ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200406

RoI Feature Propagation for Video Object Detection

Daniel Cores and **Manuel Mucientes** and **Víctor M. Brea**[1]

Abstract. How to exploit spatio-temporal information in video to improve the object detection precision remains an open problem. In this paper, we boost the object detection accuracy in video with short- and long-term information. This is implemented with a two-stage object detector that matches and aggregates deep spatial features over short periods of time combined with a long-term optimization method that propagates detections' scores across long tubes. Short-time spatio-temporal information in neighboring frames is exploited by Region-of-Interest (RoI) temporal pooling. The temporal pooling works on linked spatial features through tubelets initialized from anchor cuboids. On top of that convolutional network, a double head processes both temporal and current frame information to give the final classification and bounding box regression. Finally, long-time information is exploited linking detections over the whole video from single detections and short-time tubelets. Our system achieves competitive results in the ImageNet VID dataset.

1 INTRODUCTION

Object detection has been one of the most active research topics in computer vision for the past years. Although object detection accuracy in images has improved significantly based on Convolutional Neural Networks (CNN), the use of temporal information in videos to boost the detection precision or to perform action recognition is still an active research area.

The simplest way to detect objects in video is to execute an object detection framework at frame level that only uses spatial information. The main issue with this approach is that it does not exploit the temporal information available in videos to address challenges such as motion blur, occlusions or changes in objects appearance at certain frames.

Object detection frameworks implement two main tasks: bounding box regression and object classification. The bounding box regression in a given frame is highly related with the spatial information available in that frame. However, the appearance of one object in previous frames might provide valuable information to classify the object in the present frame. This brings about the problem of how detections in different frames are linked and how the system aggregates this spatio-temporal information.

State-of-the-art object detection frameworks are based on two major approaches: one stage and two stage architectures. In one stage methods [21], [23], the network head has to process a dense set of candidate object locations, with a high imbalance between objects of interest and background examples. Two stage frameworks [9], [10], [24] address this issue adding an object proposal method as the first stage that filters out most of the background candidate locations. Then, the network head refines the proposal set. We develop a two stage spatio-temporal architecture in which the proposals generated by the first stage are also used to propagate information from previous frames.

This paper proposes a novel method that extends the Faster R-CNN [24] framework to use both temporal and spatial information to enhance the network classification accuracy. Our deep convolutional network can be trained end to end. Also, it implements a double head approach [29], one of the branches to process temporal information, and another to deal with spatial data. As the network is built on the Feature Pyramid Network (FPN) [19] backbone, the design is more complex than single-scale approaches, as it requires to add a multiple Region Proposal Network (RPN) and a multiple level Region of Interest (RoI) feature extraction.

The main contributions of this work are:

- A tubelet proposal method that works with FPN models dealing with multiple Region Proposal Networks (RPN) and extracting RoI features at different pyramid levels. As far as we know, this is the first spatio-temporal framework with a multiple level backbone such as FPN.
- A temporal propagation method of spatial features: the proposed network includes a method that aggregates spatial information from the previous N frames. This method summarizes the information in order to provide an output feature map with the same size as if the network was working with a single frame. Therefore, the processing time of the network head remains constant, independently of N.
- A spatio-temporal double head. One branch of the network head calculates the final bounding box regression and the first guess about the object class probabilities based on the information extracted from the current frame. The other branch complements the classification output using the accumulated information through the previous N frames, including the current one. This improves the object classification task, not only allowing to distinguish better among different object categories, but also between object and non-object.
- A long-term optimisation method creates long tubes that associate object instances throughout the video, rescoring all detections belonging to the same tube. Standard linking algorithms match detections in one frame with the next one, being unable to give a complete tube if some detections are missed. To overcome this, our linking algorithm uses the tubelets initially calculated by the convolutional network to join broken fragments of the same tube.

2 RELATED WORK

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

[1] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Spain, email: {daniel.cores, manuel.mucientes, victor.brea}@usc.es

Two stage architectures were first popularized by R-CNN [10]. This method takes a precalculated object proposal set and then applies a deep CNN to extract per region features to perform object classification. This method was improved in Fast R-CNN [9], adding a RoI pooling layer that allows to run a per image CNN instead of per region. This work also modifies the network header to calculate both object classification and bounding box regression to refine the proposals. This way, all backbone computations can be reused, improving the execution time. All these methods rely on a region proposal method independent of the network. This issue is addressed in Faster R-CNN [24], defining a Region Proposal Network (RPN) that performs the proposal generation task based on the same deep features used by the network head for object classification and bounding box regression. The R-FCN object detector [2] re-implements the network header avoiding the fully connected layers used by previous work. Instead, it follows a fully convolutional approach changing the RoI pooling by a position sensitive RoI pooling.

One stage detectors such as SSD [21] and YOLO [23] do not refine a previously calculated proposal set. Instead, they generate a dense grid of bounding boxes and directly calculate the final detection set. Having a dense set of proposals increases the imbalance between object of interest and background examples due to the lack of a first step that filters most of the negative proposals. Recent works [20] try to overcome this problem modifying the cost function to prevent easy examples from having a huge impact on the network training process.

Video object detection. The recently introduced ImageNet object detection from videos (VID) challenge has brought significant attention over the video object detection problem. Still, how temporal information available in videos should be used to improve the detection performance remains an open problem.

