Correlation-based ConvNet for Small Object Detection in Videos

Brais Bosquet and Manuel Mucientes and Víctor M. Brea
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)
Universidade de Santiago de Compostela, Spain
Email: {brais.bosquet, manuel.mucientes, victor.brea}@usc.es

Abstract—**The detection of small objects is of particular interest in many real applications. In this paper, we propose STDnet-ST, a novel approach to small object detection in video using spatial information operating alongside temporal video information. STDnet-ST is an end-to-end spatio-temporal convolutional neural network that detects small objects over time and correlates pairs of the top-ranked regions with the highest likelihood of containing small objects. This architecture links the small objects across the time as tubelets, being able to dismiss unprofitable object links in order to provide high-quality tubelets. STDnet-ST achieves state-of-the-art results for small objects on the publicly available USC-GRAD-STDdb and UAVDT video datasets.**

I. Introduction

Convolutional neural networks (ConvNets) have achieved significant success for object detection [1], [2], [3]. Nevertheless, the performance drops as objects get smaller [4] and, consequently, small object detection is progressively gaining more interest in the scientific community [5], [6], [4]. Small object detection is of particular interest in applications such as sense and avoid on board of Unmanned Aerial Vehicles (UAVs) or video surveillance tasks, where objects appear usually small. Apart from being harder to detect, the scarcity of small annotated objects plays a fundamental role. This has recently been addressed with new specific small object datasets, such as USC-GRAD-STDdb [5], or UAVDT [7], where the number of small objects is quite significant. Small objects are typically defined as objects with a size below 32 × 32 pixels in widely adopted image datasets as MS COCO [8], or even smaller, below 16 × 16 pixels as in the case of USC-GRAD-STDdb [5].

The rise of spatio-temporal detection approaches, due to the release of ImageNet video object detection challenge (VID) [9], has led to an improvement of the precision of the detections. These networks exploit information from several frames by linking the same objects across the video to form sequences, or tubelets [10], [11], [12].

This paper addresses small object detection in video with STDnet-ST, a novel spatio-temporal convolutional neural network. STDnet-ST is built on STDnet [5] that provides the most promising areas where to look for small objects while dismissing the rest of the image. In this paper, we define small objects as any object of less than 16 × 16 pixels without definitive visual cues to assign them to a category, following our previous work [5]. Still, STDnet-ST is able to detect larger objects, like small objects as defined in MS COCO [8]. In summary, the main contributions of this paper are:

- STDnet-ST, a spatio-temporal neural network built on STDnet that operates with two input frames that are integrated through a correlation module. The correlation is performed in a natural way over the most promising regions with a high likelihood of having objects.
- STDnet-ST ends up in a final tubelet linking procedure based on the Viterbi algorithm. The tubelet linking has three novelties: (i) it uses the correlations generated by the ConvNet to link the objects of the tubelet; (ii) it scores the associations between the objects taking into account the confidence variability of the tubelet; and (iii) the tubelet suppression algorithm avoids unprofitable tubelets by inserting additional nodes to each frame in the Viterbi algorithm based on the information coming from promising areas without detections.
- STDnet-ST achieves state-of-the-art results for small object detection on the publicly available datasets USC-GRAD-STDdb and UAVDT, over the small object subset *S* ($\leq$ 1,024 px^2), popularly reported in MS COCO [8] metrics, as well as in the very small object subset *XS* ($\leq$ 256 px^2), defined in [5].

II. Related Work

During the last few years, two lines of research based on deep learning have demonstrated their efficiency in object detection: two-stages, or region proposal based detectors following [13], and one-stage, or one-shot detectors like SSD [1] or YOLO [3]. Based on these architectures, a large number of outstanding improvements has been made [14], [15], [16], [5].

When it comes to the small object detection field, the trend is to work with shallow feature maps, where small objects still have distinctive features. In this line, the two-stage Feature Pyramid Network (FPN) [15] has based its success on merging feature maps at different scales with a Region Proposal Network (RPN) per scale [13]. This design not only works well for detecting small objects on its shallow

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).

feature map (stride 4), but it is also the baseline of the leading solutions in the global COCO object detection challenge[1]. In contrast, architectures like Faster R-CNN present stride 16, which might not suffice for a good accuracy in small object detection. Following the architecture of FPN, RetinaNet [16] removes the RPNs and adds a class subnet and a bounding box subnet to detect objects in one-stage, including small ones.

