A fine-tuning approach based on spatio-temporal features for few-shot video object detection

Daniel Cores[a,*], Lorenzo Seidenari[b], Alberto Del Bimbo[b], Víctor M. Brea[a], Manuel Mucientes[a]

[a] *Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain*
[b] *Media Integration and Communication Center (MICC), University of Florence, Italy*

Abstract

This paper describes a new Fine-Tuning approach for Few-Shot object detection in Videos that exploits spatio-temporal information to boost detection precision. Despite the progress made in the single image domain in recent years, the few-shot video object detection problem remains almost unexplored. A few-shot detector must quickly adapt to a new domain with a limited number of annotations per category. Therefore, it is not possible to include videos in the training set, hindering the spatio-temporal learning process. We propose augmenting each training image with synthetic frames to train the spatio-temporal module of our method. This module employs attention mechanisms to mine relationships between proposals across frames, effectively leveraging spatio-temporal information. A spatio-temporal double head then localizes objects in the current frame while classifying them using

*Corresponding author
Email addresses: daniel.cores@usc.es (Daniel Cores), lorenzo.seidenari@unifi.it (Lorenzo Seidenari), alberto.delbimbo@unifi.it (Alberto Del Bimbo), victor.brea@usc.es (Víctor M. Brea), manuel.mucientes@usc.es (Manuel Mucientes)

both context from nearby frames and information from the current frame. Finally, the predicted scores are fed into a long-term object-linking method that generates object tubes across the video. By optimizing the classification score based on these tubes, our approach ensures spatio-temporal consistency. Classification is the primary challenge in few-shot object detection. Our results show that spatio-temporal information helps to mitigate this issue, paving the way for future research in this direction. FTFSVid achieves 41.9 AP50 on the Few-Shot Video Object Detection (FSVOD-500) and 42.9 AP50 on the Few-Shot YouTube Video (FSYTV-40) dataset, surpassing our spatial baseline by 4.3 and 2.5 points. Additionally, FTFSVid outperforms previous few-shot video object detectors by 3.2 points on FSVOD-500 and 14.5 points on FSYTV-40, setting a new state-of-the-art.
Keywords: few-shot object detection, video object detection, few-shot learning

---

1. Introduction

Object detection aims to localize and classify objects of interest in images. This field has witnessed a significant improvement, mostly due to the adoption of deep learning techniques. Nevertheless, training current state-of-the-art models requires large amounts of labeled data for each object category. This makes the application of these techniques in data-scarce scenarios infeasible. Also, the high annotation costs, even when sufficient unlabeled data is available, might limit the usefulness of these methods in real-world applications. Therefore, the development of few-shot object detection models capable of learning from few annotated examples has become a very active

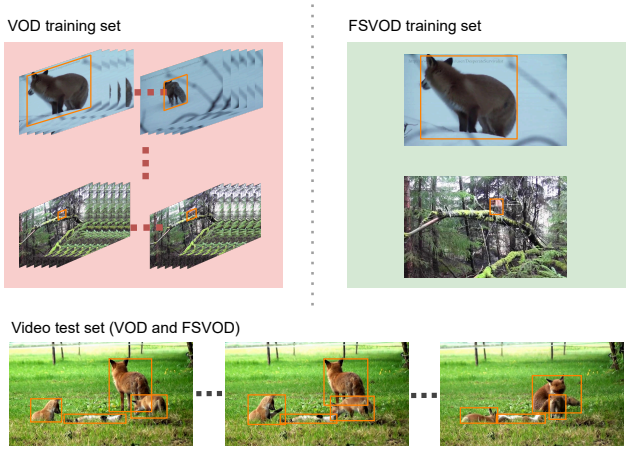


Figure 1: The training set of traditional video object detection (VOD) comprises fully annotated videos. However, in a few-shot video object detection (FSVOD) setting, this training set only contains isolated frames. Both VOD and FSVOD use the same test set, which contains full videos. Thus, a FSVOD algorithm must learn to leverage spatio-temporal relationships among objects in different frames from a single image training set.

research topic.

Traditional single-image object detectors can be adapted to video object detection by processing each video frame independently. However, video detection introduces unique challenges, such as motion blur, out-of-focus frames, occlusions, and significant changes in object appearance that can hinder detection accuracy in some frames. Recent studies [10, 7, 8] show that leveraging spatio-temporal information in videos can improve detection precision and address these issues effectively. Despite the success of traditional video object detection methods, there is still a significant gap in developing few-shot algorithms specifically tailored for videos, leaving this area almost unexplored.

The main challenge in adapting traditional video object detectors to data-

scarce scenarios lies in their dependence on multiple labeled video frames during each training iteration. This requirement violates the few-shot constraints, which restrict the training set to only a few single images per category. Consequently, as shown in Fig. 1, few-shot video object detection demands novel strategies to train models that can effectively extract spatio-temporal information from test videos while being trained solely on single images.

Given these considerations, we propose a few-shot object detector that effectively exploits spatio-temporal information available in videos. The main contributions of this work are:

- A novel training strategy that creates synthetic frames to facilitate learning spatio-temporal features from single images. The synthetic data simulates occurrences of the few annotated objects in different locations, replicating their movement across nearby frames.

- A new spatio-temporal double head that uses both spatial and spatio-temporal information. One branch localizes objects in the current frame and performs spatial classification, while the other uses spatio-temporal data to improve classification and confidence estimation.

- An enhanced long-term object-linking strategy that utilizes short-term spatio-temporal information to associate instances of the same object across the entire video. Long-term confidence optimization is applied to these trajectories, improving the alignment between estimated detection confidence and actual detection quality.

- Our framework outperforms both single image and previous video object detectors in the FSVOD-500 and FSYTV-40 datasets [11], which have been specifically designed for the evaluation of few-shot video object detectors.

2. Related Work

2.1. Object Detection

There are two main families of object detectors: one- and two-stage architectures. Two-stage detectors [23, 35, 18, 34] rely on a proposal generator to calculate a first set of regions with a high probability of containing an object of interest, and then refine this initial set to compute the final detection set. One-stage architectures [29, 36] follow a completely different approach, directly calculating the final detection set without any intermediate stage.

Regarding the video object detection problem, state-of-the-art methods rely on feature aggregation throughout a set of input frames to leverage spatio-temporal information, boosting the precision of single image detectors. Current methods for spatio-temporal feature aggregation fall into two main categories: pixel- and object-level aggregation. Pixel-level methods [1, 13] seek to calculate relationships between all the pixels in the current reference frame and all the supporting neighboring frames to enhance the whole feature map for the reference frame. Alternatively, object-level feature aggregation methods [10, 7, 8] focus on mining relationships between object instances from different frames.

2.2. Few-Shot Object Detection

Traditional object detectors require large manually-labeled training sets, limiting the applicability of these models in real scenarios. Therefore, there is a need for new detection frameworks with lower data requirements in the target domain. Few-shot learning techniques address this problem by extracting general knowledge from base data and rapidly adapting to novel scarce data. Few-shot image classification [15, 32] has been widely studied as the first attempt to apply few-shot techniques in computer vision. The promising results obtained in the image classification field have led to the development of few-shot object detection frameworks, more robust against very limited training sets.