Two-stream networks such as [1], [4], [5] or [27] have become the standard approach in action recognition. One of the branches processes video frames, while the other one takes precomputed dense optical flow frames as input. Diba *et al.* [3] proposes an end to end model able to extract temporal features in inference time by training the network with optical flow images. Although action recognition is a related problem, the benefits of adding optical flow information to spatio-temporal object detectors might not be straightforward. To be able to distinguish some action classes such as *"sitting down"* and *"getting up"*, motion information given by optical flow might be crucial. This is not so evident in object detection. This can be seen in [14], that uses the same architecture for object detection and action recognition, showing how traditional fusion methods work for actions but not for objects.

Still, optical flow has been proven successful in [34] for object detection. In this case, optical flow information finds feature correspondences in consecutive frames rather than providing motion cues for classification. These correspondences are used to aggregate spatial features over time. Since our framework is based on a two-stage architecture, we can extract RoI feature maps with a fixed size centered on the object applying RoI Align [12]. Associating proposals in consecutive frames allows us to directly associate features in the same position of each feature map in consecutive frames, avoiding the usually long calculation time of optical flow.

Short-term object linking making up tubelets has become a widely adopted technique in video object detection. Object tracking has been used to link detections generated by a frame level object detector in [17]. T-CNN [16] works on two single frame detectors and also applies tracking to link these detections over time. A slightly different approach is presented in [15], replacing the tracking algorithm by a Tubelet Proposal Network. Firstly, this network takes proposals for the first frame and extracts features applying the same static proposal across the time. Then, the pooled features are employed to calculate the bounding box displacement in each frame to build the tubelet proposal.

Alternatively, the idea of *anchor box* in single frame object detectors can be extended to the spatio-temporal domain. This idea is implemented in the ACT-detector [13] for action recognition and in [28] for spatio-temporal object detection. A Cuboid Proposal Network (CPN) was defined in [28] as the first step for short tubelet detection. We implement a similar idea, where each *anchor box* in the *anchor cuboid* is regressed independently by the corresponding RPN using information from one frame.

Feichtenhofer *et al.* [6] perform tracking and object detection simultaneously using a multi-task objective training. They use tracking information as input to a long-term object linking algorithm to build long tubes, and aggregate detection scores throughout the tube. Similar methods have been adopted by [28], linking small tubelets by the overlap of detections in the same frame instead of tracking, and by T-CNN [16] with a post processing tracking method. We extend this idea exploiting short-term information to overcome missed detections, being able to build larger tubes. In doing so, we only use object proposals and detections given by the network without any tracking method to aid the object linking.

3 SPATIO-TEMPORAL FRAMEWORK

The proposed framework is a two stage spatio-temporal object detector able to improve the accuracy in each frame t by taking as input a sequence of N frames $f_{t-N-1}, ..., f_{t-1}, f_t$. Figure 1 shows an overview of our network architecture. Although we will describe the framework in this paper based on the Feature Pyramid Network (FPN) backbone [19], the same ideas can be easily applied to single scale models. In fact, working with FPN is a more complex approach because of the multiple Region Proposal Network (RPN).

Our network backbone shares the convolutional weights among all input frames. Therefore, f_{t-1} at time t becomes f_{t-2} at time $t+1$, and so on. This reduces the impact of increasing the number of input frames in the system performance.

In our network architecture, initially, tubelets are sequences of N *anchor boxes*, one per frame, in the same position, with the same area and aspect ratio. Then, each RPN modifies the *anchor box* independently in each frame with features from the corresponding backbone (Section 3.1). We also use shared weights for the RPN throughout all input frames, so the RPN bounding box regression and classification can also be reused following the same strategy explained for the backbone. As a result, this process outputs a set of tubelet proposals that must be filtered to remove the spatially redundant ones. To do that, we add a Tubelet Non-Maximum Suppression (T-NMS) [28] algorithm on top of the RPNs. The tubelet generation is further addressed in Section 3.1.

Having the tubelet proposals and the backbone features, the RoI Align method [12] extracts a per frame and per tubelet feature map of fixed size. Figure 1 shows the most general case in which we have a multiple scale backbone applying RoI Align at different levels (3 levels in the figure). The RoI Align result is a fixed size feature map centered on the object and, therefore, features can be directly associated by position and propagated from neighboring frames. By doing this aggregation after the RoI Align method, we work with summarized data, so small changes in the object appearance in some frames