Another approach to diminish the RPN stride was proposed in [5], where STDnet looks for the top-ranked regions with more likelihood of containing small objects from shallow layers while dismisses the remaining part of the input image. This allows STDnet to keep a low stride of 4 throughout the network, improving the final accuracy while keeping a reasonable computing time.

Concerning video object detection, several methods have been re-adapted from successful architectures for action detection [17], [18]. The two-streams ConvNets have achieved remarkable results. For example, in [18], a Faster R-CNN with two RPNs operates over two streams: a spatial RGB image input and a motion input obtained by applying optical flow over several frames. In [10], two input frames are correlated to extract motion information of the objects across time. The correlation operator computes the entire feature maps at different scales and estimates local features similarity for various offsets between the two frames. Finally, they link the detected objects into tubelets and reweight the detections' scores within them. The problem of using the whole feature maps is that, as an object becomes smaller, its movement has a considerably smaller influence in the correlation.

Differently, [19] tracks the detections in the current frame through neighboring frames to modify the original detections for higher accuracy. The linking among detections is based on the mean optical flow vector within boxes. Similarly, in [11] they link objects into long tubelets using a tracking algorithm, and then adopt a classifier to aggregate the detection scores in the tubelets. Finally, [12] 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 this paper we propose STDnet-ST, a spatio-temporal ConvNet for video object detection. STDnet-ST is based on our previous network, STDnet [5], and takes as input two consecutive frames. Through the underlying STDnet, STDnet-ST computes a fixed number of the top-ranked regions with more likelihood of containing small objects from each input, which are correlated between them. The main difference with previous correlation-based solutions like [10] is that STDnet-ST runs the correlation operator not on the whole feature maps, but in small areas with a high likelihood of having objects inside. This is essential for small object detection, as the correlation values calculated for the whole feature maps are mostly due to the background, while correlating specific regions allows to obtain values influenced by the objects.

III. STDnet-ST Architecture

STDnet-ST is a spatio-temporal convolutional neural network for the detection of small objects in video composed of two components: the spatio-temporal ConvNet and the tubelet linking.

The spatio-temporal ConvNet takes as inputs the current (f_t) and previous (f_{t-1}) frames, and returns the set of detections ($\mathcal{D}_t$) together with their confidences ($\mathcal{P}_t$). For each paired detections at t and $t-1$, it also computes the correlations (C_t, Sec. III-A) that will be used to associate the detections from different time instants.

The STDnet-ST tubelet linking comprises the correlation-based tubelet linking and the tubelet suppression procedure. The correlation-based tubelet linking (Sec. III-B2) generates the optimal tubelets along time for each of the objects by linking the detections obtained at different time instants ($t = 1, \dots, \tau$). The tubelets allow to modify the spatial confidence of the detections at τ using the previous $\tau - 1$ detections (Sec. III-B2). Also, the scores are updated taking into account the confidence variability of the tubelet. An essential step for the tubelet linking is the use of the correlation provided by the spatio-temporal ConvNet, which evaluates the likelihood of the association of two detections. Finally, the tubelet suppression algorithm (Sec. III-B3) dismisses the tubelets obtained by the correlation-based tubelet linking that might contain incorrect data associations. This is achieved, again, through the correlation operator.

A. Spatio-temporal ConvNet

STDnet-ST consists of two sibling branches together with a correlation operation between them. Each of the branches is based on the STDnet architecture [5], which is focused on the detection of small objects in images. Figure 1 shows the architecture of STDnet-ST.

STDnet performs well in detecting small objects by keeping high resolution feature maps across the entire network. This is possible because STDnet focuses its attention only on the regions of the image that most likely contain small objects. The main components of STDnet are the following —for a more detailed description refer to [5]:

- Region Context Network (RCN) and RoI Collection Layer (RCL). RCN is a novel detector of promising areas applied at early stages of the computation to select those regions that most likely contain small objects. Then, the m_t top-ranked regions $\mathcal{R}_t = \{r_t^1, \dots, r_t^{m_t}\}$ are gathered in a single feature map by the RCL. The RCN returns regions of a fixed size that most likely contain an object centered in it. From these regions, the RCL generates a new synthetic feature map of disjoint feature map parts — i.e., two neighboring pixels in the feature map that belong to two different regions might not be neighboring pixels in the original image.

