16–18] among the different ways to tackle this issue [19–22].

This paper addresses small object detection with STDnet-ST, a novel spatio-temporal convolutional neural network aimed at video small object detection. STDnet-ST is built on STDnet [7]. STDnet is a fully convolutional neural network which provides the most likely areas of the image with small objects. Once the most promising areas with objects are selected, the rest of the image is dismissed, allowing to keep high level of detail in those selected areas without affecting the computational performance. In this paper, we define small objects as any potentially moving object of less than 16×16 pixels without definitive visual cues to assign them to a category, following our previous work [7].

The main contributions of this work are (Fig. 1):

- STDnet-ST, a spatio-temporal neural network built on STDnet for small object detection that operates with two input frames simultaneously. Both inputs are integrated together through a

* Corresponding author.
E-mail addresses: brais.bosquet@usc.es (B. Bosquet), manuel.mucientes@usc.es (M. Mucientes), victor.brea@usc.es (V.M. Brea).

https://doi.org/10.1016/j.patcog.2021.107929
0031-3203/

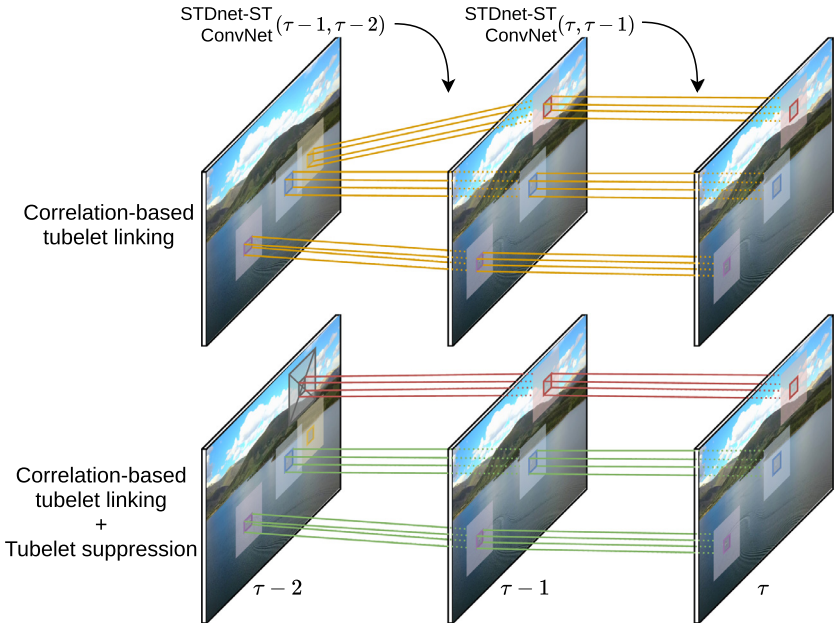

Fig. 1. STDnet-ST has two components: *STDnet-ST ConvNet* and *STDnet-ST tubelet linking*. *STDnet-ST ConvNet* performs small object detection and correlation over two consecutive frames. *STDnet-ST tubelet linking* creates tubelets in two stages: first, the correlation-based tubelet linking creates tubelets (orange) across the last τ frames; then, tubelet suppression, generates additional nodes (⊠) to avoid unprofitable tubelets (red) while providing high quality ones (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

correlation module at shallower layers and a final tubelet linking.
  - The spatio-temporal ConvNet simultaneously generates the detections of the current frame, together with the correlations between the current and previous frames. The correlation is performed in a natural way over the most promising regions of the image, i.e., regions provided by the shallowest layers of our network with a high likelihood of having objects.
  - The tubelet linking is based on the Viterbi algorithm, but we include three novelties. First, it uses the correlations generated by the ConvNet to link the objects of the tubelet. Second, it scores the associations between the objects, taking into account the confidence variability of the tubelet, which is an indicator of the tubelet confidence. Third, the tubelet suppression algorithm avoids unprofitable tubelets. This is achieved by inserting additional nodes to each frame in the Viterbi algorithm based on the information coming from promising areas without detections. All these contributions allow STDnet-ST to increase the confidence of the detections most likely to be true positives within high quality tubelets and decrease the confidence of those most likely to be false positives within unprofitable tubelets.
- STDnet-ST achieves state-of-the-art results for small object detection on the publicly available datasets USC-GRAD-STDdb, UAVDT and VisDrone2019-VID, over the very small object subset *XS* ($\leq$ 256 px), defined in Bosquet et al. [7].

2. Related work

The image object detection scope has followed two parallel trends: region proposal based detectors (*two-stages*), according to the milestone set by Faster R-CNN [2], and detectors that directly predict boxes from feature maps (*one-shot* or *one-stage*), with SSD [23] and YOLO [24] as pioneers. A large number of outstanding improvements have been derived from these architectures, being the two-stage Feature Pyramid Network (FPN) [25] noteworthy since it remains as the baseline of the leading solutions in the COCO object detection challenge[1].

The trend in small image object detection is to work on data from as fine a feature map as possible, where small objects still have distinctive features. In this line, FPN's success in the small object subset of MS COCO is mainly based on merging feature maps at different scales with a Region Proposal Network (RPN) per scale [2]. Here, the coarsest RPN makes use of a shallow feature map with stride 4, preserving fine details. In contrast, architectures like Faster R-CNN present stride 16 as a starting point to seek for objects, which might not suffice for a good accuracy in small object detection. Following the same idea, RetinaNet [5] is an FPN-based architecture that removes RPNs and adds two subnetworks —class subnet and bounding box subnet— to detect objects in one-stage, including small objects. The main improvement is obtained through a novel loss function (Focal Loss) to address the class imbalance in one-shot detectors. Recently, Yuan et al. [26] studies how to optimize the feature map multi-scale integration by using *gates* to extract only useful semantic information, resulting in a more effective feature map for object detection.

Similarly, our previous approach, STDnet [7], is a ConvNet for image object detection able to keep a low stride of 4 from shallow layers. The key point is the retrieval of the top-ranked regions with more likelihood of containing small objects from shallow layers of the network. This allows to dismiss the remaining part of the input image without affecting the final accuracy while keeping a reasonable computing time.

As another approach, MDFN [27] is a recent one-shot ConvNet that proposes only to exploit high-layers and, at the same time, improve the small and occluded object detection. This is done by introducing inception modules with multi-scale filters to enhance both the semantic and contextual information. Here, as in STDnet, it is shown that context is quite relevant for detecting small objects.

[1] http://cocodataset.org/#detection-leaderboard (Accessed: 2020-02-10).

Another promising research direction is based on boosting the scarce features of small objects using super-resolution (SR) techniques. On the one hand, this can be achieved by increasing the resolution of the whole input image [28], but it affects the computing time considerably. On the other hand, this can be handled by focusing only on the areas where there are small objects and applying there SR techniques. As an example, Noh et al. [29] propose a SR feature generator based on a GAN that learns to augment the features under the guidance of a SR feature discriminator.

Concerning the refinement of the final bounding box, there are solutions built on existing two-stage architectures that add additional headers to the existing one. These additional headers can be composed of the last convolution blocks [3,7] or of fully-connected layers [2,4,25]. Finally, a classifier assigns a category and a bounding-box regressor applies a final regression for each proposal. In this line, various studies have attempted to improve the quality of the final header. Gidaris and Komodakis [30] replicate the regressor stage to refine the bounding-box iteratively. In [31], they combine various classifiers trained with the integral loss. Similar ideas are exploited in Cai and Vasconcelos [32] to build Cascade R-CNN, improving two-stages detectors by applying consecutive headers trained with different proposals so that each one is fed by the previous.

In [33], to also address the inaccurate localization, they propose a hierarchical objectness network (HON) that refines the candidate proposals by what they call *stripe objectness*, which computes the in-out objectness and border objectness, instead of regressing the coordinates. With a similar purpose, Tao et al. [34] introduce a Focused Attention (FA) mechanism along with a class aware RPN (CARPN) which uses a new strategy for anchor generation that covers all scales but with fewer anchors to considerably reduce false positive proposals.

Video object detection has been widely studied for the last few years [16–18]. Several methods have been re-adapted from successful architectures in action detection [35–37]. Two-stream ConvNets are spatio-temporal networks that have achieved remarkable results [35]. The two-stream method has been studied by [37], where a Faster R-CNN has two RPNs operating over two streams of spatial and motion information from stacking optical flow over several frames.