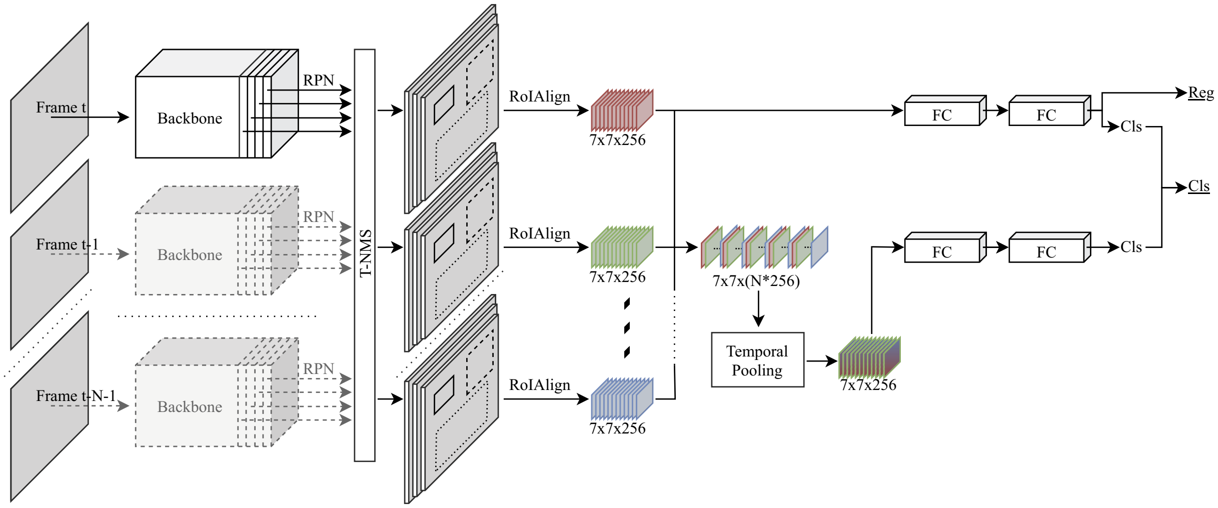

Figure 1. Network architecture. Since we use the same backbone weights for every frame, feature maps computed previously can be reused (lighter "Backbone" boxes). In this example, we have multiple RPNs and we extract RoI features at different pyramid levels. Bounding box proposals style border represent that the bounding boxes belong to the same tubelet.

do not have a significant impact in the output feature map.

Then, we apply a Temporal Pooling operator that takes as input a joined feature map from all RoI Align outputs associated with the same tubelet. As Figure 1 shows, this feature map has as joined dimension N times the original RoI Align size in every frame. The Temporal Pooling method reduces this dimension to a fixed size independently of N. Section 3.2 describes the joining and pooling processes in more detail.

Lastly, we introduce a double head approach to process both the current frame and the spatio-temporal information (Section 3.3). The spatial branch is fed by the RoI Align output in the current frame, while the spatio-temporal branch takes the Temporal Pooling output. This network can be trained end-to-end without heavily engineered proposals.

Object detections are linked, making up long tubes. The linking algorithm employs short-term information given by tubelet associations to grow the tubes. Classification score is propagated throughout each object tube, updating detections belonging to it (Section 3.4).

3.1 Tubelet proposals

The original Faster R-CNN model [24] works with a set of k *anchor boxes* at each sliding position $W \times H$, generating $W \times H \times k$ anchors in total. The network tubelets are initialized as a sequence of N *anchor boxes*, generating *anchor cuboids* [13]. Each *anchor box* in the *anchor cuboid* represents the same anchor in the same position for all frames. Therefore, the final number of *anchor cuboids* remains the same as the original number of *anchor boxes* in the single frame approach.

The FPN (Feature Pyramid Network) backbone [19] distributes the *anchor boxes* among the RPNs by area. Thus, each *anchor cuboid* is mapped to a pyramid level according to its *anchor boxes* area to be processed by the corresponding RPN. In our implementation, every single *anchor box* b^i in the *anchor cuboid* sequence $(b^{t-N-1}, ..., b^{t-1}, b^t)$ is regressed independently in its corresponding frame by each RPN. This means that, the RPN can resuse the bounding box regression calculations for $b^i \in (b^{t-N-1}, ..., b^{t-1})$, and only needs to calculate b^t at each time instant.

The proposed method generates spatially redundant tubelets in the same way the single-frame approach does with box proposals. Nevertheless, a per frame and per RPN non-maximum suppression (NMS) might break the tubelets, removing some of their bounding boxes. Instead, we apply an extension of the NMS algorithm called Tubelet Non-Maximum Suppression (T-NMS) proposed by [28] that discards redundant tubelet proposals. To do that, we define both the tubelet score and the overlap metric.

The score of a given tubelet τ_i is calculated as:

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

being b_s^i the score associated with the proposal b at frame i.

The overlap between a pair of tubelets τ_i and τ_j is defined as:

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

The T-NMS algorithm takes the whole set of tubelets and globally removes the spatially redundant ones instead of running a per RPN filtering. The resultant subset $\mathcal{T}$ will be the final collection of proposals.

3.2 Temporal Pooling

The RoI Align method [12] takes the proposal set filtered by the T-NMS algorithm and the backbone feature maps from the corresponding frame as the input data. In the spatial case, the FPN model maps each RoI to different feature map levels according to the bounding box area. In this spatio-temporal approach, for a given tubelet $\tau_i = (b_i^{t-N-1}, ..., b_i^{t-1}, b_i^t)$ every single bounding box b^j goes to its corresponding pyramid level in the frame f_j. This means that RoIs belonging to the same tubelet could be mapped to different pyramid levels in their respective frame. This makes our method robust against scale variations in the tubelet sequence. This scale variability can be caused by changes in the object appearance, specially in those objects whose area is close to the mapping threshold of a certain pyramid level. In these cases, small changes in the bounding box associated with one frame might cause this threshold to be exceeded in this specific frame. Even if the actual object does not change enough, different bounding boxes in the same tubelet might have enough area differences due to slight errors in the RPN regression. The reason is that the RPN output is just the first adjustment of the *anchor box*, and not the final detection box, so it does not fit the object with the same precision as the final output.