The first attempt to solve the few-shot object detection problem was through meta-learning. Methods based on meta-learning redefine the detection problem as a comparison problem in which a distance metric is learned to differentiate objects from different categories. Then, target objects are compared with the support set, i.e. few annotated samples per object category. Following this trend, FSRW [16] extends YOLOv2 [22] one-stage architecture with a feature reweighting module and a meta-feature extractor. As an alternative, Meta R-CNN [33] proposes a two-stage architecture based on Faster/Mask R-CNN. This method applies attention mechanisms between object proposals and a set of prototypes representing each category of interest to calculate the final bounding box and object category. As a step forward, a multi-relation head is proposed in [12] to combine different similarity metrics. This work also modifies the region proposal network (RPN) to take into account category prototypes in the proposal generation

stage and not only in the network head, as in previous works. Calculating category prototypes general enough to represent each object category while differentiating objects from different categories is a transversal issue for all these methods. DAnA [6] focuses on calculating more robust category prototypes with the introduction of a background attenuation block to reduce the influence of the background and a new summarizing method to combine the information from all annotations of each category.

Fine-tuning methods appear as an alternative to meta-learning frameworks with a completely different approach. TFA [30] starts this new line of research, proving that a simple transfer-learning approach obtains competitive results compared to more complex meta-learning frameworks. In this work, a traditional object detector is trained in base categories with abundant labels while only the last layers of the detector are optimized with novel categories, which only contain a few annotations per category. DeFRCN [21] introduces a gradient decoupled layer (GDL) and a prototypical calibration layer (PCB) to the original Faster RCNN, boosting its performance in data-scarce scenarios. The GDL scales the influence of the different loss components, while the PCB decouples the classification and localization tasks.

As base images might contain unlabeled objects of novel categories, these objects are treated as negatives in the base training. This might hinder the training process, negatively affecting the recall of the model with novel categories. This issue was first addressed in [5], defining a pseudo-labeling strategy to automatically mine annotations of novel categories in base images. The pseudo-labeling approach was also explored in [17] to mine pseudo-annotations in unlabeled images to automatically expand the training set

with new annotations. This work focuses on scenarios in which a large set of unlabeled images is available.

The few-shot video object detection problem remains almost unexplored, with only one first attempt to leverage spatio-temporal information in few-shot settings [11]. This work also proposes for the first time two datasets specifically designed to evaluate the performance of few-shot video object detectors. The object detector defined in [11] includes a tube proposal network and a Tube-based Matching Network to compare tube proposals with the support annotations following a meta-learning approach. Following the current trend in single-image object detection, we propose for the first time a fine-tuning based video object detection framework that leverages spatio-temporal information through an object-level feature aggregation strategy. We prove through experimentation that fine-tuning approaches are more robust than meta-learners.

3. Proposed Method

3.1. Problem Definition

As in previous works [16, 30, 12, 21], the training data comprises a base $\mathcal{D}_{base}$ and a novel $\mathcal{D}_{novel}$ training sets. $\mathcal{D}_{base}$ contains abundant annotations of base categories $\mathcal{C}_{base}$, while $\mathcal{D}_{novel}$ only contains a limited number of annotations for novel categories $\mathcal{C}_{novel}$. There must be no overlap between novel and base categories, $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$. The number of annotations per category in $\mathcal{D}_{novel}$ —shot size— is typically set in the literature to a value between 1 and 30. As the proposed framework follows a fine-tuning approach, the training process is organized into two general stages. First, the network is

trained in $\mathcal{D}_{base}$. Then, it is fine-tuned in $\mathcal{D}_{novel}$.

To evaluate our video object detector, we keep the same definition of $\mathcal{D}_{novel}$ as for single-image few-shot object detection evaluation: it contains K annotated object instances in isolated video frames. Annotating object trajectories throughout the video would increase the annotation effort and data availability requirements, setting a completely different scenario than the single image domain. This allows a fair comparison between single image baselines and spatio-temporal detectors, proving that the benefits of exploiting spatio-temporal information do not come from having larger training sets.

The video object detection problem focuses on boosting the precision of the detection in each frame f_t by considering information from a set of input frames, in our case N previous frames $f_{t-N}, ..., f_{t-1}$. We also consider the whole video to perform a long-term confidence score optimization to ensure spatio-temporal consistency throughout the entire video.

3.2. FTFSVid architecture

This paper introduces a new few-shot object detection framework that leverages spatio-temporal information available in videos to improve the quality of the final detection set. The proposed spatio-temporal feature aggregation method is added to a strong spatial baseline [21] that modifies the original Faster RCNN architecture to improve its performance in low data availability scenarios.

The proposed method is depicted in Fig. 2. First, object proposals are generated for each input frame, only considering spatial information. These proposals are calculated using a Region Proposal Network (RPN), which

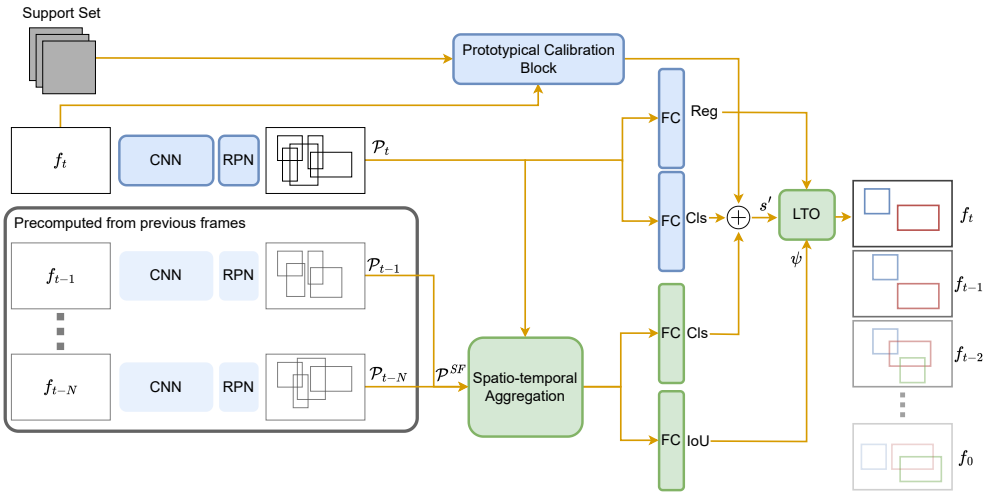


Figure 2: Overview of the FTFSVid architecture: The blue components handle spatial information, while the green components incorporate spatio-temporal information. Proposals from previous frames are reused, so only the proposals for the new frame need to be computed.

takes as input the feature maps extracted by a ResNet-101 backbone. Then, a spatio-temporal aggregation module calculates relationships between object proposals from the current frame f_t, and support proposals $\mathcal{P}^{SF}$, extracted from $f_{t-N}, ..., f_{t-1}, f_t$. This aggregation process is detailed in Sec. 3.4.