Concerning video object detection, the solution addressed in Feichtenhofer et al. [16] builds on R-FCN [4] with a correlation operator inserted between two input frames to extract motion information of the objects across time. The correlation operates over the entire feature maps at different scales and estimates local feature similarity for various offsets between the two frames. Then, they link the detected objects into tubelets and reweight the detections' scores within them. Correlating whole feature maps implies that, as an object becomes smaller, their movement represents a considerably smaller influence, even though the correlation acts on several scales.

The approach in Kang et al. [19] performs video object detection in the current frame and tracks the objects through neighboring frames in order to modify their original detections for higher accuracy. The linking among detections in different frames is based on the mean optical flow vector within boxes. Similarly, the approach in Kang et al. [17] links objects into long tubelets using a tracking algorithm and then adopts a classifier to aggregate the detection scores in the tubelets.

Another alternative for video object detection introduced in Tang et al. [18] proposes a modified RPN called Cuboid Proposal Network (CPN) for detecting objects in multiple input frames. The cuboid proposals are regressed and classified to create short tubelets. Consecutive short tubelets are merged into long tubelets by a linking algorithm that takes the best detection for each overlapping frame between two tubelets.

In Flow-Guided Feature Aggregation (FGFA) [20], authors aggregate spatial features over time based on feature correspondences computed by optical flow to improve detections. Deng et al. present Relation Distillation Networks (RDN) [21], which aggregate and propagate object relation using the region proposals of current and neighboring frames to enhance the features of each object proposal, and thus capturing the core features of a given object across a video. In [22], authors introduce Memory Enhanced Global-local Aggregation (MEGA), a spatio-temporal ConvNet that relies on a novel Long Range Memory (LRM) module to efficiently aggregate global and local information from key frames. MEGA achieves state-of-the-art result (85.4% mAP) on ImageNet VID dataset [14].

The spatio-temporal ConvNet for video object detection we present in this paper, STDnet-ST, is built on our previous network, STDnet [7], which aimed at image object detection. STDnet-ST works on two consecutive frames. The retrieval of a fixed number of the top-ranked regions with more likelihood of containing small objects by the underlying STDnet eases the spatio-temporal procedure of STDnet-ST. In fact, as a difference with previous correlation-based solutions like [16], which runs correlation on the whole feature maps, STDnet-ST correlates pairs of regions with a high likelihood of having objects inside. This is a key point for small object detection, as the influence of the objects in the correlations calculated for the whole feature maps decreases with the size of the objects themselves —correlation values are mostly due to the background. Estimating the correlation for specific regions of the image allows to obtain correlation values influenced by the objects. This, in turn, permits to process only high quality tubelets by linking the objects inside such regions, which increases accuracy.

3. STDnet-ST architecture

STDnet-ST is a spatio-temporal convolutional neural network for the detection of small objects in video, i.e., objects smaller than 16×16 as defined in this paper. STDnet-ST has two components:

- The spatio-temporal convolutional neural network, which takes as inputs the current (f_t) and previous (f_{t-1}) frames, and returns the set of detections ($\mathcal{D}_t$), their confidences ($\mathcal{P}_t$), and the correlations (Section 3.1) among the detections at t and $t-1$ (C_t). These correlations will be used to associate the detections of both time instants.
- The STDnet-ST tubelet linking, which is based on the Viterbi algorithm, and includes the correlation-based tubelet linking and the tubelet suppression procedure.
  - The correlation-based tubelet linking (Section 3.2.2), that links the detections obtained at different time instants ($t = 1, \ldots, \tau$), generating the optimal tubelets along time for each of the objects. The final goal of tubelet linking is to update the scores of the detections at time τ using the previous $\tau - 1$ detections, according to the confidence of the whole tubelet (Section 3.2.2). A key element of the tubelet linking is the correlation provided by the spatio-temporal ConvNet, which evaluates the likelihood of the association of two detections. Also, the scores are updated taking into account the confidence variability of the tubelet, which indicates the confidence of the whole tubelet.
  - The tubelet suppression algorithm, that filters the tubelets obtained by the correlation-based tubelet linking, eliminating those that contain incorrect data associations (Section 3.2.3).

3.1. Spatio-temporal ConvNet

Fig. 2 shows the architecture of STDnet-ST, which consists of two sibling branches together with a correlation operation among

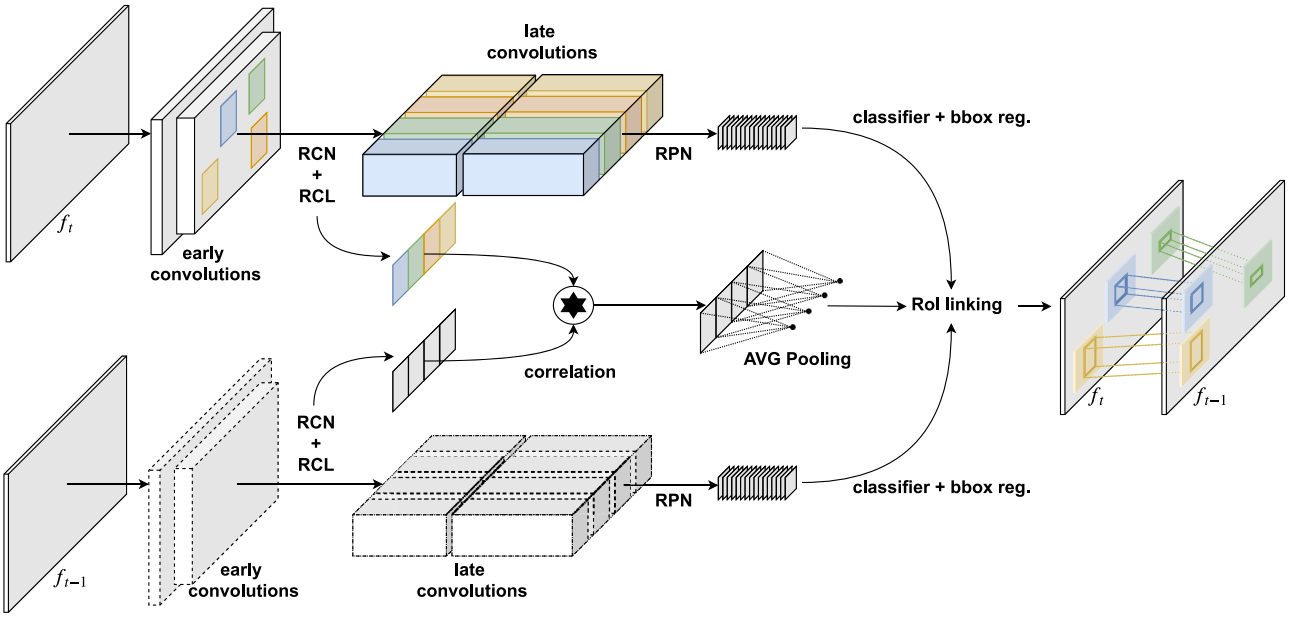

Fig. 2. STDnet-ST ConvNet architecture. Each branch performs RCN+RCL to obtain the most promising regions (RCN regions) that are further refined into detections by the RPN and a classifier. Simultaneously, the two sets of RCN regions feed a correlation module that associates the correlation values to the final detections.

selected regions. Each of the branches is based on the STDnet architecture [7], which is focused on the detection of small objects in images, i.e., it does not take into account temporal information.

The ability of STDnet to detect small objects is due to the high resolution of the deeper feature maps of the ConvNet. This high resolution of the last feature maps is possible because STDnet provides the most promising regions of the image in the early stages, thus focusing only on those regions that most likely contain small objects. The main components of STDnet are the following —for a more detailed description refer to [7]:

- Early convolutions. In the shallower convolutional layers, STDnet learns simple features from the objects of interest.
- Region Context Network (RCN). Just after the shallower convolutions, STDnet applies a novel detector of promising areas over the last feature map to select those regions that most likely contain small objects. Then, the m_t top scored regions $\mathcal{R}_t = \{r_t^1, \ldots, r_t^{m_t}\}$ are gathered in a single feature map by the RoI Collection Layer (RCL). There are two main differences between RCN+RCL and a typical RPN (Region Proposal Network): (i) RCN returns always regions of a fixed size that contain at least an object centering in it, while RPN returns bounding boxes of objects; (ii) RCL generates a new synthetic feature map, meaning that two neighboring pixels in the feature map that belong to two different regions are not neighboring pixels in the original image.
- In the late convolutions stage, the feature maps have a high resolution due to the memory saved by ruling out non promising areas. As the output of the RCL is a feature map with disjoint areas, all convolutions are designed to keep the features of each region separated from each other through padding.
- STDnet has a single RPN that takes as input the fourth convolutional block (*conv4*), which contains the most promising areas provided by the RCN but with richer semantic information.
- The last stage of STDnet refines the outputs of the RPN, generating the final bounding boxes and classifications of the objects.