On top of that, we introduce an operator called *Temporal Pooling* that summarizes the whole tubelet information in a feature map with just the same size as the RoI Align output for a single frame, independently of the number of input frames. To be able to aggregate temporal data, the first step must be to find the correspondences among features throughout all input frames. The RPNs adjust *anchor boxes* applying bounding box regression, so all boxes in a given tubelet are optimized to fit the target object. Then, for each RoI in the tubelet, the RoI Align method produces a feature map of fixed size (in our case of $7 \times 7 \times 256$, Figure 1). Thus, since all these feature maps are centered on the object in each input frame, we can aggregate values in the same position in each feature map. Although errors in the RPN bounding box regression might cause feature mismatches, our method is robust against small variations since we use pooled features. These pooled features are a broad representation of features extracted from the backbone, so slight changes on the proposal bounding box do not have a significant effect.

Our method concatenates the N input feature maps with size $W \times H \times C$ into one feature map with size $W \times H \times N \cdot C$ (Figure 1). Then, feature map channels are reordered, so that channels at position i from all input feature maps are concatenated consecutively (see the input to the Temporal Pooling operator in Figure 1). Finally, the Temporal Pooling applies the following transformation to get each element of the resultant feature map X:

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

being x_{ijk} an element in the position $i \times j$ in channel k of the output feature map, and $y_{ij(N(k-1)+t)}$ an element in the position $i \times j$ in channel $(N(k-1)+t)$ from the concatenated feature map. This propagates the highest activation values through the tubelet.

3.3 Spatio-temporal double head

The network head follows a double head approach. As Figure 1 shows, we build two fully connected heads specialized in spatial and spatio-temporal information, respectively. On the one hand, the spatial head processes the output of the RoI Align method in the current frame. This branch calculates the final bounding box regression and the spatial classification based only on the current appearance of the object. On the other hand, the spatio-temporal head classifies the object based on the output of the Temporal Pooling operator. Therefore, this branch takes into account the appearance of the object in the previous N frames. This head does not propose a bounding box regression because the most relevant information to do that is the location of the object in the current frame and not in the previous ones.

Finally, both classification vector scores are aggregated as follows [29]:

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

where p_{spt} and p_{tmp} are the score vectors from the spatial and temporal heads, respectively.

3.4 Long-term object linking

The network outputs a set of detection boxes per category at frame level and their classification scores. Linking these boxes over time to identify single action/object instances has become a standard approach in both action recognition [11], [26], and recently in object detection [6], [16], [28]. These methods try to join single boxes or small tubelets into larger tubes in order to propagate the classification score throughout the video.

Our long-term object linking proposal performs a two step approach. First, the frame to frame linking problem is defined as an optimization problem that maximizes the global tube linking score to build long object tubes. Then, the second step joins these tubes, getting larger ones, overcoming network errors such as false negatives or misclassified detections that might break object tubes in the first step.

In this implementation, each detection $d_t^i = \{x_t^i, y_t^i, w_t^i, h_t^i, p_t^i\}$ in the set D_t indexed by i in the frame t, is composed of a box centered at (x_t^i, y_t^i) with width w_t^i and height h_t^i, and an associated confidence p_t^i for the object class. A lower threshold β is applied over the detection set D_t before the object linking algorithm to prevent that low confidence detections adversely affect the tube creation. The linking score $ls(d^i, d^j)$ between two detections d^i and d^j at different frames is defined as:

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

Then, the optimal tube $\hat{v}$ can be found by solving the following optimization problem applying the Viterbi algorithm to the detection set per object category:

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

where $\mathcal{V}$ is the set of all possible tubes.

Algorithm 1 describes the tube building process in detail. This method solves Equation 6 (Algorithm 1:4) iteratively to find all tubes ending in frame i. In each iteration, all detections in the best tube $\hat{v}$ are removed from the candidate set $\mathcal{D}$ (Algorithm 1:5) and the new tube is added to the result set (Algorithm 1:6). When there are no more candidate detections in the current frame (Algorithm 1:3), the same process is applied to find all tubes ending in the previous frame (Algorithm 1:2). Doing this iteratively leads to tubes of different lengths, linking all detections.

Algorithm 1: Long-term tubes creation

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

Previous work defined action/objects tubes as sequences of consecutive detection boxes, without interruptions. In fact, the output of the method described above fits completely this definition of object tube. Nevertheless, object occlusions or even network errors such as false negatives or misclassified examples could artificially break large tubes into smaller chunks. Figure 2 shows an example where an object tube is broken in two parts due to a network error. In this case, there is a false negative (there are two object instances and only one