The localization task is exclusively performed using information from the current frame, as this provides the most accurate data for object positioning, which can change from frame to frame. In contrast, the object classification task combines both spatial and spatio-temporal information to improve single-image classification accuracy. By incorporating multiple input frames, this approach addresses challenges such as motion blur, occlusions, and sudden changes in object appearance that can hinder classification. Additionally, the spatio-temporal branch predicts a class-agnostic confidence score, which aids in guiding long-term object tracking and linking. Spatial and spatio-temporal classification scores are combined as follows:

$$s = s_{tmp} + s_{spt}(1 - s_{tmp}) \tag{1}$$

s_{spt} being the spatial classification score and s_{tmp} the spatio-temporal score.

Our temporal branch is implemented as a multilayer perceptron (MLP) with two hidden layers and two fully connected output layers: one for temporal classification and the other for class-agnostic confidence score prediction.

We keep the main building blocks of DeFRCN [21] in our implementation, including the Prototypical Calibration Block (PCB). The main goal of this component is to decouple the localization and classification tasks at inference time. A more robust classification score is achieved by modifying the original classification score according to a similarity metric between each

detection and the K annotated objects from the predicted category. First, a per-category prototype is computed by averaging the feature maps of each annotated object for the corresponding category. These feature maps are the result of applying RoI Align over the features extracted for the whole image using a CNN trained on the ImageNet dataset. Features for each detection are also calculated following the same strategy. Finally, the classification score is computed as:

$$s' = \beta \cdot s + (1 - \beta) \cdot s^{\cos} \tag{2}$$

$s^{\cos}$ being the similarity between the detection and the corresponding category prototype calculated as a cosine distance between the two feature maps. The hyperparameter β modulates the influence of this component over the original classification score.

The last step of the proposed architecture is a long-term confidence score optimization method (LTO) that modifies the confidence score of each detection based on the spatio-temporal consistency and the category agnostic score calculated by the spatio-temporal double head. First, we create a set of object tubes linking object detections across the entire input video, and then we perform the score optimization over the generated object tubes. This method is described in Sec. 3.6.

3.3. Single image spatio-temporal training

The training process of common video object detectors usually implies a random selection of support frames for each input frame in each training iteration [10, 7, 8]. This requires to work with videos in the training set, making it impossible to train these models with single images. Therefore,

these algorithms are not suitable for few-shot scenarios in which there are a few images for each object category. Thus, a few-shot video object detector must learn from a limited set of training images, while considering spatio-temporal information at test time.

Therefore, we propose to generate synthetic support frames in each training iteration. The synthetic frame generation process involves introducing variations of objects from the input training image, repositioning them within the same background to simulate movement and slight changes in appearance as seen in consecutive frames of real videos. It is important to note that the generated frames should only capture the subtle variations in object appearance between nearby frames. The goal of this method is to create synthetic data that enables the relation module to effectively leverage short-term temporal information.

Algorithm 1 describes the process of generating a set of synthetic support frames $\mathcal{F}$. The goal of this method is to generate Q synthetic support frames f_q for each training image I_t by inserting transformations of objects from I_t in different positions of f_q. First, f_q is initialized as I_t (Alg. 1:3). Then, each annotated object $\alpha_{t(j)}$ from I_t is randomly transformed and inserted L times in f_q. The number of insertions depends on the size of the object relative to the image size, with a maximum number of insertions of γ (Alg. 1:7). The random transformation operator applies random horizontal flipping over the original object (Alg. 1:9). A random position is also generated to insert the object in f_q (Alg. 1:10), ensuring that the new object fits inside the image boundaries. Directly inserting the cropped object in a different position generates undesirable artifacts that might damage image features, hindering the

Algorithm 1: Synthetic support frames generation

Input : Single image from the training set: I_t

Input : Object annotations: $\mathcal{A}_t = \{\alpha_{t(j)}\}_{j=1}^{\eta_t}$

Output : Generated synthetic frames: $\mathcal{F}$

1 $\mathcal{F} \leftarrow \emptyset$

2 **for** q **in** $1, ..., Q$ **do**

3 $\quad f_q \leftarrow I_t$

4 $\quad$ **for** j **in** $1, ..., \eta_t$ **do**

5 $\quad\quad L_x \leftarrow \lceil width(I_t)/width(\alpha_{t(j)}\}) \rceil$

6 $\quad\quad L_y \leftarrow \lceil height(I_t)/height(\alpha_{t(j)}\}) \rceil$

7 $\quad\quad L \leftarrow min(L_x, L_y, \gamma)$

8 $\quad\quad$ **for** l **in** $1, ..., L$ **do**

9 $\quad\quad\quad \hat{\alpha}_l \leftarrow \mathcal{T}(\alpha_{t(j)})$

10 $\quad\quad\quad (x, y) \leftarrow \mathcal{P}(f_q)$

11 $\quad\quad\quad f_q \leftarrow \mathcal{C}(f_q, \hat{\alpha}_l, (x, y))$

12 $\quad \mathcal{F} \leftarrow \mathcal{F} \cup \{f_q\}$

13 **return** $\mathcal{F} = \{f_q\}_{q=1}^{Q}$

learning process. Therefore, each transformed object $\hat{\alpha}_l$ is inserted in the position (x, y) in f_q by a seamless cloning operator [20] (Alg. 1:11). This method can insert an object from a source image into a target image, removing undesirable artifacts. It does not require a segmentation mask, which allows defining the boundaries of objects in the source image based on the annotated bounding boxes.

Previous work on synthetically generated data for object detection [3]

has emphasized the importance of maintaining consistency between the foreground and background, as object detectors also rely on background features. These works highlight the need for selecting plausible backgrounds when inserting new objects and propose robust position selection methods. However, these approaches are unsuitable for data-scarce environments because they require larger training sets. In contrast, our implementation introduces a training-free generation method that reuses the background of the current frame, leveraging the high spatio-temporal redundancy of nearby frames in real videos. We slightly modify and reposition objects from the current frame within the same scene. Our experimental results demonstrate that this approach is generalizable, achieving performance comparable to models trained with additional real data.

3.4. Spatio-temporal relation head

Attention mechanisms were successfully applied to mine relationships among objects on the same image [14], and also among object proposals in different frames of the same video [14, 10, 8]. The core idea of these methods is the multi-head attention model defined in [28] applied in the natural language processing field.

The relation module calculates M relation features $\mathbf{r}^m$ between each object proposal $p_{t(i)} \in \mathcal{P}_t$ and support proposals in $\mathcal{P}^{SF}$:

$$\mathbf{r}^m(p_{t(i)}, \mathcal{P}^{SF}) = \sum_{r=1}^{R} \sum_{j=1}^{|\mathcal{P}_r|} w^m_{t(i),r(j)} \left(W_V \, \phi(p_{r(j)})\right), \qquad m = 1, ..., M \tag{3}$$

where $\mathcal{P}^{SF} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_R\}$ is the set of all proposals calculated in the sup-

port frames. As the support frames are synthetically generated at training time, $\mathcal{P}^{SF}$ contains proposals extracted from the synthetic set $\mathcal{F}$, generated following Algorithm 1. At test time, $\mathcal{P}^{SF}$ contains proposals calculated in previous real frames. Each proposal is defined by its appearance features $\phi(p_{t(i)})$ and geometry features $\mathbf{b}(p_{t(i)})$. The relation weight $w^m_{t(i),r(j)}$ is computed as a pairwise attention between each proposal $p_{t(i)}$ in the current frame t, and each support proposal $p_{r(j)}$ in SF. The linear transformation W_V is a learnable component.