STDnet-ST works with two consecutive video frames, t and $t-1$. Both branches —based on STDnet— share the same weights throughout the execution. Each of the branches generates a set of detections ($\mathcal{D}_t$) and their corresponding confidences ($\mathcal{P}_t$). As seen in Fig. 2, the two branches are connected through the correlation module. The correlation assesses the degree of matching between a pair of RCN regions at t and $t-1$, in order to link the final detections provided by the RPN thereafter. The hypothesis on which it relies is that each RCN region specializes in detecting a single object centered in it, allowing a straightforward extension over two detections. The correlation module consists of the two region composed feature maps generated by RCL for each branch, a correlation operator, an average pooling and a final RoI linking operation. The operation of the correlation module is as follows:

1. First, it calculates the correlation for each pair of RCN regions $< r_{t-1}^i, r_t^j >$, where $r_{t-1}^i \in R_{t-1}$, $r_t^j \in R_t$, $i = 1, \ldots, m_{t-1}$, and $j = 1, \ldots, m_t$. The output is a correlated feature map with the same width and height as the input regions, and where each pixel is obtained as the dot product of the pixels placed at that position in both regions —the depth of the correlated feature map is a single channel, due to the dot product. The correlation operator will produce a feature map with $m_{t-1} \times m_t$ regions, each one representing the correlation between two of the RCN regions.
2. Then, an average pooling is applied to summarize each of the regions of the correlated feature map in a single value associated to each pair of RCN regions, generating $m_{t-1} \times m_t$ correlation scores.
3. Finally, the correlation scores of each pair of RCN regions are associated to the final detections by the RoI linking operation. The RoI linking operation takes as input the final detections ($\mathcal{D}_t$) —generated by the RPN and further refined by the classifier— for each STDnet-ST branch, as well as the correlation scores, and outputs the correlation scores but associated to each pair of final detections, generating the matrix $\mathcal{C}_t$. $\mathcal{C}_t$ has a size of $n_{t-1} \times n_t$, where n_{t-1} and n_t are respectively the number of detections at times $t-1$ ($\mathcal{D}_{t-1}$) and t ($\mathcal{D}_t$). Those correlation scores not included in $\mathcal{C}_t$ —not all RCN regions have an associated final detection— are kept as they are involved in the tubelet suppression algorithm (Section 3.2.3).

3.2. STDnet-ST tubelet linking

The object linking across a sequence of frames to build tubelets is a popular approach to combine the temporal information. The

final goal of this stage is to increase the confidence of those detections that have a high likelihood of being true positives and to reduce the confidence of those detections with a low likelihood of being true positives.

First, we describe a baseline tubelet linking approach based on the spatial overlap between boxes in neighboring frames without considering the motion information. Then, we present the STDnet-ST tubelet linking with its two components: (i) the correlation-based tubelet linking (Section 3.2.2), which is based on the correlation scores generated by the ConvNet (Section 3.1); and (ii) the tubelet suppression procedure (Section 3.2.3) that removes those unlikely tubelets retrieved from the correlation-based tubelet linking.

3.2.1. Baseline tubelet linking

The baseline tubelet linking is based on [37], although they apply it to action detection in video, while we use it for spatio-temporal object detection. Given a set of τ frames, first, the tubelet linking calculates the set of scores between pairs of detections in two consecutive time instants ($\mathcal{S}_t$). Then, it applies the Viterbi algorithm [36] to find the most likely sequences, i.e., tubelets ($\mathcal{V}$), for all detections in the τ frames. Finally, it recalculates the score of each detection in τ ($\hat{\mathcal{P}}_\tau$) given the tubelet it belongs to.

The first step of tubelet linking calculates the score matrix $\mathcal{S}_t = \{s_t^{11}, \dots, s_t^{n_{t-1}n_t}\}$, where s_t^{ij} is the score between two equal category detections d_{t-1}^i and d_t^j in two consecutive frames, and is given by:

$$s_t^{ij} = p_{t-1}^i + p_t^j + \lambda \cdot \text{IoU}(d_{t-1}^i, d_t^j) \tag{1}$$

where p_t^j is the confidence returned by STDnet-ST for the jth detection at frame t, IoU is the overlap —measured as the intersection over union— between two detections, and λ is a parameter that balances the importance between the confidences returned by the ConvNet and the IoU. Thus, s_t^{ij} estimates the likelihood that the ith detection at frame $t-1$ and the jth detection at frame t are both true positive detections and come from the same object.

Next, the Viterbi algorithm is applied to obtain the most probable sequences of detections. This algorithm maximizes the conditional probability of the tubelets —each one represents an object seen at different time instants— given a set of detections $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_\tau\}$ and their corresponding scores $\mathcal{S} = \{\mathcal{S}_2, \dots, \mathcal{S}_\tau\}$ over time of the same category. Given the whole set of possible tubelets $\mathcal{V}$, the tubelet with the highest likelihood is:

$$\hat{v} = \arg\max_{v \in \mathcal{V}} \sum_{t=2}^{\tau} s_t^{i(v)j(v)} \tag{2}$$

where $i(v)$ and $j(v)$ are the detections at times $t-1$ and t for a given tubelet $v \in \mathcal{V}$.

Once the optimal tubelet $\hat{v}$ is found, those detections within $\hat{v}$ are removed from $\mathcal{D}$ and $\mathcal{S}$, and the process (Eq. (2)) is repeated iteratively to obtain the set of optimal tubelets $\hat{\mathcal{V}}$. Finally, the new confidences for the detections of the last frame τ within each tubelet $\hat{v}$ are updated as:

$$p_\tau^{i(\hat{v})} = \frac{1}{\tau} \sum_{t=1}^{\tau} p_t^{i(\hat{v})} \tag{3}$$

where $p_t^{i(\hat{v})}$ is the confidence of the detection at time t belonging to tubelet $\hat{v}$. Thus, the confidence of the detections at the last frame are updated with the average confidences of their corresponding tubelets. In this way, tubelet linking increases the confidences of those detections with a low confidence in the last frame, but with a strong track record in previous time instants, which indicates that the detection is a true positive. This process also helps to reduce the confidence of the detections that belong to a tubelet with a weak track record, as this is often supposed to be a false positive. The tubelet linking algorithm is repeated for each category of the dataset.

3.2.2. Correlation-based tubelet linking

Associating the detections in two consecutive frames through IoU might work fine in some scenarios but, in general, it is a very weak feature for object linking. Some scenarios where IoU might generate wrong associations are: small objects that barely overlap between consecutive frames, fast motions of the object and/or the camera, many objects with partial overlaps among them, and videos with a low frame rate or skipping frames.

The proposed correlation-based tubelet linking addresses the preceding points by introducing the correlation score as the feature for data association. In this way, STDnet-ST can associate small objects regardless of their mutual distance in consecutive frames. Also, it is possible to avoid the association of objects with very different features, but placed in the same position in consecutive frames.

Correlation-based tubelet linking modifies Eq. (1) by replacing the spatial overlap (IoU) with the correlation score to compute the score matrix $\mathcal{S}_t$. Each element of $\mathcal{S}_t$ is calculated as:

$$s_t^{ij} = p_{t-1}^i + p_t^j + \lambda \cdot c_t^{ij} \tag{4}$$

where c_t^{ij} is the correlation obtained by the STDnet-ST ConvNet for the ith detection at time $t-1$ and the jth detection at time t, defined as:

$$c_t^{ij} = \rho(r_{t-1}^{k(i)}, r_t^{l(j)}) \tag{5}$$

where ρ represents the correlation module function, and $r_{t-1}^{k(i)}$ and $r_t^{l(j)}$ are the RCN regions at $t-1$ and t associated to detections d_{t-1}^i and d_t^j. So that, $l(j)$ and $k(i)$ are the RoI linking outputs that associate each RCN region r_{t-1}^k and r_t^l with their corresponding detections d_{t-1}^i and d_t^j.