[1] http://cocodataset.org/#detection-leaderboard (Accessed: 2020-03-18)

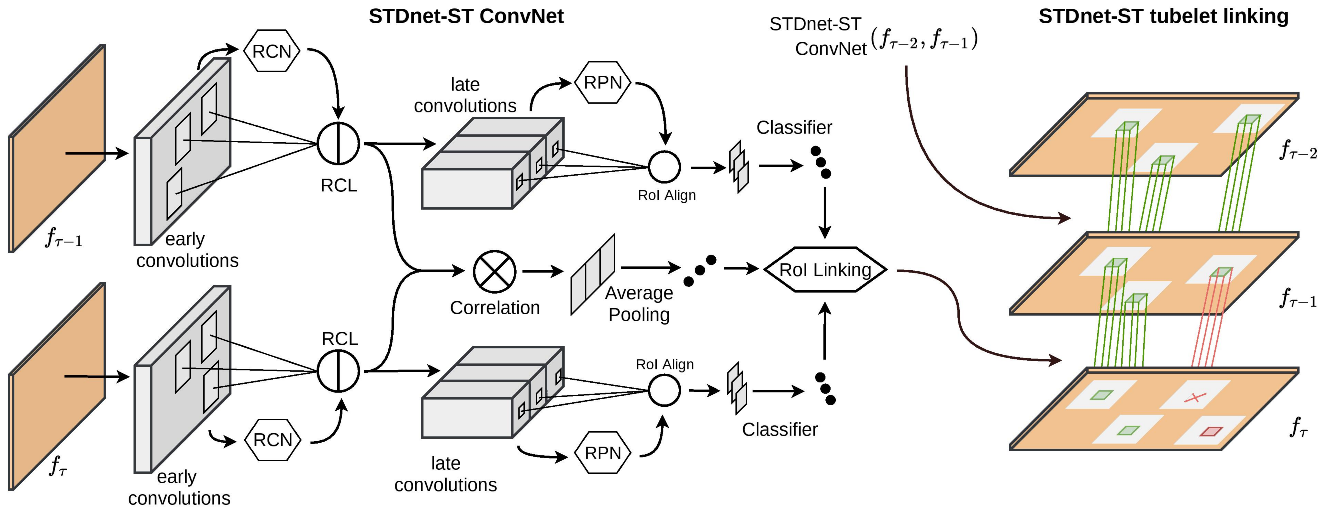

Fig. 1. STDnet-ST architecture. The STDnet-ST ConvNet takes two consecutive frames f_τ and $f_{\tau-1}$ as input and processes them through two branches that perform RCN+RCL to obtain the most promising regions (RCN regions). An RPN and a classifier locate objets inside the RCN regions. Simultaneously, the two sets of RCN regions feed a correlation module that associates the final detections. Finally, the STDnet-ST tubelet linking generates tubelets by binding the objects in the last τ frames and removing those that are unprofitable.

- Early and late convolutions. These convolutions belong to the selected backbone. Late convolutions have a high resolution —the same resolution as the last early block— due to the memory saved by ruling out non-promising areas. The RCL output is a feature map with disjoint areas, so late convolutions are designed to keep the features of each region separated from each other through padding.
- Region Proposal Network (RPN) and classifier. An RPN is applied to seek small objects on the last late convolutional block of promising areas. Finally, STDnet refines the outputs of the RPN to assign each proposal to a category, and by performing a category adapted bounding box regression.

The inputs to the STDnet-ST are two consecutive video frames, t and $t-1$. Both images pass through two STDnet-based branches that share the same weights throughout the execution. Each of the branches generates a set of detections ($\mathcal{D}_t$) and their corresponding confidences ($\mathcal{P}_t$). The correlation placed between them assesses the degree of matching between a pair of RCN regions at t and $t-1$. Then, as each RCN region specializes in detecting a single object centered in it, the correlation value between two RCN regions at t and $t-1$ is assigned to the final detections, contained in them, which are generated by the RPN and the classifier.