The definition of the pairwise relational weight $w^m_{t(i),r(j)}$ is:

$$w^m_{t(i),r(j)} = \frac{g^m_{t(i),r(j)}\, exp(a^m_{t(i),r(j)})}{\sum_q g^m_{t(i),q}\, exp(a^m_{t(i),q})} \tag{4}$$

$a^m_{t(i),r(j)}$ being the pairwise appearance similarity weight and $g^m_{t(i),r(j)}$ the geometry similarity weight.

The appearance similarity weight is as a normalized dot product:

$$a^m_{t(i),r(j)} = \frac{\langle W_H\, \phi(p_{t(i)}), W_Q\, \phi(p_{r(j)})\rangle}{\sqrt{d_h}} \tag{5}$$

where W_H and W_Q are learnable parameters that project original appearance features to a subspace to measure their similarity, d_H being the feature dimension after the projection.

The geometry similarity weight is calculated as:

$$g^m_{t(i),r(j)} = max\{0, W_G\, \mathcal{E}(\mathbf{b}(p_{t(i)}), \mathbf{b}(p_{r(j)}))\} \tag{6}$$

where W_H is also a learnable parameter. Geometry features containing the parameters of the bounding box (x, y, w, h) are encoded as:

$$\left(log\left(\frac{|x_i - x_j|}{w_i}\right), log\left(\frac{|y_i - y_j|}{h_i}\right), log\left(\frac{w_j}{w_i}\right), log\left(\frac{h_j}{h_i}\right)\right).$$

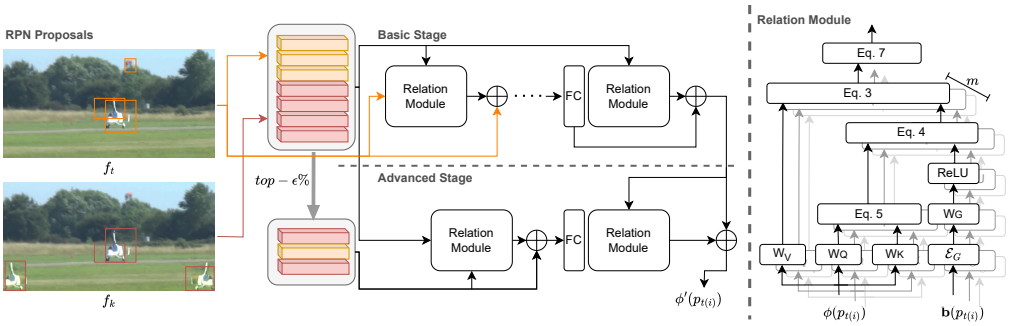


Figure 3: Spatio-temporal aggregation. As the training set only contains single images, this component is fed with the current reference frame f_t and a set of synthetically generated support frames —f_k in the image— to mine object relationships between different frames at training time. At test time, it receives the current frame and a set of previous frames from the test video.

Function $\mathcal{E}$ represents an embedding calculation into a high-dimensional representation following [28].

The final feature map for proposal $p_{t(i)}$ is enhanced by adding the concatenation of the M relational features $\mathbf{r}^m(p_{t(i)}, \mathcal{P}^{SF})$ (Eq. 3):

$$\phi'(p_{t(i)}) = \phi(p_{t(i)}) + concat[\{\mathbf{r}^m(p_{t(i)}, \mathcal{P}^{SF})\}_{m=1}^{M}] \tag{7}$$

Previous work [10, 7, 8] has proven that a simple relation module does not suffice to model object relationships between different video frames. Thus, a multi-stage approach is needed to extract useful relation features. Fig. 3 illustrates the proposed multi-stage spatio-temporal aggregation module. First, in the basic stage, relational features between supporting proposals in $\mathcal{P}^{SF}$ and proposals in the current frame $\mathcal{P}_t$ are used to enhance proposals in the current frame. This stage involves a sequence of relational modules with residual connections that iteratively enhance proposal features in the current

frame. The advanced stage consists of a two-step relation distillation. In the first step, proposals in $\mathcal{P}^{SF}$ are enhanced by calculating their relationships with top-ϵ% proposals in $\mathcal{P}^{SF}$. Then, the resulting proposal set of the first step and the output of the basic stage are fed to a new relation module that generates the final feature maps $\phi'(p_{t(i)})$.

3.5. Class-agnostic confidence score

The proposed network head follows a two branch architecture, one branch localizes and classifies the object based on spatial features, while the spatio-temporal branch classifies the object and predicts a category-agnostic confidence score. Thus, setting a double objective optimization loss function for both branches. Regarding the spatio-temporal branch, the classification loss function follows the standard cross-entropy loss for multi-class classifiers, and the category agnostic score is optimized with a binary cross-entropy loss. The objective is to predict the overlap of each detection with the ground truth, regardless of its category. Therefore, the final spatio-temporal loss is computed as:

$$\mathcal{L} = \mathcal{L}_{CLS} + \mathcal{L}_{IoU} \tag{8}$$

In this implementation, the overlap is calculated as the intersection over union (IoU). The following equation describes the target generation process for training:

$$\psi(p_{i(j)}, I_t) = \max_{\forall \alpha_{i(k)} \in I_t} IoU(p_{i(k)}, \alpha_{i(j)}) \tag{9}$$

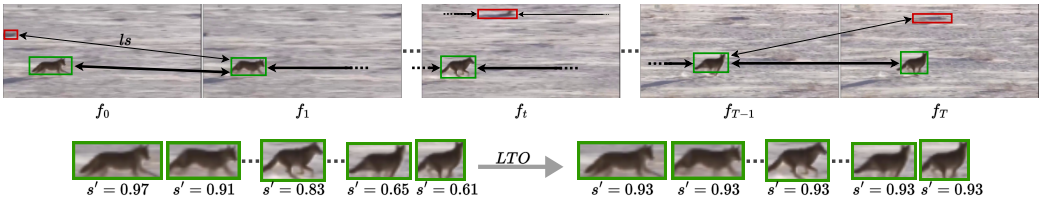


Figure 4: Long-term optimization. First, final detections are linked, calculating a linking score —ls– between consecutive frames, generating long tubes. Then, the confidence scores of detections belonging to each tube are updated. As isolated detections —red boxes— cannot be linked to any large tube, their confidence score remains unchanged.

This equation calculates the maximum overlap between each object proposal $p_{i(k)}$ in the current training image I_t and every annotation $\alpha_{i(j)}$ in I_t.