The second novelty is the modification of Eq. (3) as follows:

$$p_\tau^{i(\hat{v})} = \begin{cases} \max_{t=1}^{\tau} p_t^{i(\hat{v})} & \text{if } \sigma(\{p_t^{i(\hat{v})}\}_{t=1}^{\tau}) \leq \kappa \\ \frac{1}{\tau} \sum_{t=1}^{\tau} p_t^{i(\hat{v})} & \text{otherwise} \end{cases} \tag{6}$$

where σ is the standard deviation of the confidences of the tubelet $\hat{v}$, and κ is a threshold. Our hypothesis is that when the confidence variability in a tubelet is small, the last detection might be a true positive and the confidence of that detection can be updated to the maximum confidence of the tubelet. On the other hand, when the variability is high, the confidence is updated with the average confidence, like in the baseline tubelet linking, as the likelihood of being a true positive is lower.

3.2.3. Tubelet suppression procedure

The main downside of the original Viterbi algorithm is that it generates all possible tubelets $\hat{\mathcal{V}}$, even though they are unlikely given their scores. A typical example is a tubelet created with false and true positive detections, only because there is no other possible data association. This causes a decrease in the global accuracy, as discussed in Section 4. STDnet-ST tubelet linking manages this situation by defining a tubelet suppression algorithm based on adding dummy detection nodes. Thus, the Viterbi algorithm might build a tubelet using one or more dummy nodes, and these tubelets will be later deleted.

These dummy nodes can be generated owing to the two-level detection —i.e., RCN regions and final detections—, which provides a higher level of abstraction from the RCN regions that do not generate a final detection, but whose correlation score is useful. The tubelet suppression algorithm generates dummy nodes so that: (i) false positives at t are associated to a dummy node rather than to a true positive at $t-1$ or, (ii) true positives at t are associated

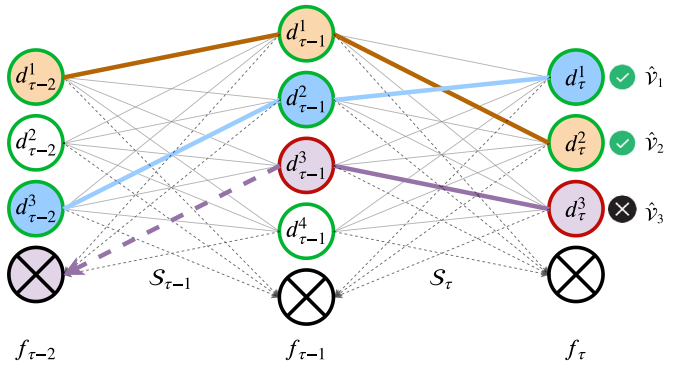

Fig. 3. An example of the optimal solution provided by the Viterbi algorithm with the tubelet suppression procedure. Nodes with a green border correspond to true positive detections, and those in red with false positive detections. The solution produces a valid tubelet for d^1_τ and d^2_τ (orange and blue) and a non-valid tubelet for d^3_τ (purple). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to a dummy node rather than to a false positive at $t-1$. The first case happens when the dummy node has a high correlation with a false positive, e.g., both RCN regions have a similar background. The second case happens when the dummy node has a high correlation with a true positive, e.g., when the RCN region includes an object that was not finally detected. Hence, the gain in the first case is given by the fact that the false positive at t is not associated to true positives and, thus the false positive confidence is not increased. The gain in the second case is given by the fact that the true positive at t is not associated to false positives that decrease its confidence.

Fig. 3 shows an example of how the Viterbi algorithm works with tubelet suppression. Each node represents a detection d^i_t or a dummy node ($\otimes$). Detections of the same frame are in the same column. Those nodes at different time instants filled with the same color represent the generated tubelets. Solid lines represent the correlation scores (c^{ij}_t) between pairs of detections (Eq. (5)), and dashed lines represent conections between detections and dummy nodes. The tubelet suppression procedure will remove the optimal tubelet $\hat{\mathcal{V}}_3$ as it links the false positive $d^3_{\tau-1}$ with the dummy node in $f_{\tau-2}$ due to the existence of an RCN region $r^i_{\tau-2}$ with a higher correlation score than any other detection in $\tau-2$. Ideally, this indicates that there is a false positive that is detected by the ConvNet at some frames ($\tau-1$ and τ), and filtered out in others ($\tau-2$). If the tubelet linking process does not consider tubelet suppression, the Viterbi algorithm would generate a tubelet including $d^2_{\tau-2}$, $d^3_{\tau-1}$ and d^3_τ and, therefore, would probably increase the confidence of d^3_τ, which is a false positive.

Algorithm 1 shows the STDnet-ST tubelet linking algorithm, including the correlation-based tubelet linking and the tubelet suppression procedure. Given the set of detections ($\mathcal{D}$) from time $t=1$ to $t=\tau$, their confidences ($\mathcal{P}$), and the set of score matrices ($\mathcal{S}$) $-s^{ij}_t$ (Eq. (4)) is the ij element of matrix $\mathcal{S}_t$, calculated from the ith detection at time $t-1$ and the jth detection at time $t-$, the algorithm returns the updated confidences ($\hat{\mathcal{P}}_\tau$) associated to the detections at time τ. First, we initialize $\hat{\mathcal{P}}_\tau$ with the confidences generated by the ConvNet (line 1). Then, we add a dummy node (line 3) to the detection set at time t —with original size n_t— as well as one column (line 5) and one row (line 6) to the score matrix at time t —with original size $n_{t-1}\times n_t$. In the added column we store the scores between a dummy node and all detections at $t-1$, while in the added row are the scores between a dummy node and the detections at t.

The scores associated with dummy nodes are based on Eqs. (4) and (5), but where one of the two RCN regions involved

Algorithm 1: STDnet-ST tubelet linking.

Input : $\mathcal{D}=\{\mathcal{D}_t=\{d^1_t,\ldots,d^{n_t}_t\}\mid t=1,\ldots,\tau\}$
Input : $\mathcal{P}=\{\mathcal{P}_t=\{p^1_t,\ldots,p^{n_t}_t\}\mid t=1,\ldots,\tau\}$
Input : $\mathcal{S}=\{\mathcal{S}_t=\{s^{11}_t,\ldots,s^{n_{t-1}n_t}_t\}\mid t=1,\ldots,\tau\}$
Output: $\hat{\mathcal{P}}_\tau$

1 $\hat{\mathcal{P}}_\tau \leftarrow \mathcal{P}_\tau$
2 **for** $t=1,\ldots,\tau$ **do**
3 $\quad\mathcal{D}_t \leftarrow \mathcal{D}_t \cup d^\varnothing_t$
4 $\quad$**if** $t>1$ **then**
5 $\quad\quad\mathcal{S}_t \leftarrow \mathcal{S}_t : d^{\varnothing\star}_{t-1}$
6 $\quad\quad\mathcal{S}_t \leftarrow \mathcal{S}_t : d^{\star\varnothing}_t$
7 $\quad\quad$**for** $i=1,\ldots,n_{t-1}$ **do**
8 $\quad\quad\quad c^{i,n_t+1}_t = \max_k \rho(r^{j(i)}_{t-1}, r^k_t) \mid r^k_t \not\to d^l_t \;\forall l=1,\ldots,n_t$
9 $\quad\quad\quad s^{i,n_t+1}_t = p^i_{t-1} + p^i_{t-1} + \lambda\cdot c^{i,n_t+1}_t$
10 $\quad\quad$**for** $i=1,\ldots,n_t$ **do**
11 $\quad\quad\quad c^{n_{t-1}+1,i}_t = \max_k \rho(r^k_{t-1}, r^{j(i)}_t) \mid r^k_{t-1} \not\to d^l_{t-1} \;\forall l=1,\ldots,n_{t-1}$
12 $\quad\quad\quad s^{n_{t-1}+1,i}_t = p^i_t + p^i_t + \lambda\cdot c^{n_t+1,i}_t$
13 **while** $\{d^1_t \neq d^\varnothing_t \;\forall t=1,\ldots,\tau\}$ **do**
14 $\quad\hat{v} \leftarrow$ `Viterbi`$(\mathcal{D},\mathcal{S})$
15 $\quad$`isvalid` $\leftarrow$ *True*
16 $\quad$**for** $t=1,\ldots,\tau$ **do**
17 $\quad\quad$**if** $d^{i(\hat{v})}_t \neq d^\varnothing_t$ **then**
18 $\quad\quad\quad\mathcal{D}_t \leftarrow \mathcal{D}_t \setminus d^{i(\hat{v})}_t$
19 $\quad\quad\quad$**if** $t>1$ **then**
20 $\quad\quad\quad\quad\mathcal{S}_t \leftarrow$ `deleteColumn`$(\mathcal{S}_t, i(\hat{v}))$
21 $\quad\quad\quad$**if** $t<\tau$ **then**
22 $\quad\quad\quad\quad\mathcal{S}_{t+1} \leftarrow$ `deleteRow`$(\mathcal{S}_{t+1}, i(\hat{v}))$
23 $\quad\quad$**else**
24 $\quad\quad\quad$`isvalid` $\leftarrow$ *False*
25 $\quad$**if** `isvalid` **then**
26 $\quad\quad\hat{p}^{i(\hat{v})}_\tau =$ `updateConfidence`$(\mathcal{P},\hat{v})$ [Eq. 6]
27 **done**