The correlation module consists of the two synthetic feature maps generated by RCL for each branch, a correlation operator, an average pooling and a final RoI linking operation. First, the correlation operator evaluates each pair of RCN regions $< r^i_{t-1}, r^j_t >$, where $r^i_{t-1} \in R_{t-1}$, $r^j_t \in R_t$, $i = 1, \ldots, m_{t-1}$, and $j = 1, \ldots, m_t$, and R_t is the set of regions generated by the RCN at time t. The output is a feature map with $m_{t-1} \times m_t$ regions, each representing the correlation between two of the RCN regions. Then, an average pooling is applied to summarize each of the regions of the correlated feature map in a single value associated with each pair of RCN regions, generating $m_{t-1} \times m_t$ correlation scores. The kernel size employed by the average pooling has the same size as the correlated regions which, at the same time, are the same size as the input RCN regions. Finally, the correlation scores of each pair of RCN regions are associated with the final detections by the RoI linking operation. The RoI linking operation takes as input the final detections ($\mathcal{D}_t$) from each STDnet-ST branch, as well as the correlation scores, and outputs the correlation scores but associated with 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 ($\mathcal{D}_t$). Even though not all RCN regions have an associated final detection, these correlation scores —not included in $\mathcal{C}_t$— are saved, as they are involved in the tubelet suppression algorithm (Sec. III-B3).

B. *STDnet-ST tubelet linking*

The object linking involves the association of an object within a target frame with the same object in the previous τ frames, generating the so-called tubelets. The final goal is to increase the confidence of those detections in the target frame that have assembled a tubelet, i.e., that have a high likelihood of being true positives, or to reduce the confidence of those detections within the target frame that have not produced a tubelet, i.e., that have a low likelihood of being true positives.

1) Baseline tubelet linking: The baseline tubelet linking is based on [20] but applied to object detection in video instead of to action detection. First, the tubelet linking calculates the set of scores between all possible pairs of detections in two consecutive time instants, the score matrix $\mathcal{S}_t = \{s^{11}_t, \ldots, s^{n_{t-1}n_t}_t\}$, where s^{ij}_t is the score between two equal category detections d_{t-1}^{i} and d_t^j in two consecutive frames. So that, s_t^{ij} estimates the likelihood that the i-th detection at frame $t-1$ and the j-th detection at frame t are both true positive detections and come from the same object. s_t^{ij} is given by:

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

where p_t^j is the confidence returned by the ConvNet for the j-th detection at frame t, IoU is the overlap or intersection over union between two detections, and λ balances the importance between the confidences and the IoU.

Then, the Viterbi algorithm computes the most probable sequences of detections, i.e., the tubelets, where each one represents the same object at different time frames. Thus, it maximizes the conditional probability of the possible tubelets $\mathcal{V}$ given a set of detections $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_\tau\}$ and their corresponding scores $\mathcal{S} = \{\mathcal{S}_2, \ldots, \mathcal{S}_\tau\}$ over time of the same category:

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

where $i\left(v\right)$ and $j\left(v\right)$ are the detections at times $t-1$ and t for a given tubelet $v \in \mathcal{V}$. Then, after each 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 confidences for the detections in the target frame τ within one of the optimal tubelets $\hat{v}$ are updated as the average confidences of the detections within $\hat{v}$:

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

where $p_t^{i(\hat{v})}$ is the confidence of the i-th detection at time t belonging to tubelet $\hat{v}$.

2) Correlation-based tubelet linking: There are several cases where using IoU overlap might be a weak feature for matching objects. To name a few: objects so small that they barely overlap even if they move slowly, fast object and/or camera motions, objects very close to each other, or low frame rate videos. The proposed correlation-based tubelet linking avoids IoU matching and the drawbacks that this entails by introducing the correlation score as the feature for data association.