Previous work on the use of the IoU to increase the correlation of the confidence score with the actual detection precision has shown the effectiveness of binary cross-entropy over other common regression loss function alternatives such as L2 loss [31]. Therefore, our $\mathcal{L}_{IoU}$ is defined as:

$$\begin{aligned} \mathcal{L}_{IoU} = \underset{n=1,\ldots,B}{\text{mean}} [\psi(p_{i(j)}, I_t) \cdot \log \hat{\psi}(p_{i(j)}) + \\ (1 - \psi(p_{i(j)}, I_t)) \cdot \log(1 - \hat{\psi}(p_{i(j)}))] \end{aligned} \tag{10}$$

$\hat{\psi}(p_{i(j)})$ being the predicted overlap for a proposal $p_{i(j)}$ and $\psi(p_{i(j)}, I_t)$ the actual maximum overlap with the ground truth defined in Eq. 9.

This loss function is used to train the model to predict the overlap of each object proposal with the ground truth in the current frame. Experimental results show that, simply including the IoU-aware loss function, increases the detection precision (Sec. 4.3).

3.6. Long-term Optimization (LTO)

Most state-of-the-art object detectors exclusively rely on the classification score to calculate the final detection confidence. A strong confidence prediction is crucial, as the common practice for removing redundant proposals consists of applying a Non-Maximum Suppression (NMS) operator that removes spatially redundant detections with lower confidence. Also, a lower confidence threshold is usually imposed to reduce false positives.

In order to get more accurate confidence scores for the final detection set, we apply a long-term optimization to ensure spatio-temporal consistency. This technique has been successfully applied in both object detection [10, 4, 9] and action recognition [26]. The goal of the long-term confidence score optimization (LTO) is to link detections in consecutive frames, building long object tubes. Then, the confidence score is updated for all detections in each tube. The link score ls between a detection $d_{t(i)}$ in frame f_t and a detection $d_{t'(j)}$ in $f_{t'}$ is calculated as:

$$ls(d_{t(i)}, d_{t'(j)}) = \hat{\psi}(d_{t(i)}) + \hat{\psi}(d_{t'(j)}) + 2 \cdot IoU(d_{t(i)}, d_{t'(j)}). \tag{11}$$

$\hat{\psi}(d_{t(i)})$ being the overlap predicted following the method described in Sec. 3.5. Thus, detections with higher predicted overlap with the ground truth are more likely to be linked. As the output of $\hat{\psi}(d_{t(i)})$ is between 0 and 1 the factor 2 ensures that the network confidence and the overlap between $d_{t(i)}$ and $d_{t'(j)}$ have the same influence on the final link score.

Each object tube $\hat{v}$ from all possible tubes $\mathcal{V}$ is calculated by iteratively

solving the following equation:

$$\hat{v} = \arg\max_{\mathcal{V}} \sum_{t=2}^{T} ls(D_{t-1}, D_t) \tag{12}$$

where D_t is the detection set in frame f_t. Once an object tube $\hat{v}$ is calculated, all detections belonging to $\hat{v}$ are removed from the candidate detection set to build a new object tube. The output of ls is a matrix with the linking score between each detection in D_{t-1} and each detection in D_t. Eq. 12 can be solved by applying the Viterbi algorithm. Then, following previous methods [10, 7, 8], detections belonging to each tube $\hat{v}$ are updated, setting their classification score to the mean classification score of the top-20% in $\hat{v}$. Fig. 4 shows the effect of this optimization, generally increasing the confidence score of detections that are spatio-temporally consistent.

4. Experiments

4.1. Experimental settings

We evaluate our model on FSVOD-500 and FSYTV-40, first proposed in [11]. These datasets define new data partitions suitable for few-shot evaluation, reusing images and annotations from previous large-scale video datasets. This new partitioning ensures that $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$, while keeping a high diversity among the different categories.

FSVOD-500 contains 2,553 annotated videos with 320 different object categories for $\mathcal{D}_{base}$, and 949 videos with 100 object categories for $\mathcal{D}_{novel}$ [1].

[1] FSVOD500 also contains a validation set with 770 annotated videos with 80 object categories. We do not use this set in the experiments reported in this section.

Annotations in FSVOD-500 are provided at 1fps, thus there are no annotations available for every frame in the video. Following the few-shot setting, object categories from these splits are completely different: base categories $\mathcal{C}_{base}$ mainly contain common categories, while novel categories $\mathcal{C}_{novel}$ are rare categories for which available data is scarce. This simulates a real-world scenario with few annotations for target categories, but in which public datasets can be used as $\mathcal{D}_{base}$.

We also compare our method with the state of the art on the FSYTV-40 dataset. It sets a completely different scenario than FSVOD-500 with significantly fewer object categories and more annotations per category. This dataset defines 30 object categories for $\mathcal{D}_{base}$ with 1,627 videos and 10 categories for $\mathcal{D}_{novel}$ with 608 videos.

Following [11], instead of defining a unique global training set for fine-tuning ($\mathcal{D}^{train}_{novel}$) and a testing set with the remaining videos ($\mathcal{D}^{test}_{novel}$), we select multiple fine-tuning sets, so that each video in $\mathcal{D}_{novel}$ is picked once for test. For this, in our implementation $\mathcal{D}_{novel}$ is randomly divided into two subsets, keeping the same distribution of videos per object category. Thus, one subset is used as $\mathcal{D}^{test}_{novel}$, while $\mathcal{D}^{train}_{novel}$ is selected from the other subset, picking K random object annotations per category. Then, the subsets are interchanged, so each video is in $\mathcal{D}^{test}_{novel}$ once. We repeat this whole process 5 times —with different random splits—, and the reported results include the mean and standard deviation of these 5 executions. The same 5 randomly generated $\mathcal{D}^{train}_{novel}$ are used in all the experiments performed in this work for a fairer comparison, avoiding the effect of selecting different fine-tuning sets for different models. The number of annotations per category K is set to 5 by default

in all experiments to facilitate the comparison with previous methods. An analysis on the influence of K is also provided.

Regarding the evaluation metric, we report the average precision on novel categories setting an IoU threshold of 0.5 (nAP50), i.e. an object detection is considered correct if the IoU with a ground truth annotation of the same category is greater than 0.5.

4.2. Implementation details

The feature extractor backbone for all experiments is a ResNet-101 pretrained on ImageNet [25]. First, we train the spatial part of FTFSVid on $\mathcal{D}_{base}$. Then, we fine-tune both the spatial and spatio-temporal parts of FTFSVid on $\mathcal{D}_{novel}^{train}$ —for the spatial part, we start with the pretrained weights on $\mathcal{D}_{base}$, while the spatio-temporal weights are randomly initialized.

For the baseline training on $\mathcal{D}_{base}$, we set the batch size to 16 and the base learning rate to 2×10^{-2} for the first 20K iterations, reducing it to 2×10^{-3} for the next 5K iterations, and to 2×10^{-4} for the last 5K iterations. The baseline fine-tuning on $\mathcal{D}_{novel}^{train}$ is also performed with 16 images per batch with a learning rate set to 1×10^{-2} for the first 9K iterations, reducing it to 1×10^{-3} for the last 1K iterations. The spatio-temporal training is performed for 40K iterations with 1 image per batch and an initial learning rate of 2.5×10^{-4}, reducing it to 2.5×10^{-5} after the first 30K iterations. The classification loss function is implemented as cross-entropy with label smoothing regularization [27].