in Eq. (5) is a free RCN region —RCN region without detection— (lines 8 and 11). The free RCN regions are those that have been discarded by the ConvNet because the likelihood of containing an object is low. In particular, for each of the detections at $t-1$, the free RCN region from t that will be selected to compute the correlation score is the one with the maximum correlation score. The same for the detections at t and the free RCN regions from $t-1$. So that, the correlation (line 8) for a given detection d^i_{t-1} within an RCN region $r^{j(i)}_{t-1}$ is the maximum correlation between $r^{j(i)}_{t-1}$ and the whole set of free RCN regions at t (r^k_t). Then, new scores (lines 9 and 12) are calculated as in Eq. (4), where p^i_{t-1} and p^j_t both come from the real detection d^i_{t-1}, i.e, $p^i_{t-1}=p^j_t$.

Next, the Viterbi algorithm is applied with the set of detections and the new set of score matrices (line 14), while every $\mathcal{D}_t$, from $t=1$ to $t=\tau$, still has detections provided by the STDnet-ST ConvNet —not just dummy nodes— (line 14). Then, for each generated tubelet by the Viterbi algorithm, the corresponding detections are deleted from the set of detections (line 18), and the corresponding row and column is also deleted from the score matrices (lines 20 and 22). A detection at time t contributes to the score matrices $\mathcal{S}_t$ and $\mathcal{S}_{t+1}$ —Fig. 3. Finally, if the tubelet is valid, i.e., it does not contain dummy nodes, the confidences of the detections at time τ

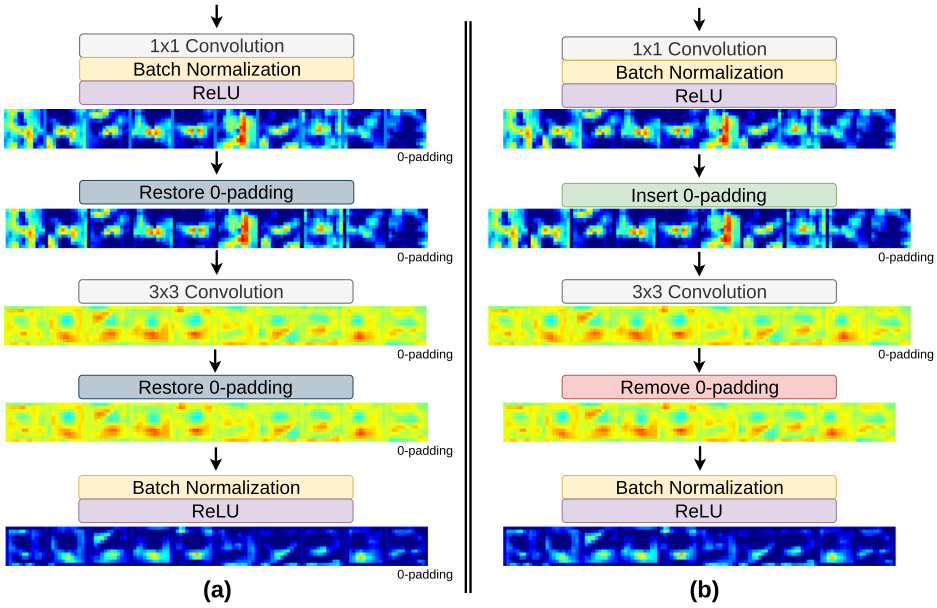

Fig. 4. Structure of a ResNet block for STDnet: (a) using the original 0-padding implementation and (b) with the 0-padding implementation proposed in this paper.

that are in the set of tubelets are updated following Eq. (6) (line 26).

3.3. Spatial STDnet enhancement

This section describes the improvements made over the original version of STDnet [7]: a new 0-padding operation between regions, and the replacement of the classical header with a cascaded header.

3.3.1. Rethinking the 0-padding operation

The first improvement concerns the structure of the convolution blocks after obtaining the promising regions by the Region Context Network (RCN) and the RoI Collection Layer (RCL). In the original STDnet, the RCL encompasses the different regions proposed by the RCN and adds a 0-padding between them so that the convolution kernels larger than 1×1 do not share information from adjacent regions.

Fig. 4 (a) shows how the 0-padding was restored before and after each convolution. However, although the convolution operations were not affected with this structure, 0-padding restoration did have an effect on batch normalization and harmed training. For instance, if there are 50 RCN regions with a size of 8×8 and 1 px 0-padding between every pair, the overall size of each channel will be $449 \times 8 = 3592$ px,[2] where $49 \times 8 = 392$ px are 0's. Thus, all those 0's —which are more than the 10% of the pixels in the feature map— were influencing the learning of the network. In addition, 1×1 convolution operations had to perform unnecessary operations on that 0-padding.

To solve this problem, a built-in operator has been implemented that inserts and removes the 0-padding before and after each convolution greater than 1×1. The new structure is represented in Fig. 4(b).

3.3.2. Cascaded header

The original STDnet header [7] (classifier + bounding box regression) has been replaced by three consecutive headers that interatively improve small objects detection. This implementation is based on the research carried out by Cai and Vasconcelos [32], where several twin headers are trained with progressively more restrictive overlap thresholds.

The idea is to improve in successive headers object detections that have a minimum overlap with the true object until reaching the threshold defined to be considered true positive. Therefore, this will improve not only the final accuracy, but also the final recall (Fig. 5). A common problem in small object detection are double detections: for large objects, non maximum suppression eliminates those double detections but, for small objects, the overlap is very small. The cascaded header will help to eliminate these false positives, as bounding boxes will be more accurate.

Differently from the implementation in Cai and Vasconcelos [32], where the cascade is applied directly on the feature map prior to RPN, the additional headers that make up the cascade approach in STDnet-ST take the information from the same feature map, but with disjoint regions. The cascaded headers have to compute the target bounding box using the predecessor header, and have to retrieve the spatial information of the regions relative to the input image from the Region Context Network (RCN).

4. Experiments

4.1. Evaluation metrics

We assess the performance of our approach and previous work with the metrics reported in MS COCO [6]. Such metrics are the Average Precision ($AP^{@.5}$), which gives the average precision of those objects detected with at least 50% IoU between the detected and the ground-truth bounding boxes, and $AP^{@[0.5,0.95]}$, which is the average AP when the IoU goes from 50% to 95% in 5% steps. In the default COCO metrics, the results are shown for three different subsets: *small* (AP_s), objects smaller than 1024 pixels area; *medium* (AP_m), objects between 1024 and 9216 pixels area, and *large* (AP_l), objects larger than 9216 pixels area. In this paper we define a new scale subset following COCO style, *very small* (AP_{xs}), to include small targets as defined in this paper, i.e., those enclosed in bounding boxes with less or equal than 256 pixels area. The *XS* subset is defined in order to evaluate the performance for very small objects.

[2] 449×8 is the shape of a feature map channel composed of 50 regions arranged horizontally with 1px padding between each pair: width = (50×8) + (1×49); height = 8.