To do this, correlation-based tubelet linking replaces the overlap between detections $\mathrm{IoU}(d_{t-1}^{i}, d_t^{j})$ in Eq. 1 with c_t^{ij}, which is the correlation calculated by the STDnet-ST ConvNet for the i-th detection at time $t-1$ and the j-th detection at time t:

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

As an added novelty, we propose the use of the confidence variability within a tubelet to update the confidences, understanding that when it is small, the last detection is more likely to be a true positive. When this happens, the maximum confidence of the tubelet will be assigned to the detection and, otherwise, the average will be selected –as in the baseline tubelet linking. Equation 3 will be reformulated as follows:

$$p_\tau^{i(\hat{v})} = \begin{cases} \max_{t=1}^{\tau} p_t^{i(\hat{v})} & \text{if } \sigma(\{p_t^{i(\hat{v})}\}_{t=1}^{\tau}) \leq \kappa \\ \frac{1}{\tau}\sum_{t=1}^{\tau} p_t^{i(\hat{v})} & \text{otherwise} \end{cases} \tag{5}$$

where σ is the standard deviation of the confidences of the tubelet $\hat{v}$, and κ is a threshold.

3) Tubelet suppression procedure: The original Viterbi algorithm generates all possible tubelets $\hat{\mathcal{V}}$ regardless of the fact that the detections that make them up might be false positives. For example, there might be a tubelet created with false and true positive detections, only because there is no other possible data association. STDnet-ST tubelet linking defines the tubelet suppression algorithm to identify this pattern. This algorithm adds dummy detection nodes that the Viterbi algorithm might use to build a tubelet, and these tubelets will be later deleted.

The complete STDnet-ST tubelet linking process includes both, the correlation-based tubelet linking and the tubelet suppression procedure. This process has three inputs: the set of detections $\mathcal{D}$ from time $t=1$ to $t=\tau$, the confidences $\mathcal{P}$ for each detection, and the set of score matrices $\mathcal{S}$. Each s_t^{ij} is the element ij of matrix $\mathcal{S}_t$ computed following Eq. 4 with the i-th detection at time $t-1$ and the j-th detection at time t. The output of the algorithm is the set of updated confidences $\hat{\mathcal{P}}_\tau$ associated to each detection at time τ.

The first step is to initialize $\hat{\mathcal{P}}_\tau$ with the confidences generated by the ConvNet. Next, for each t, we add a dummy node in $\mathcal{D}_t$ and its corresponding scores to $\mathcal{S}_t$ —a new column and a new row. In the case of the new column, these new scores are the result of Eq. 4 using the maximum correlation between each detection at $t-1$ and all RCN regions at time t that have been discarded by the ConvNet as objects. For the rows it is analogous, but by correlating the regions with detection in t with the regions without detection in $t-1$.

Then, the Viterbi algorithm is applied to $\mathcal{D}$ and $\mathcal{S}$ to find optimal tubelets. For every generated tubelet, the corresponding not dummy detections are deleted from the set of detections $\mathcal{D}$ as well as the corresponding row and column from the score matrices $\mathcal{S}$. Finally, if the tubelet does not contain dummy nodes, it is a valid one, so the confidence of the detection at time τ in the tubelet is updated following Eq. 5. The Viterbi algorithm is repeated iteratively as long as $\mathcal{D}_t$, from $t=1$ to $t=\tau$, still has detections provided by the STDnet-ST ConvNet.

IV. Experiments

A. Evaluation metrics and datasets

The evaluation metrics reported are those of MS COCO [8]: Average Precision when IoU is at least 50% (AP$^{@.5}$) and Average Precision when the IoU goes from 50% to 95% in 5% steps (AP$^{@[.5,.95]}$). In the default COCO metrics, the results are shown for three different subsets: *small* (AP$_s$), objects smaller than 1,024 pixels area; *medium* (AP$_m$), objects between 1,024 and 9,216 pixels area, and *large* (AP$_l$), objects larger than 9,216 pixels area. In USC-GRAD-STDdb, almost all objects have a size smaller than 16 × 16 pixels [5], so we define a new scale subset, *very small* (AP$_{xs}$), to include small targets as defined in this paper, i.e., smaller than 256 pixels area. The *XS* subset is defined in order to evaluate the performance for very small objects. We do not modify the definition of the *S* subset to preserve the MS COCO standards.