Input images are resized with the shortest dimension randomly set to (640, 672, 704, 736, 768, 800) pixels for training and 800 pixels for testing. For training, 2 synthetic support frames are generated for each input image

Figure 5: Examples of real reference frames and their corresponding synthetic support frames generated following Alg. 1. The synthetic support frames might appear flipped as the network data augmentation includes horizontal flipping.

with a maximum number of $\gamma = 5$ new objects inserted into each synthetic support frame. For training, 2 synthetic support frames are generated for each input image with a maximum number of $\gamma = 5$ new objects inserted into each synthetic support frame. Empirical results indicate that increasing the number of synthetic frames beyond two has minimal impact on model performance. At testing, 15 support frames are considered for each reference frame to mine object relations. The hyperparameter β that modulates the influence of the Prototypical Calibration Block on the classification score is set to 0.5. For the relation module, the ratio of proposals selected for the advanced stage is $\epsilon = 20\%$. We keep the same hyperparameters for both datasets.

4.3. Ablation Studies

We have conducted a series of ablation studies to assess the performance of each component of the detection framework. In addition to the final AP, we also provide an error type analysis to evaluate how the proposed model improves the baseline results.

STA	IoU loss	LTO	nAP_{50}
-	-	-	$37.6_{\pm 0.5}$
✓	-	-	$38.6_{\pm 0.4} \uparrow 1.0$
✓	✓	-	$38.8_{\pm 0.7} \uparrow 1.2$
✓	✓	✓	$41.9_{\pm 2.0} \uparrow 4.3$

Table 1: Ablation Studies for FTFSVid.

Tab. 1 provides a detailed comparison of each stage of our proposed framework against the spatial baseline, shown in the first row. In the second row, we see that incorporating the Spatio-Temporal Aggregation (STA) module leads to a 1.0 point increase in nAP_{50}. In this setting, the model is optimized using only the classification loss for the spatio-temporal branch. The third row introduces the IoU loss, which yields a slight performance gain, despite not yet utilizing the predicted IoU at test time to refine the detection results. We argue that the inclusion of the more complex loss function helps to alleviate overfitting, especially given the scarcity of data. Finally, when the predicted IoU is incorporated via the Long-Term Optimization (LTO) module, the nAP_{50} improves by an additional 3.1 points, resulting in a total improvement of 4.3 nAP_{50} points over the baseline.

Fig. 5 provides a qualitative evaluation of the synthetic support frames used for spatio-temporal training. To evaluate the impact of this approach on AP_{50}, we repeated the experiment described in the third row of Tab. 1, but using real data. Instead of generating synthetic support frames, real frames were randomly selected from the original videos corresponding to

the images in $\mathcal{D}_{novel}^{train}$. This method, exceeds the K-sample limit per object category, violating the constraints of few-shot learning by introducing extra data. In this scenario —which does not fulfill the few-shot constraints—, the detection framework achieved 38.9 AP_{50}, which is only 0.1 points higher than the results in Tab. 1. This marginal difference shows that training the spatio-temporal components with synthetic data yields nearly identical performance to using complete video data, which is impossible within the constraints of few-shot learning.

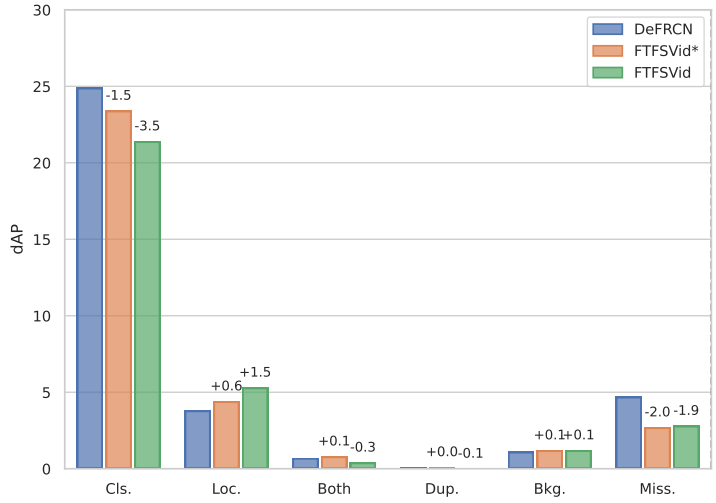


Figure 6: Error type analysis for the single image baseline (DeFRCN), FTFSVid and FTFSVid without LTO (FTFSVid*). For five random training sets, we report average dAP [2] —, the lower the better. Labels represent difference with baseline (DeFRCN)

We have performed an error type analysis based on the TIDE toolbox [2] to identify the source of errors. This framework defines six different error types: classification (Cls.), localization (Loc.), localization and classification (Both), duplicated detections (Dup.), background detected as object of in-

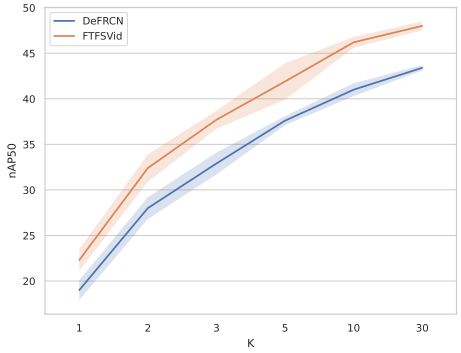


Figure 7: Influence of the shot size (K). Mean AP and standard deviation for each value of K.

terest (Bkg.) and missing objects (Miss.). Moreover, it also analyzes false positive and false negative errors. The proposed methodology consists of defining oracles that solve each error type independently and report the difference in AP (dAP). Fig. 6 shows the comparison between the single image baseline (DeFRCN), FTFSVid and FTFSVid without LTO (FTFSVid* in the figure).

As expected in a few-shot scenario, the most prominent error type is the classification error. That motivates the need to extract information from multiple frames in video sequences. We argue that combining spatial information from multiple input frames might not contribute to localize the object in the current frame, as the most valuable spatial information to localize each object proposal must come from the current frame f_t. However, this spatio-temporal information might be crucial for object classification, in order to overcome issues such as motion blur or occlusions that have a greater impact on the classification task. Our spatio-temporal method (21.4 dAP) improves the single image baseline (24.9 dAP) by 3.5 dAP in classification error.