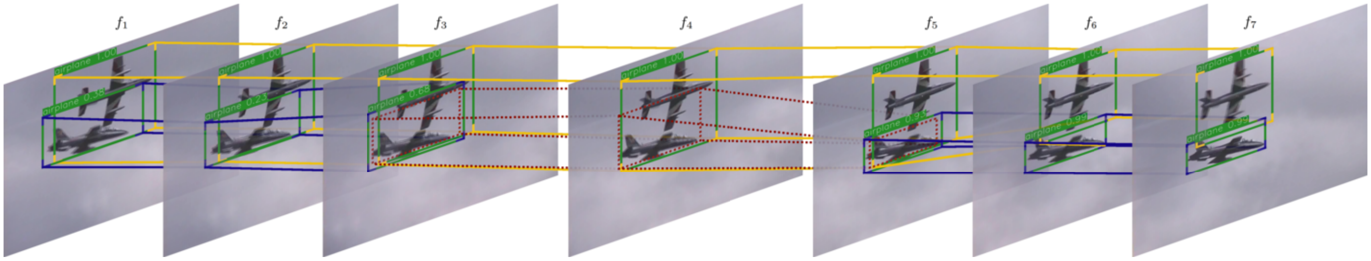

Figure 2. Long-term object-linking. Green boxes represent network detection outputs in a test video while dotted boxes represent RPN proposals in each frame (this example works with $N = 3$). The network joins two object instances in the same box detection in frame f^4, breaking the blue tube in two fragments. As the first box detection of the second fragment and the last detection of the first fragment belong to the same RPN tubelet, a larger single tube is made joining both small tubes just skipping f^4.

detection) in frame f_4 that breaks the blue tube, making one small tube from f_1 to f_3 and another one from f_5 to f_7.

RPN tubelet information solves this issue joining those small tubes to provide larger ones, making up the second step of the long-term object linking. Although the network head does not output a detection (green solid boxes) for each object in frame f_4 in Figure 2, the RPN outputs a tubelet with size N (dotted boxes in the figure), linking proposals in frames f_3 to f_5.

Algorithm 2: Long-term object tube linking

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

Algorithm 2 describes the tube linking method in detail. Formally, giving two tubes $\hat{v}_i = (d^{i,1}, ..., d^{i,m_i})$ and $\hat{v}_j = (d^{j,1}, ..., d^{j,m_j})$ with size m_i and m_j respectively, both will be joined in a tube of size $m_i + m_j$ if $d^{j,1}$ follows d^{i,m_i} in time, and both detections belong to the same RPN tubelet (Algorithm 2:6). Thus, the tubelet allows to link both tubes as it contains detections from both of them, although the tubes do not have temporal overlap.

The detection set D_t is the output of Non-Maximum Suppression (NMS) followed by a Bounding Box Voting transformation [7] to the actual network output. This method takes the highest score detection d and removes all other detections with an overlap with d higher than a given threshold. Therefore, the final detection d has many associated network outputs. Since each network detection came from one RPN tubelet, d has also many RPN tubelets associated, one per each suppressed detection. Therefore, there can be many tubelets τ that contain the first or the last detection box of a specific tube. The function $\gamma(b_l^k, d^{i,m_i})$ returns True if the detection d^{i,m_i} is associated with the box proposal b_l^k in the tubelet τ_l (Algorithm 2:6).

For a given tube $\hat{v}_i = (d^{i,1}, ..., d^{i,m_i})$ we might have more than one candidate tube $\hat{v}_j$ to join with. We use the RPN tubelet score defined by Equation 1 to choose the best candidate to link for $\hat{v}_i$. We choose the highest score of all tubelets associated with d^{i,m_i} and $d^{j,1}$ (Algorithm 2:7-8). Then, a cost matrix $\mathcal{C}$ can be constructed with as many rows as ending candidate tubes and as many columns as starting tubes. Each element c_{ij} is the maximum score for tubelets containing d^{i,m_i} and $d^{j,1}$. This becomes an assignment problem that can be easily solved with the *Hungarian Method* (Algorithm 2:10-13). For each tube assignment (i, j), we remove the second tube from the output set (Algorithm 2:12) and build a new tube joining the two original fragments (Algorithm 2:13). Once this process has finished, for a giving tube $\tilde{v}_i$ the score of all detections $d^{i,j} \in \tilde{v}_i$ is set to the mean of the $\alpha = 10\%$ highest scores in that tube (Algorithm 2:15).

4 EXPERIMENTAL RESULTS

4.1 Datasets

We use the ImageNet VID dataset [25] to test our method. It contains 3,862 training and 555 validation videos with annotated objects of 30 different categories. As the test subset annotations are not publicly available and the challenge evaluation server is closed, we use the Average Precision (AP) and Mean AP over the validation set as the main evaluation metrics following the standard approach established by previous works [6] [17] [28] [34].

Following the training procedure proposed by [6], we also use data from the ImageNet DET dataset to enhance the training set. ImageNet DET consists of 456,567 training and 20,121 validation images of 200 different categories that include the 30 object classes used in VID. We add to the training set at most 2,000 images per VID object class from ImageNet DET. This upper bound prevents the bias of the training set for large object categories in DET.

4.2 Implementation details

Single frame. We train our single frame Faster R-CNN baseline network using both ImageNet VID and ImageNet DET following the strategy explained in Section 4.1 for DET. In the single frame case we select 20 uniformly distributed frames over time from each video in the VID training set. The chosen backbone is a Feature Pyramid Network (FPN) based on ResNeXt-101 [30] pretrained on the ImageNet classification dataset.

All input images are scaled so that the smallest dimension is 720 at most. If the highest dimension remains above 1280 pixels, the image is scaled down to prevent that. In any case, the image scaling keeps the original aspect ratio.

We use the SGD learning algorithm with a learning rate set to $2, 5 \times 10^{-4}$ for the first 240K iterations, $2, 5 \times 10^{-5}$ for the next 80K iterations and $2, 5 \times 10^{-6}$ for the last 40K iterations. RPN redundant proposals are suppressed by NMS with a threshold of 0.7, while the final detection set is filtered by means of NMS with an IoU threshold equal to 0.5.