We conduct extensive experiments on two publicly available datasets: USC-GRAD-STDdb [5] and UAVDT [7]. USC-GRAD-STDdb comprises 115 video segments with more than 25,000 annotated frames. The resolution of the video is HD 720p (1,280 × 720). The test subset holds 11,337 objects, where almost 90% of them (10,136 objects) correspond to the *very small* subset, which leads us to only evaluate that subset, since any other subset does not contain enough data. UAVDT contains 23,829 frames of training data and 16,580 images of test data of $\approx$ 1,024 × 540 resolution. 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%) and 281,532 within the *small* subset (74.9%) —with the *very small* subset included into the *small* subset.

B. Implementation Details

We implemented STDnet-ST based on STDnet [5] updated to Caffe2. The input size is determined by the resolution of the dataset under study. The RCN region size for USC-GRAD-STDdb is 32 × 32, as most of the objects belong to the *XS* size. The training of STDnet-ST is done for 40k iterations with two step decay. For UAVDT, with objects with more varying sizes, including those larger than the *S* category, the training process requires pre-training. Thus, first, we run a pre-training phase with Faster R-CNN during 20k iterations followed by a fine-tuning with STDnet-ST for other 20k iterations with two step decay. In order to retrieve all objects within the *S* category, we set the RCN region size to 48 × 48 pixels. Also, as reported in [5], for both datasets, RCN between *conv3* and *conv4* and the initialization of anchors by k-means lead to the best performance metrics. We set the base learning rate to 0.0025, a momentum of 0.9, and a decay parameter of 0.0001 on weights and biases. The spatio-temporal hyperparameters τ and κ are set to 4 and 0.02, 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 [21]. The Faster R-CNN [13] with Feature Pyramid Network (FPN) [15] is adopted as the baseline detection network.

C. Results on USC-GRAD-STDdb

Table I 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 III-B1; *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. 5; *Correlation linking* means the correlation-based tubelet linking as addressed in Section III-B2; and *Tubelet suppression* concerns the tubelet suppression procedure presented in Section III-B3. 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 [5].

TABLE I
ABLATION STUDY ON USC-GRAD-STDDB FOR THE DIFFERENT TUBELET LINKING COMPONENTS OF STDNET-ST.

Baseline linking	Confidence variability	Correlation linking	Tubelet suppression	AP$_{xs}^{@[.5,.95]}$	AP$_{xs}^{@.5}$
—				18.3	57.8
✓				19.2	59.9
✓	✓			19.3	60.3
		✓		19.3	60.1
	✓	✓		19.5	60.5
		✓	✓	19.8	61.4
	✓	✓	✓	**20.1**	**62.1**

TABLE II
EVALUATION METRICS ON THE USC-GRAD-STDDB DATABASE.

Method	AP$_{xs}^{@[.5,.95]}$	AP$_{xs}^{@.5}$
FPN [15]	17.3	54.5
FPN [15] ++	18.7	57.2
STDnet [5]	18.3	57.8
STDnet-ST	**20.1**	**62.1**

As it can be observed, STDnet-ST outperforms STDnet from 18.3% to 20.1% for AP$_{xs}^{@[.5,.95]}$ and from 57.8% to 62.1% for AP$_{xs}^{@.5}$. The correlation-based linking, together with the confidence variability contribute to increase 0.3% AP$_{xs}^{@[.5,.95]}$ and 0.6% AP$_{xs}^{@.5}$ —Table I, rows 2 and 5. Also, the tubelet suppression procedure adds a gain of 0.6% AP$_{xs}^{@[.5,.95]}$ and 1.6% AP$_{xs}^{@.5}$ over the previous result —Table I, rows 5 and 7. Two conclusions can be drawn from Table I. First, the importance of the correlation obtained by the ConvNet of STDnet-ST, as it contributes to both the correlation-based linking and the tubelet suppression. 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 II provides a comparison in terms of accuracy between the state-of-the-art FPN and our STDnet-ST, which outperforms FPN by 2.8% AP$_{xs}^{@[.5,.95]}$ and 7.6% AP$_{xs}^{@.5}$. Since the baseline tubelet linking and the confidence variability methods are independent of the architecture —they only consider detections by frame—, Table II also compares the performance of FPN with these components —referred as FPN++—, i.e., FPN with spatio-temporal information, which improves its baseline results by 1.4% AP$_{xs}^{@[.5,.95]}$ and 2.7% AP$_{xs}^{@.5}$. Even so, the results of the spatio-temporal FPN remain below STDnet-ST by 1.4% AP$_{xs}^{@[.5,.95]}$ and 4.9% AP$_{xs}^{@.5}$.