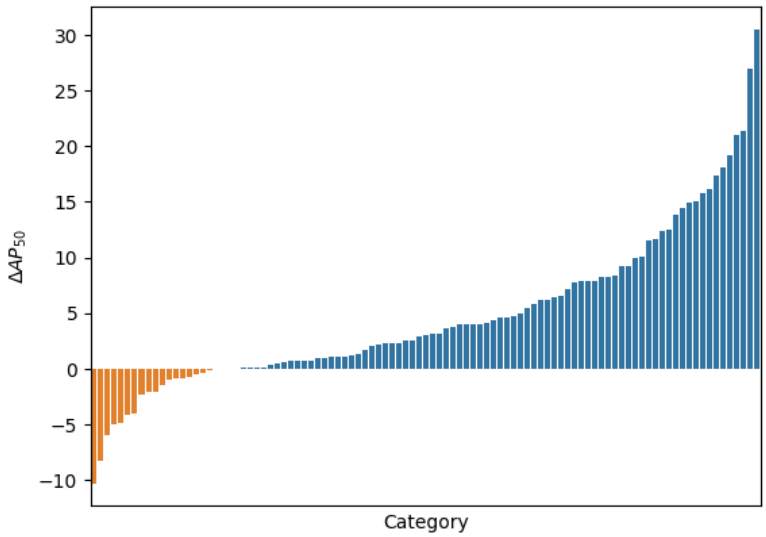


Figure 8: Performance improvement of FTFSVid over the spatial baseline for each novel category in FSVOD-500. Blue bars indicate categories where spatio-temporal information enhances performance, while orange bars represent categories for which spatio-temporal information decreases performance compared to the spatial baseline.

Fig. 7 analyzes the influence of the shot size (K) for both DeFRCN and FTFSVid. Our approach outperforms the spatial baseline in all the cases, from 1 annotated example per category up to 30 annotated examples per category. As expected, the results also show an improvement as the number of annotations in $\mathcal{D}_{novel}^{train}$ increases. DeFRCN improves 18.6 AP_{50} by increasing K from 1 to 5 while FTFSVid improves 19.6 AP_{50}. The maximum AP_{50} for FTFSVid is 48.0, outperforming DeFRCN by 4.6 points (43.4 AP_{50}) and achieving a total improvement —from 1-shot to 30-shot— of 25.7 points, which is 1.3 points higher than DeFRCN. Thus, our method performs better in very limited data availability scenarios, and also exploits better new annotations with a higher increment in the final nAP_{50}.

Fig. 8 reports the nAP improvement of FTFSVid over the spatial baseline (DeFRCN) across the 100 novel categories in FSVOD-500. FTFSVid increases AP_{50} in 81 out of 100 categories, with declines greater than 5 points in only three categories. The largest decline occurs in the 'Horseshoe crab' category, where annotation sizes are six times larger than the average, indicating an out-of-distribution category in terms of object size that impacts FTFSVid. Overall, FTFSVid shows strong performance across a wide range of categories, achieving improvements of up to 30 AP50 points in the 'Gemsbok' category.

4.4. Experimental Results

Tab. 2 shows the comparison with state-of-the-art methods for the FSVOD-500 and FSYTV-40. As the few-shot video object detection problem remains almost unexplored with only one previous work, for comparison purposes we also include single image few-shot object detectors, traditional video object detectors and methods based on multiple object tracking (MOT). The results for traditional video object detectors and MOT-based methods were originally reported in [11]. In those experiments, authors extracted class-agnostic object tubes and applied a meta-learner classifier to calculate the final detection set. We extended this experiments including our single image baseline —DeFRCN[21]—, and a transformer-based single image object detector, ViTDet [18].

FTFSVid outperforms previous approaches, improving our single image baseline (DeFRCN) by 4.3 points and previous few-shot video object detectors by 3.2 points in the FSVOD-500 dataset. Traditional video object detectors that include attention mechanisms for mining proposal relationships

Table 2: Results on FSVOD-500 and FSYTV-40. The comparison includes single-image object detectors, traditional video object detectors, multiple object trackers + meta-learning classifiers, and few-shot video object detectors.

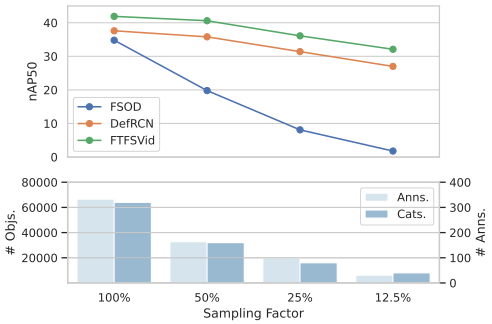
Type	Method	FSVOD-500 nAP_{50}	FSYTV-40 nAP_{50}
Obj. Det.	Faster R-CNN [23]	$26.4_{\pm 0.4}$	$15.4_{\pm 1.7}$
	ViTDet [18]	$36.4_{\pm 0.9}$	$41.5_{\pm 1.5}$
Few-shot Obj. Det.	TFA [30]	$31.0_{\pm 0.8}$	$20.8_{\pm 1.6}$
	FSOD [12]	$31.3_{\pm 0.5}$	$20.9_{\pm 1.8}$
	DeFRCN [21]	$37.6_{\pm 0.5}$	$40.4_{\pm 1.8}$
Vid. Obj. Det.	MEGA [7]	$26.4_{\pm 0.5}$	$13.0_{\pm 1.9}$
	RDN [10]	$27.9_{\pm 0.4}$	$13.4_{\pm 2.0}$
Mult. Obj. Track.	CTracker [19]	$30.6_{\pm 0.7}$	$14.4_{\pm 2.5}$
	FairMOT [37]	$31.0_{\pm 1.0}$	$16.0_{\pm 2.2}$
	CenterTrack [38]	$30.5_{\pm 0.9}$	$15.6_{\pm 2.0}$
Few-shot Vid. Obj. Det.	FSVOD [11]	$38.7_{\pm 0.7}$	$28.4_{\pm 1.2}$
	FTFSVid	$\mathbf{41.9_{\pm 2.0}}$	$\mathbf{42.9_{\pm 2.3}}$

[10, 7] fail to perform few-shot object detection, falling behind single-image few-shot detectors. This proves the need for algorithms specifically designed for few-shot video object detection.

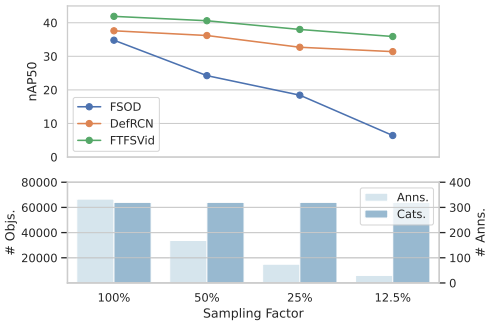
Regarding the FSYTV-40 dataset, our method outperforms the single image baseline by 2.5 points and clearly outperforms previous methods by 14.5 points. Our method is the first fine-tuning based few-shot object detector specifically designed for videos, in contrast with previous methods based on meta-learning. This makes our framework more robust against different training sets, specifically when the number of categories in $\mathcal{D}_{base}$ is lower. This lack of diversity among categories in $\mathcal{D}_{base}$ hinders the learning process of meta-learners.

We also evaluate whether the differences are statistically significant through the Wilcoxon Signed Rank Test on paired samples over multiple runs of DeFRCN and FTFSVid. We have used the STAC platform [24] to perform this analysis. We obtain a probability of having statistically significant differences ($1 - $ p-value) of 95.7% for FSVOD-500 and FSYTV40. We cannot replicate the same analysis for comparing our method with previous state-of-the-art approaches, as the authors only report the aggregated information. In summary, we have set a strong baseline that achieves competitive results compared to previous methods, and we can conclude that FTFSVid outperforms this baseline and the differences are statistically significant.