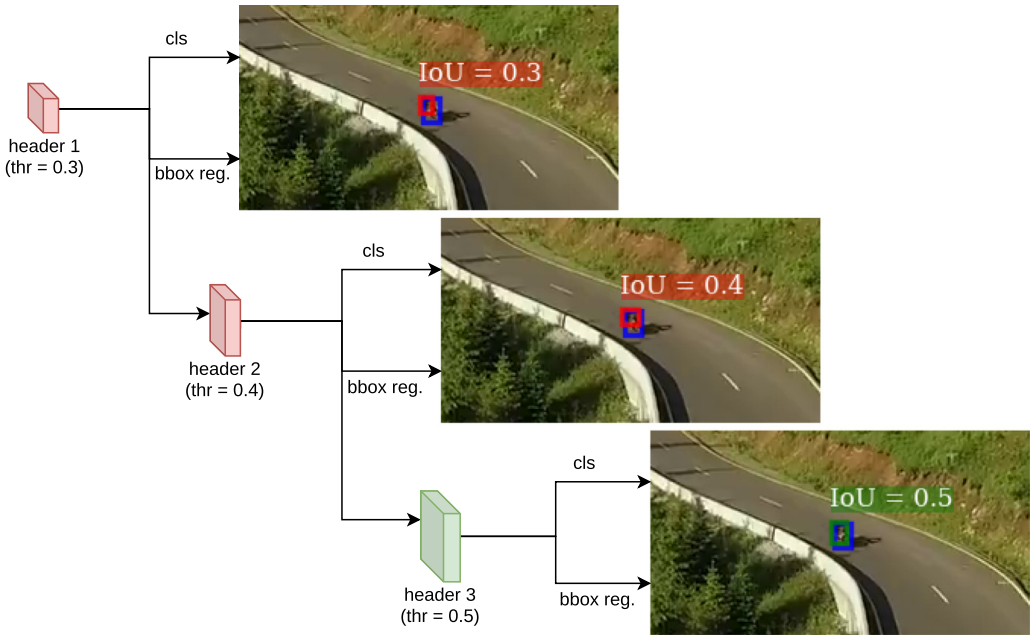

Fig. 5. Performance of the cascaded header over a rough proposal towards a true positive detection.

4.2. Datasets

We conduct extensive experiments on three publicly accessible datasets: USC-GRAD-STDdb [7], UAVDT [12] and VisDrone2019-VID [13].

- **USC-GRAD-STDdb** [7]. It comprises 115 video segments with more than 25,000 annotated frames. The resolution of the video is HD 720p (1280 × 720). There are more than 56,000 objects, with most of them ranging from 16 ($\approx 4 \times 4$) up to 256 ($\approx 16 \times 16$) as pixel area, i.e., small objects as defined in this paper. The videos in USC-GRAD-STDdb comprise three main landscapes —air, sea and land— with five object categories, namely: air (drone, bird), 57 videos with 12,139 frames; sea (boat), 28 videos with 7099 frames; and land (vehicle, person), 30 videos with 6619 frames. Nevertheless, the evaluation will be carried out as a single category. The test subset holds 11,337 objects, where almost 90% of them (10,136 objects) correspond to the *very small* subset.
- **UAVDT** [12]. It contains 23,829 frames of training data and 16,580 images of test data of $\approx$ 1024 × 540 resolution. The videos are recorded with an UAV platform over different urban areas. The ground truth targets are vehicles labeled as car, bus and truck, but evaluated as a single category. UAVDT comprises a total of 375,884 test objects, where 76,215 are considered within the *very small* subset (20.3%).
- **VisDrone2019-VID** [13]. The VisDrone2019-VID challenge provides a total of 96 HD/Full HD video sequences, including 56 sequences for training (24,201 frames in total), 7 sequences for validation (2819 frames in total) and 17 sequences for development testing (test-dev) (6635 frames in total). There is also a blind test (test-challenge) subset that comprises 16 videos, but the evaluation system does not report the metrics for the *extra small* subset, so these data will be dismissed and the results will be reported only for the test-dev subset. The dataset is mainly focused on people and vehicles, where ten diffent categories of interest are labeled: pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle. The 17 sequences for testing hold 310,228 test objects, where 27,027 (8.7%) are considered *extra small*.

4.3. Implementation details

We implemented STDnet-ST based on STDnet [7]. Faster R-CNN [2] with Feature Pyramid Network (FPN) [25] is adopted as the baseline detection network for small object detection. We have also compared our proposal with the state-of-the-art spatio-temporal approaches: FGFA [20], RDN [21] and MEGA [22],[3] with the anchors' size best suited for small objects as defined in Bosquet et al. [7]. These three approaches achieve state-of-the-art results in ImageNet VID dataset [14].

Training phase The input size of STDnet-ST is determined by the resolution of the video under study, namely, 1280 × 720 pixels in USC-GRAD-STDdb, 1024 × 540 in UAVDT and 1920 × 1080 in VisDrone2019-VID. For USC-GRAD-STDdb, as most of the objects belong to the *XS* size, i.e., below 256 pixels area, RCN regions of size 32 × 32 suffice to enclose all objects. In these cases the STDnet-ST training phase is continuous during 40k iterations with two step decay. For UAVDT and VisDrone2019-VID, with objects with more varying sizes, including those larger than the *XS* category, i.e., below 256 pixels area, the training process requires pre-training. Thus, first, we run a pre-training phase with Faster R-CNN during 20k iterations to address all object sizes followed by a fine-tuning with STDnet-ST for other 20k iterations with two step decay. In order to retrieve all objects with more diverse aspect ratios, we set the RCN region size to 48 × 48 pixels. Also, as reported in Bosquet et al. [7], for both datasets, RCN between *conv3* and *conv4* and the initialization of anchors by k-means lead to the best performance metrics. Finally, when training the model, we set the base learning rate to 0.0025, a momentum of 0.9, and parameter decay of 0.0001 on weights and biases.

Test phase The input size and the RCN region size are the same as those of the training phase. The maximum number of RCN regions is set to 100. The spatio-temporal hyperparameters τ, κ and λ are set to 4, 0.02 and 1.0, respectively, derived by experimental studies over a validation subset from the USC-GRAD-STDdb training set. We also apply a box-voting scheme after non-maximum suppression [30].

[3] https://github.com/Scalsol/mega.pytorch.

Table 1
Ablation study on USC-GRAD-STDdb for the different tubelet linking components of STDnet-ST++. Results without correlation features are implemented directly over one branch of STDnet-ST++ —i.e., the first three rows—, and those that use them represent the different versions of STDnet-ST++ —i.e., the last four rows. The first row refers to STDnet as it is defined in Bosquet et al. [7] with the enhancements proposed in Section 3.3, i.e., STDnet++.

Baseline linking	Confidence variability	Correlation linking	Tubelet suppression	$AP_{xs}^{@[0.5,0.95]}$	$AP_{xs}^{@.5}$
–				18.9	59.1
✓				20.1	61.4
✓	✓			20.3	61.8
		✓		20.4	61.6
	✓	✓		20.6	62.0
		✓	✓	20.9	62.6
	✓	✓	✓	**21.4**	**63.4**

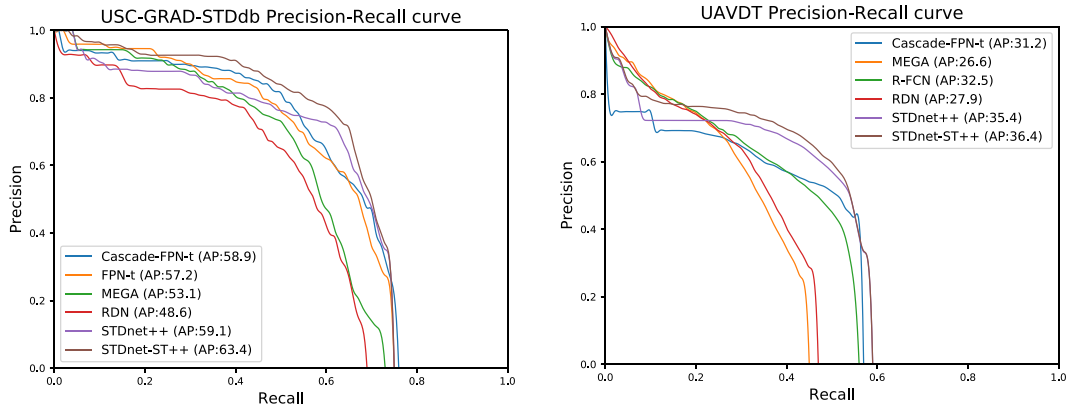

Fig. 6. Precision-Recall curves and $AP_{xs}^{@.5}$ of the most relevant approaches in Table 2 for USC-GRAD-STDdb (left), and Table 3 for UAVDT (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Evaluation metrics for different methods on USC-GRAD-STDdb database. -t indicates the use of baseline tubelet linking and confidence variability to compute the final score of each tubelet.