Spatio-temporal. Our spatio-temporal network is initialized with the single frame model keeping all backbone and RPN layers frozen. The training set sampling strategy is slightly different from the single frame baseline. Now, the network needs N consecutive input frames in each iteration instead of just one. To do that, we select 2 groups of input frames with size $N \times 15$, giving 60 training examples per video. Images from ImageNet DET are also included in the training set as described in Section 4.1. The only difference is that each image is converted into a small video with N repeated frames. We have trained our spatial baseline also with this sampling strategy to assess that it does not bias the results analysis. We have not seen any significant differences in the precision values of the test for the two strategies in the single frame case.

We train the spatio-temporal model following the described sampling strategy with a learning rate set to 1.25×10^{-3} for the first 180K iterations, 1.25×10^{-4} for the next 60K iterations and 1.25×10^{-5} for the last 30K iterations. We need to feed the network with N frames in each train iteration, and also for test: the current frame and the $N - 1$ previous ones. To do this, we replicate the first frame of each video $N - 1$ times. Then, the network can work on these replicated frames to process the first $N - 1$ real ones following this rule. The network output detection set is also filtered applying a confidence threshold $\beta = 0.05$ preventing detections with lower scores from degrading the long-term object linking method.

4.3 Ablation studies

We test our framework under different conditions, changing the number of input frames and removing some components to prove how they affect the final detection result.

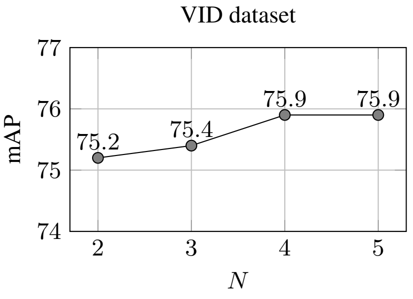

Figure 3. Detection mAP with different tubelet lengths tested on the ImageNet VID validation set without long-term information.

Spatial head Cls	Spatio-temporal head Cls	Long-term object linking	Mean AP
✓			74.3
	✓		74.5
✓	✓		75.9
✓	✓	✓	78.2

Table 1. Influence of each component on the framework precision on ImageNet VID dataset.

Figure 3 shows how the number of input frames N affects the network precision testing on the ImageNet VID validation set. We have tested the network up to a maximum length of 5 frames. Figure 3 shows that to increase N from 4 to 5 has no impact on the network precision. On this basis, we hypothesize that it is very likely that increasing the number of input frames above 5 does not improve or even degrades the network output. This is because of the tubelet initialization, as our *anchor cuboids* regression method expects the same object to be associated with the same *anchor box* in the same position in every input frame. If the object movement exceeds the network receptive field this assumption is not true, which is more likely for large N values. It should also be said that 2 input frames suffice to exceed the single model FPN with ResNetX-101 by 0.5 points; i.e., 75.2% of mAP in Figure 3 vs 74.7% in Table 2.

In order to evaluate how the temporal information helps to enhance the object classification accuracy, we perform two tests on the ImageNet VID dataset: one of them with the current frame only, and the other one with the spatio-temporal information propagated throughout the last N frames for $N = 4$, combining spatial and temporal information by means of Equation 4. Table 1 summarizes the results. Working with information from the current frame is different from just running the spatial network baseline. This is due to the fact that, instead of the original NMS method to remove redundant RPN proposals in the current frame, we resort to T-NMS, which removes redundant tubelets instead of boxes. It can be seen that the network precision with one head branch is lower than both of them combined. This proves that both branches provide complementary information for the classification task. Furthermore, Table 1 also shows how the long-term object linking improves significantly the final Mean AP. This is consistent with previous works with this kind of post processing methods, and reveals that the RPN tubelet information can be valuable to link object instances throughout the video.

4.4 Results

We compare our framework with state-of-the-art spatio-temporal object detectors in Table 2. The method described in [17] has two main components, a tubelet proposal module based on a single frame object detector combined with object tracking. On top of that, it performes a tubelet classification and re-scoring module using a Temporal Convolutional Network (TCN) to achive 47.5% mAP. In [15], the first module is replaced by a Tubelet Proposal Network, and the classification task is performed by an encoder-decoder LSTM achieving 68.4% mAP. T-CNN [16] is the winner of the ILSVRC2015 with 73.8% mAP. This work uses two single frame object detectors, the DeepId-Net [22] and CRAFT [31] frameworks combined by an NMS process. Both detection sets are processed separately applying context suppression and detection propagation using optical flow. Furthermore, it uses tracking algorithms to build object tubelets. A binary classifier identifies positive and negative tubelets re-scoring detections belonging to those tubelets accordingly. Multi-Class Multi-