TABLE III
EVALUATION METRICS ON THE *very small* AND *small* SUBSETS OF THE UAVDT DATABASE.

Method	$\text{AP}_{xs}^{@[.5,.95]}$	$\text{AP}_{s}^{@[.5,.95]}$	$\text{AP}_{xs}^{@.5}$	$\text{AP}_{s}^{@.5}$
Faster R-CNN [7]	6.6	16.1	26.0	49.9
R-FCN [7]	9.2	20.7	32.5	53.4
RON [7]	3.7	15.1	19.7	49.0
SSD [7]	6.0	20.1	23.5	**54.3**
FPN [15]	11.8	24.8	29.7	49.2
FPN [15] ++	12.0	25.1	30.3	49.7
STDnet [5]	12.5	27.0	35.1	53.7
STDnet-ST	**13.1**	**27.2**	**36.0**	54.0

D. Results on UAVDT

The experimental results on the UAVDT dataset [7] are shown in Table III, which compares the performance between our approach, its baseline STDnet, Faster R-CNN, R-FCN, RON, SSD, FPN and FPN++. The first four rows are computed using the bounding box results provided in [7], and directly adapted to the MS COCO results format [8].

First of all, the results confirm that the spatial STDnet performs better than the rest of the state-of-the-art spatial approaches, not only for very small objects, but also for the *small* subset. The detections provided by STDnet fit better to the ground truth, yielding 0.7% $\text{AP}_{xs}^{@[.5,.95]}$ and 2.2% $\text{AP}_{s}^{@[.5,.95]}$ higher than any other spatial approach. This metric is considered the primary challenge metric by MS COCO [8] because it encompasses AP adding information on how it behaves as the IoU reaches perfection. It is also noteworthy that STDnet outperforms FPN++, which exploits 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* and *small* subsets. STDnet-ST overcomes spatio-temporal FPN (FPN++) by 1.1% $\text{AP}_{xs}^{@[.5,.95]}$ and 2.1% $\text{AP}_{s}^{@[.5,.95]}$, and also R-FCN by 3.5% $\text{AP}_{xs}^{@.5}$.

V. CONCLUSION

We have introduced STDnet-ST, a spatio-temporal ConvNet to detect small targets in video. STDnet-ST is composed of two ConvNet 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.

We have conducted experiments over two publicly accessible datasets with a large number of small objects: USC-GRAD-STDdb [5] and UAVDT [7]. STDnet-ST achieves state-of-the-art results in both datasets for very small objects, clearly outperforming its counterparts by 1.4% $\text{AP}_{xs}^{@[.5,.95]}$ on USC-GRAD-STDdb, and by 1.1% $\text{AP}_{xs}^{@[.5,.95]}$ on UAVDT. In addition, STDnet-ST obtains the best result on the UAVDT *small* subset, that encompasses the 67.5% of its test objects, reaching a precision of 27.2% $\text{AP}_{s}^{@[.5,.95]}$, which is an increase of 2.1% over any other method.

REFERENCES

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

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[4] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv preprint arXiv:1809.02165*, 2018.

[5] B. Bosquet, M. Mucientes, and V. M. Brea, "STDnet: A ConvNet for Small Target Detection," in *British Machine Vision Conference (BMVC)*, 2018, p. 253.

[6] P. Hu and D. Ramanan, "Finding tiny faces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 951–959.

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

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.

[9] 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," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3038–3046.

[11] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang *et al.*, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2017.

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

[13] S. Ren, K. He, R. Girshick, and 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.

[14] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 379–387.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[17] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 568–576.

[18] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," in *British Machine Vision Conference (BMVC)*, 2016.

[19] K. Kang, W. Ouyang, H. Li, and 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. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 744–759.

[21] S. Gidaris and 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.