Method	$AP_{xs}^{@[0.5,0.95]}$	$AP_{xs}^{@.5}$
FGFA [20]	11.7	37.5
RDN [21]	15.5	48.6
MEGA [22]	17.4	53.1
FPN [25]	17.3	54.5
Cascade-FPN [32]	17.4	55.9
FPN-t	18.7	57.2
Cascade-FPN-t	19.1	58.9
STDnet [7]	18.3	57.8
STDnet++	18.9	59.1
STDnet-ST	20.1	62.1
STDnet-ST++	**21.4**	**63.4**

In addition, there are some differences between the original STDnet [7] and the STDnet used as base of STDnet-ST for this paper: (1) STDnet-ST has been implemented in Caffe2; (2) the RoI pooling dimension is reduced from 7×7 to 4×4 when the cascaded header is applied to keep constant the computational cost; (3) the last ResNet convolutional block (*conv5*) is replicated to fine-tune each header during the training phase; and (4) all possible positive RCN regions are passed through the network instead of limiting their number, in order to perform well in datasets with many objects of interest per image.

4.4. Results on USC-GRAD-STDdb

Tables 1 and 2 show experimental results on USC-GRAD-STDdb [7]. Our approach is compared to the state-of-the-art FPN [25], as it proved to be the most competitive method for the present dataset [7], and FPN with cascaded header (Cascade-FPN [32]), for fair comparisons. From here on, the names of the different versions of STDnet are as follows: STDnet refers to the original ConvNet; STDnet++, refers to STDnet with the enhancements detailed in Section 3.3; STDnet-ST and STDnet-ST++ refer to the spatio-temporal architectures defined in Sections 3.1 and 3.2 adopting STDnet and STDnet++ as base network, respectively. Finally, as the baseline tubelet linking and the confidence variability methods are independent of the architecture, we have also tested the performance of FPN and Cascade-FPN with these components —referred as FPN-t and Cascade-FPN-t.

Table 1 studies the influence of the different components defined in this paper to exploit the temporal information from a video dataset. *Baseline linking* refers to the baseline method to generate tubelets defined in Section 3.2.1; *Confidence variability* refers to the modification of the confidences of the detections based on the confidences of the tubelets due to their variability, as addressed in Eq. (6); *Correlation linking* means the correlation-based tubelet linking as addressed in Section 3.2.2; and *Tubelet suppression* concerns the tubelet suppression procedure presented in Section 3.2.3.

As it can be observed, the use of temporal information leads to higher performance. STDnet-ST++ outperforms STDnet++ from 18.9% to 21.4% for $AP_{xs}^{@[0.5,0.95]}$ and from 59.1% to 63.4% for $AP_{xs}^{@.5}$. In this ablation study, it is also possible to determine the contribution of each of the components to the performance of STDnet. The correlation-based linking, together with the confidence variability contribute to increase 0.5% $AP_{xs}^{@[0.5,0.95]}$ and 0.6% $AP_{xs}^{@.5}$ —Table 1, rows 2 and 5. Also, the tubelet suppression procedure adds a gain of 0.8% $AP_{xs}^{@[0.5,0.95]}$ and 1.4% $AP_{xs}^{@.5}$ over the previous result —Table 1, rows 5 and 7.

Two conclusions can be drawn from Table 1. First, the importance of the correlation obtained by the ConvNet of STDnet-ST. The correlation-based tubelet linking is capable of improving the IoU-based baseline tubelet linking by comparing the early features of the objects and their context; and more importantly, it allows to build the tubelet suppression procedure with a higher level of abstraction that cannot be found by associating detections

Fig. 7. Some object detection results of STDnet-ST++ for USC-GRAD-STDdb (top), UAVDT (middle) and VisDrone2019-VID (bottom) test sets. A confidence threshold of 0.6 was used to display these images. For each image, green boxes are true positives, red boxes false positives and blue boxes false negatives. The yellow rectangles are ignored regions. Only objects that belong to the *XS* size are displayed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

by spatial overlap. Second, the importance of the confidence variability when combined with the tubelet suppression procedure, as some of the tubelets that were composed by false negatives are discarded and, therefore, the confidence variability is more reliable.

Table 2 provides a comparison in terms of accuracy between the state-of-the-art FGFA, RDN, MEGA, FPN, Cascade-FPN, and our architectures STDnet-ST and STDnet-ST++. STDnet-ST++ outperforms FPN by 4.1% $AP_{xs}^{@[0.5,0.95]}$ and 8.9% $AP_{xs}^{@.5}$ and Cascade-FPN by 4.0% $AP_{xs}^{@[0.5,0.95]}$ and 7.5% $AP_{xs}^{@.5}$. FPN with spatio-temporal information improves its baseline by 1.4% $AP_{xs}^{@[0.5,0.95]}$ and 2.7% $AP_{xs}^{@.5}$. Even so, the results of the spatio-temporal FPN and Cascade-FPN remain below STDnet-ST and STDnet-ST++. When compared to the spatio-temporal approaches FGFA, RDN and MEGA, STDnet-ST++ outperforms them by at least 4.0% $AP_{xs}^{@[0.5,0.95]}$ and 10.3% $AP_{xs}^{@.5}$. This is mainly due to the fact that both the RPN placed in deep layers and the association methods used in these approaches have a high performance on large objects but a lower impact when dealing with extremely small objects. With respect to the original version of STDnet, STDnet-ST++ improves the result by a 3.1% $AP_{xs}^{@[0.5,0.95]}$ and a 5.6% $AP_{xs}^{@.5}$. The most relevant results are shown in Fig. 6(left) using Precision-Recall curves.

4.5. Results on UAVDT

The experimental results on the UAVDT dataset [12] are shown in Table 3. The first four rows are computed using the bounding box results provided in Du et al. [12], and directly adapted to the MS COCO results format [6].

Results confirm that the spatial STDnet performs better than the rest of the state-of-the-art spatial approaches. Moreover, it can be seen how the enhancements introduced for STDnet (STDnet++) improve $AP_{xs}^{@[0.5,0.95]}$ by 0.6% and $AP_{xs}^{@.5}$ by 2.9% higher than any other spatial approach. $AP^{@[0.5,0.95]}$ is considered the primary challenge metric by MS COCO [6], because it encompasses AP adding information on how it behaves as the IoU reaches perfection. It is also noteworthy that STDnet outperforms FPN-t and Cascade-FPN-t, which exploit spatio-temporal information.

As expected, our spatio-temporal proposal, STDnet-ST++, accomplishes better performance with respect to its spatial version, achieving state-of-the-art results in the UAVDT dataset for the *very small* subset. STDnet-ST++ overcomes spatio-temporal Cascade-FPN (Cascade-FPN-t) by 1.0% $AP_{xs}^{@[0.5,0.95]}$ and 5.2% $AP_{xs}^{@.5}$, and also R-FCN by 4.1% $AP_{xs}^{@[0.5,0.95]}$ and 3.9% $AP_{xs}^{@.5}$. Fig. 6(right) shows the Precision-Recall curves.

Table 3
Evaluation metrics on the *very small* subset of UAVDT dataset, i.e., objects under 16×16 pixels.

Method	$AP_{xs}^{@[0.5,0.95]}$	$AP_{xs}^{@.5}$
Faster R-CNN [12]	6.6	26.0
R-FCN [12]	9.2	32.5
RON [12]	3.7	19.7
SSD [12]	6.0	23.5
FGFA [20]	6.3	20.7
RDN [21]	9.3	27.9
MEGA [22]	9.2	26.6
FPN [25]	11.8	29.7
FPN-t	12.0	30.3
Cascade-FPN [32]	12.0	30.5
Cascade-FPN-t	12.3	31.2
STDnet [7]	12.5	35.1
STDnet++	12.6	35.4
STDnet-ST	13.1	36.0
STDnet-ST++	**13.3**	**36.4**

Table 4
Evaluation metrics on the *very small* subset of VisDrone2019-VID dataset, i.e., objects under 16×16 pixels.