	airplane	antelope	bear	bicycle	bird	bus	car	cattle	dog	domestic cat	elephant	fox	giant panda	hamster	horse	lion
Kang *et al.* [17]	72.7	75.5	42.2	39.5	25.0	64.1	36.3	51.1	24.4	48.6	65.6	73.9	61.7	82.4	30.8	34.4
Kang *et al.* [15]	84.6	78.1	72.0	67.2	68.0	80.1	54.7	61.2	61.6	78.9	71.6	83.2	78.1	91.5	66.8	21.6
Kang *et al.* [16]	83.7	85.7	84.4	74.5	73.8	75.7	57.1	58.7	72.3	69.2	80.2	83.4	80.5	93.1	84.2	67.8
Lee *et al.* [18]	86.3	83.4	88.2	78.9	65.9	90.6	66.3	81.5	72.1	76.8	82.4	88.9	91.3	89.3	66.5	38.0
Feichtenhofer *et al.* [6]	90.2	82.3	87.9	70.1	73.2	87.7	57.0	80.6	77.3	82.6	83.0	97.8	85.8	96.6	82.1	66.7
Tang *et al.* [28]	90.5	80.1	89.0	75.7	75.5	83.5	64.0	71.4	81.3	92.3	80.0	96.1	87.6	97.8	77.5	73.1
FPN-X101 baseline	91.7	82.3	81.3	68.8	74.8	80.0	62.8	57.4	72.0	80.5	75.9	88.8	88.7	89.9	75.0	54.8
ours (Short-Term)	92.2	83.1	83.0	69.9	75.6	81.0	63.5	61.3	74.3	83.2	77.1	89.7	89.8	91.4	76.1	57.9
ours (Short&Long-Term)	87.8	84.6	88.1	70.3	76.4	85.0	62.5	64.5	79.2	90.4	77.3	91.2	90.3	96.9	79.0	70.7

	lizard	monkey	motorcycle	rabbit	red panda	sheep	snake	squirrel	tiger	train	turtle	watercraft	whale	zebra	Mean AP
Kang *et al.* [17]	54.2	1.6	61.0	36.6	19.7	55.0	38.9	2.6	42.8	54.6	66.1	69.2	26.5	68.6	47.5
Kang *et al.* [15]	74.4	36.6	76.3	51.4	70.6	64.2	61.2	42.3	84.8	78.1	77.2	61.5	66.9	88.5	68.4
Kang *et al.* [16]	80.3	54.8	80.6	63.7	85.7	60.5	72.9	52.7	89.7	81.3	73.7	69.5	33.5	90.2	73.8
Lee *et al.* [18]	77.1	57.3	88.8	78.2	77.7	40.6	50.3	44.3	91.8	78.2	75.1	81.7	63.1	85.2	74.5
Yang *et al.* [32]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	76.2
Feichtenhofer *et al.* [6]	83.4	57.6	86.7	74.2	91.6	59.7	76.4	68.4	92.6	86.1	84.3	69.7	66.3	95.2	79.8
Tang *et al.* [28]	81.5	56.0	85.7	79.9	87.0	68.8	80.7	61.6	91.6	85.5	81.3	73.6	77.4	91.9	80.6
FPN-X101 baseline	78.6	55.2	85.8	66.7	68.6	60.1	59.2	53.8	89.6	84.2	77.2	72.0	75.2	90.5	74.7
Ours (Short-Term)	79.5	56.7	86.2	69.8	72.8	60.9	54.7	56.3	90.2	84.4	78.0	73.4	75.2	91.1	75.9
Ours (Short&Long-Term)	81.5	57.3	89.6	77.5	82.5	63.0	55.9	58.0	90.9	82.8	79.3	73.5	68.3	91.9	78.2

Table 2. VID validation set results. We use a number of input frames $N = 4$ to test our framework.

Object Tracking (MCMOT) [18] achieves 75.5% mAP, combining two different object detectors with multi object tracking (MOT) techniques. The ILSVRC2016 winner [32] comprises a 3 step cascade R-FCN [2] with a correlation tracker and context inference. They are able to improve the accuracy up to 81.2% using multi-scale testing and ensembles. Our system outperforms all previous methods (78.2% mAP) with a single model implementation.

Our network is trained end-to-end without any precomputed proposals such as those used in [6]. That method reuses the proposal set from [33], which implements a two-stage cascade RPN with multiscale testing adding to the proposal set those calculated by the approach addressed in [8]. Also, that method is based on the R-FCN [2] framework using convolutional features to perform object detection and tracking simultaneously. This procedure leads to 79.8% mAP.

Tang *et al.* [28] implement a short tubelet detection framework to identify tubelets with temporal overlapping. Then, given two tubelets, they join those tubelets analyzing the spatial overlap between bounding boxes belonging to each tubelet in the common frame. They perform a multi-scale training and testing to boost the precision to 80.6% mAP. This cannot be directly compared with our results since we only test our system with single scale images, creating a more realistic real-life testing environment.

Finally, we include in Table 2 our single frame baseline using a Feature Pyramid Network with ResNeXt101 getting 74.7% mAP. Our spatio-temporal network outperforms this baseline by 1.2% mAP only taking into account short-term information, i.e., performing the tubelet proposal and the Temporal Pooling method to feed the spatio-temporal head. Table 2 also shows how adding long-term information improves the AP for almost every object class increasing the final mAP by 3.5%.

5 CONCLUSION

We have described a framework that exploits spatio-temporal information to improve object detection precision in videos. We introduce a short-term propagation method that relies on network proposals to link features along time without any other external information, such as associations given by tracking algorithms or optical flow. We also define a spatio-temporal header that takes advantage of both spatial and spatio-temporal features to perform bounding box regression and object classification. We are able to further improve the detection results by combining this short-term information with long-term knowledge with our object linking method. The proposed method achieves competitive results in the widely used ImageNet VID dataset. Moreover, the same ideas can be easily extended to add spatio-temporal information to other single frame frameworks.