Method	$AP_{xs}^{@[0.5,0.95]}$	$AP_{xs}^{@.5}$
FGFA [20]	3.8	16.8
RDN [21]	4.7	20.7
MEGA [22]	4.8	21.0
FPN [25]	6.2	19.9
Cascade-FPN [32]	6.1	20.2
FPN-t	6.3	20.2
Cascade-FPN-t	6.2	20.4
STDnet [7]	7.2	21.4
STDnet++	7.3	22.0
STDnet-ST	7.5	21.9
STDnet-ST++	**7.5**	**22.4**

4.6. Results on VisDrone2019-VID

The experimental results on the Visdrone2019-VID dataset [13] are shown in Table 4. In first place, it is confirmed that the STDnet based approaches outperform their counterparts. In second place, STDnet++ improves STDnet by 0.1% $AP_{xs}^{@[0.5,0.95]}$ and by 0.6% $AP_{xs}^{@.5}$. Finally, regarding the spatio-temporal approaches, STDnet-ST++ boosts 0.2% $AP_{xs}^{@[0.5,0.95]}$ and 0.4% $AP_{xs}^{@.5}$ its baseline, while improving 1.2% $AP_{xs}^{@[0.5,0.95]}$ and 2.0% $AP_{xs}^{@.5}$ compared to the best FPN-based approach. Examples of detections with STDnet-ST++ on the three datasets reported in this paper are shown in Fig. 7.

5. Conclusion and future work

We have introduced STDnet-ST, a spatio-temporal ConvNet to detect small targets in video. STDnet-ST is composed of two branches, and it binds the detections of two input frames by a correlation module to create spatio-temporal small object tubelets. Those tubelets are refined at the tubelet linking stage, which applies the Viterbi algorithm to the detections based on correlation linking, and implements a tubelet suppression procedure that allows STDnet-ST to dismiss unprofitable tubelets while preserving only high quality ones.

Furthermore, certain components of the STDnet structure [7] have been reformulated, leading to the definition of STDnet++ and STDnet-ST++. Enhancements have been made to 0-padding operation, for improving the network learning, and a cascaded header, to obtain better performance by turning false positives with low overlap into true positives.

In order to validate the proposed architecture, we have conducted experiments over three publicly available datasets with a large number of small objects: USC-GRAD-STDdb [7], UAVDT [12] and VisDrone2019-VID [13]. STDnet-ST++ achieves state-of-the-art results in all these datasets for very small objects, clearly outperforming its counterparts by 2.3% $AP_{xs}^{@[0.5,0.95]}$ on USC-GRAD-STDdb, by 1.0% $AP_{xs}^{@[0.5,0.95]}$ on UAVDT and by 1.2% $AP_{xs}^{@[0.5,0.95]}$ on VisDrone2019-VID.

Results show how the three main characteristics of STDnet-ST are key to achieve small object detection: (i) the use of high resolution feature maps throughout the architecture allows to locate the objects and adjust their bounding boxes; (ii) performing correlation over RCN regions allows to correctly associate objects in two consecutive frames, therefore, improving the detection precision; (iii) the correlation-based tubelet linking together with tubelet suppression procedure provide high quality tubelets to increase the final accuracy. The tubelet suppression procedure is possible due to the RCN regions, that provide a limited number of areas without objects where to look for possible correlations with false positive detections, therefore avoiding their linking with true positive detections.

As future work, we plan to address the limited number of small objects present in current datasets. Considering that manual object annotation is extremely time-consuming and that tracking-based annotation is far from being perfect, we will work on the definition of a pipeline to generate synthetic small objects from larger ones. Specifically, the recent advances in Generative Adversarial Networks (GANs) seem to be a promising route both for the generation of synthetic objects close to real ones and for suitable placement in different contexts. Super-resolution GANs are attractive in the former task, and inpainting and blending GANs for the latter one.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was partially funded by the Spanish Ministry of Science, Innovation and Universities 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] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[2] 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, pp. 91–99.
[3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[4] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, in: Advances in Neural Information Processing Systems (NIPS), 2016, pp. 379–387.
[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.
[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in: European Conference on Computer Vision (ECCV), 2014, pp. 740–755.
[7] B. Bosquet, M. Mucientes, V.M. Brea, STDnet: exploiting high resolution feature maps for small object detection, Eng. Appl. Artif. Intell. 91 (2020) 103615.
[8] D. Zeng, F. Zhao, S. Ge, W. Shen, Fast cascade face detection with pyramid network, Pattern Recognit. Lett. 119 (2018) 180–186.

[9] Y. Bai, Y. Zhang, M. Ding, B. Ghanem, Finding tiny faces in the wild with generative adversarial network, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 21–30.
[10] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: asurvey, arXiv preprint arXiv:1809.02165 (2018).
[11] Y. Zhang, M. Ding, Y. Bai, B. Ghanem, Detecting small faces in the wild based on generative adversarial network and contextual information, Pattern Recognit. 94 (2019) 74–86.
[12] 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.
[13] P. Zhu, et al., VisDrone-VID2019: the vision meets drone object detection in video challenge results, in: IEEE International Conference on Computer Vision (ICCV) Workshops, 2019.
[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[15] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4724–4733.
[16] C. Feichtenhofer, A. Pinz, A. Zisserman, Detect to track and track to detect, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3038–3046.
[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 Trans. Circuits Syst. Video Technol. 28 (10) (2017) 2896–2907.
[18] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, J. Wang, Object detection in videos by high quality object linking, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[19] 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, pp. 817–825.
[20] 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, pp. 408–417.
[21] 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.
[22] 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.
[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. Berg, SSD: single shot multibox detector, in: European Conference on Computer Vision (ECCV), 2016, pp. 21–37.
[24] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525.
[25] 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, pp. 2117–2125.
[26] J. Yuan, H.-C. Xiong, Y. Xiao, W. Guan, M. Wang, R. Hong, Z.-Y. Li, Gated CNN: integrating multi-scale feature layers for object detection, Pattern Recognit. (2019) 107–131.
[27] W. Ma, Y. Wu, F. Cen, G. Wang, MDFN: multi-scale deep feature learning network for object detection, Pattern Recognit. 100 (2020) 107–149.
[28] P. Hu, D. Ramanan, Finding tiny faces, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 951–959.
[29] J. Noh, W. Bae, W. Lee, J. Seo, G. Kim, Better to follow, follow to be better: towards precise supervision of feature super-resolution for small object detection, in: IEEE International Conference on Computer Vision (ICCV), 2019, pp. 9725–9734.
[30] 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, pp. 1134–1142.
[31] S. Zagoruyko, A. Lerer, T.-Y. Lin, P.O. Pinheiro, S. Gross, S. Chintala, P. Dollár, A multipath network for object detection, in: British Machine Vision Conference (BMVC), 2016.
[32] Z. Cai, N. Vasconcelos, Cascade R-CNN: delving into high quality object detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6154–6162.
[33] J. Wang, X. Tao, M. Xu, Y. Duan, J. Lu, Hierarchical objectness network for region proposal generation and object detection, Pattern Recognit. 83 (2018) 260–272.
[34] X. Tao, Y. Gong, W. Shi, D. Cheng, Object detection with class aware region proposal network and focused attention objective, Pattern Recognit. Lett. 130 (2020) 353–361.
[35] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems (NIPS), 2014, pp. 568–576.
[36] G. Gkioxari, J. Malik, Finding action tubes, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 759–768.
[37] X. Peng, C. Schmid, Multi-region two-stream R-CNN for action detection, in: European Conference on Computer Vision (ECCV), 2016, pp. 744–759.

Brais Bosquet received the B.Sc. degree and the M.Sc. degree from the Universidade de Santiago de Compostela, Spain, in 2014 and 2015, respectively. He is currently a Ph.D. candidate at the Universidade de Santiago de Compostela. His research interests include object detection, neural networks and image processing.

Manuel Mucientes is an associate professor in computer science and artificial intelligence within the CiTIUS of the Universidade de Santiago de Compostela. He has authored or coauthored more than 100 papers in international journals, book chapters, and conferences. His current research interests are computer vision, in the topics of object detection and tracking based on deep learning; robotics, focused on UAVs (Unmanned Aerial Vehicles); process mining, and machine learning.

Victor M. Brea received the Ph.D. degree in Physics from the Universidade de Santiago de Compostela, Spain, in 2003. He is currently an associate professor in the Centro Singular de Investigación en Tecnoloxas da Información (CiTIUS), Universidade de Santiago de Compostela, Spain. His main research interests include object detection and tracking in the field of computer vision, as well as CMOS vision sensors, and the design of energy efficient sensors.