ACKNOWLEDGEMENTS

This research was partially funded by the Spanish Ministry of Science, Innovation and Universities under grants TIN2017-84796-C2-1-R, TIN2017-90567-REDT 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] Joao Carreira and Andrew Zisserman, 'Quo vadis, action recognition? a new model and the kinetics dataset', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6299–6308, (2017).

[2] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, 'R-FCN: Object detection via region-based fully convolutional networks', in *Advances in Neural Information Processing Systems (NIPS)*, pp. 379–387, (2016).

[3] Ali Diba, Ali Mohammad Pazandeh, and Luc Van Gool, 'Efficient two-stream motion and appearance 3D CNNs for video classification', *arXiv preprint arXiv:1608.08851*, (2016).

[4] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes, 'Spatiotemporal residual networks for video action recognition', in *Advances in Neural Information Processing Systems (NIPS)*, pp. 3468–3476, (2016).

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, 'Convolutional two-stream network fusion for video action recognition', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941, (2016).

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman, 'Detect to track and track to detect', in *IEEE International Conference on Computer Vision (ICCV)*, pp. 3038–3046, (2017).

[7] Spyros Gidaris and Nikos Komodakis, 'Object detection via a multi-region and semantic segmentation-aware CNN model', in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1134–1142, (2015).

[8] Spyros Gidaris and Nikos Komodakis, 'LocNet: Improving localization accuracy for object detection', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 789–798, (2016).

[9] Ross Girshick, 'Fast R-CNN', in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, (2015).

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, 'Rich feature hierarchies for accurate object detection and semantic segmentation', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, (2014).

[11] Georgia Gkioxari and Jitendra Malik, 'Finding action tubes', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 759–768, (2015).

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, 'Mask R-CNN', in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, (2017).

[13] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid, 'Action tubelet detector for spatio-temporal action localization', in *IEEE International Conference on Computer Vision (ICCV)*, pp. 4405–4413, (2017).

[14] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid, 'Joint learning of object and action detectors', in *IEEE International Conference on Computer Vision (ICCV)*, pp. 4163–4172, (2017).

[15] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang, 'Object detection in videos with tubelet proposal networks', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 727–735, (2017).

[16] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al., 'T-CNN: Tubelets with convolutional neural networks for object detection from videos', *IEEE Transactions on Circuits and Systems for Video Technology*, **28**(10), 2896–2907, (2017).

[17] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang, 'Object detection from video tubelets with convolutional neural networks', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 817–825, (2016).

[18] Byungjae Lee, Enkhbayar Erdenee, Songguo Jin, Mi Young Nam, Young Giu Jung, and Phill Kyu Rhee, 'Multi-class multi-object tracking using changing point detection', in *European Conference on Computer Vision (ECCV)*, pp. 68–83, (2016).

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, 'Feature pyramid networks for object detection', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, (2017).

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, 'Focal loss for dense object detection', in *IEEE International Conference on Computer Vision (CVPR)*, pp. 2980–2988, (2017).

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, 'SSD: Single shot multibox detector', in *European Conference on Computer Vision (ECCV)*, pp. 21–37, (2016).

[22] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, et al., 'DeepID-Net: Deformable deep convolutional neural networks for object detection', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2403–2412, (2015).

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, 'You only look once: Unified, real-time object detection', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, (2016).

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, 'Faster R-CNN: Towards real-time object detection with region proposal networks', in *Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99, (2015).

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, 'ImageNet Large Scale Visual Recognition Challenge', *International Journal of Computer Vision*, **115**(3), 211–252, (2015).

[26] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip Torr, and Fabio Cuzzolin, 'Deep learning for detecting multiple space-time action tubes in videos', in *British Machine Vision Conference (BMVC)*, pp. 58.1–58.13, (2016).

[27] Karen Simonyan and Andrew Zisserman, 'Two-stream convolutional networks for action recognition in videos', in *Advances in Neural Information Processing Systems (NIPS)*, pp. 568–576, (2014).

[28] Peng Tang, Chunyu Wang, Xinggang Wang, Wenyu Liu, Wenjun Zeng, and Jingdong Wang, 'Object detection in videos by high quality object linking', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2019).

[29] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu, 'Rethinking classification and localization in R-CNN', *arXiv preprint arXiv:1904.06493*, (2019).

[30] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, 'Aggregated residual transformations for deep neural networks', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1492–1500, (2017).

[31] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li, 'Craft objects from images', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6043–6051, (2016).

[32] Jing Yang, Hui Shuai, Zhengbo Yu, Rongrong Fan, Qiang Ma, Qingshan Liu, and Jiankang Deng. ILSVRC2016 object detection from video: Team NUIST.

[33] Xingyu Zeng, Wanli Ouyang, Junjie Yan, Hongsheng Li, Tong Xiao, Kun Wang, Yu Liu, Yucong Zhou, Bin Yang, Zhe Wang, et al., 'Crafting GBD-Net for object detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**(9), 2109–2123, (2017).

[34] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei, 'Flow-guided feature aggregation for video object detection', in *IEEE International Conference on Computer Vision (ICCV)*, pp. 408–417, (2017).