We further analyze the impact of the number of categories and annotations in $\mathcal{D}_{base}$ (Fig.9). To do this, we create several subsets of the original base categories from the FSVOD-500 dataset. We then compare the effect of reducing the number of base categories on our method (FTFSVid) and



(a) Reducing the number of categories and, consequently, the total number of annotations in $\mathcal{D}_{base}$ by a sampling factor.



(b) Reducing the number of annotations in $\mathcal{D}_{base}$ keeping a constant number of categories.

Figure 9: Experiments on the FSVOD-500 dataset with a subset of $\mathcal{D}_{base}$ reducing the number of categories or the number of annotations per category.

FSOD [11], the best-performing publicly available meta-learner. As shown in Fig. 9a, the performance drop is more severe for the meta-learner, with a decline of around 15 points when reducing the number of categories from 320 to 160, while FTFSVid shows minimal changes. Furthermore, when reducing the number of categories by a factor of 8 —from 320 to 40—, the meta-learner becomes ineffective, whereas our method continues to produce reasonably strong results. To isolate the impact of the number of categories from the reduction in training samples, we performed a complementary experiment (Fig. 9b). In this case, we fixed the number of base categories to 320 but reduced the number of videos per category by the same factor as in Fig. 9a. This ensured that the overall size of $\mathcal{D}_{base}$ remained comparable. The results reveal that reducing the number of categories has a greater negative effect on performance than reducing the number of objects. For example, when applying a reduction factor of 4, the $nAP50$ of FSOD is approximately

10 points higher when the number of categories is kept constant. This indicates that meta-learner performance is highly sensitive to the diversity of $\mathcal{D}_{base}$, relying more on category variety.

5. Conclusions

We have proposed FTFSVid, a new few-shot video object detection framework that, first, applies attention mechanisms to mine proposal relationships between different frames. Then, a spatio-temporal double head classifies object proposals leveraging spatio-temporal information, and it also predicts the overlap of each proposal with the ground truth. Finally, overlapped predictions are used in an object linking method to create long tubes and optimize classification scores. Moreover, we have defined a new training strategy to learn from single images while considering a group of input frames at inference time.

Our proposal presents the first few-shot video object detector that effectively employs fine-tuning techniques, allowing for robust adaptation across diverse data availability scenarios. This includes cases with both a limited number of distinct categories and those with significant category variability. Our method, FTFSVid, surpasses previous leading few-shot video object detectors, specifically FSVOD, by achieving improvements of 3.2 and 14.5 AP_{50} points on FSVOD-500 and FSYTV-40 datasets, respectively. Additionally, we have proved that the proposed spatio-temporal components enhance performance over the single-image baseline by 5.5 AP_{50} points on FSVOD-500 and 1.4 on FSYTV-40. These improvements are statistically significant, establishing a new state-of-the-art in few-shot video object detection.

Although incorporating spatio-temporal information significantly reduces classification errors in few-shot object detectors, these errors remain the most prevalent. This is expected given the challenging few-shot setting, where the limited number of objects per category makes accurate classification inherently difficult.

Acknowledgment

This research was partially funded by the Spanish Ministerio de Ciencia e Innovación (grant number PID2020-112623GB-I00), and the Galician Consellería de Cultura, Educación e Universidade (grant numbers ED431C 2018/29, ED431C 2021/048, ED431G 2019/04). These grants are co-funded by the European Regional Development Fund (ERDF).

References

[1] Bertasius, G., Torresani, L., Shi, J., 2018. Object detection in video with spatiotemporal sampling networks, in: IEEE Int. Conf. Comput. Vis. (ICCV).

[2] Bolya, D., Foley, S., Hays, J., Hoffman, J., 2020. TIDE: A general toolbox for identifying object detection errors, in: European Conf. Comput. Vis. (ECCV), Springer. pp. 558–573.

[3] Bosquet, B., Cores, D., Seidenari, L., Brea, V.M., Mucientes, M., Bimbo, A.D., 2022. A full data augmentation pipeline for small object detection based on generative adversarial networks. Pattern Recognit. , 108998.

[4] Bosquet, B., Mucientes, M., Brea, V.M., 2021. Stdnet-st: Spatio-temporal convnet for small object detection. Pattern Recognition 116, 107929.

[5] Cao, Y., Wang, J., Lin, Y., Lin, D., 2022. Mini: Mining implicit novel instances for few-shot object detection. arXiv preprint arXiv:2205.03381 .

[6] Chen, T.I., Liu, Y.C., Su, H.T., Chang, Y.C., Lin, Y.H., Yeh, J.F., Chen, W.C., Hsu, W., 2021. Dual-awareness attention for few-shot object detection. IEEE Trans. Multimed. .

[7] Chen, Y., Cao, Y., Hu, H., Wang, L., 2020. Memory enhanced global-local aggregation for video object detection, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 10337–10346.

[8] Cores, D., Brea, V.M., Mucientes, M., 2021. Short-term anchor linking and long-term self-guided attention for video object detection. Image and Vision Computing 110, 104179.

[9] Cores, D., Brea, V.M., Mucientes, M., 2022. Spatiotemporal tubelet feature aggregation and object linking for small object detection in videos. Appl. Intell. , 1–13.

[10] Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T., 2019. Relation distillation networks for video object detection, in: IEEE Int. Conf. Comput. Vis. (ICCV), pp. 7023–7032.

[11] Fan, Q., Tang, C.K., Tai, Y.W., 2022. Few-shot video object detection, in: European Conf. Comput. Vis. (ECCV).

[12] Fan, Q., Zhuo, W., Tang, C.K., Tai, Y.W., 2020. Few-shot object detection with attention-RPN and multi-relation detector, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 4013–4022.

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

[14] Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y., 2018. Relation networks for object detection, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 3588–3597.

[15] Jia, J., Feng, X., Yu, H., 2024. Few-shot classification via efficient meta-learning with hybrid optimization. Engineering Applications of Artificial Intelligence 127, 107296.

[16] Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T., 2019. Few-shot object detection via feature reweighting, in: IEEE Int. Conf. Comput. Vis. (ICCV), pp. 8420–8429.

[17] Kaul, P., Xie, W., Zisserman, A., 2022. Label, verify, correct: A simple few shot object detection method, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 14237–14247.

[18] Li, Y., Mao, H., Girshick, R., He, K., 2022. Exploring plain vision transformer backbones for object detection, in: European Conf. Comput. Vis. (ECCV), Springer. pp. 280–296.

[19] Peng, J., Wang, C., Wan, F., Wu, Y., Wang, Y., Tai, Y., Wang, C., Li,

J., Huang, F., Fu, Y., 2020. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking, in: European Conf. Comput. Vis. (ECCV), pp. 145–161.

[20] Pérez, P., Gangnet, M., Blake, A., 2003. Poisson image editing, in: ACM SIGGRAPH 2003 Papers, pp. 313–318.

[21] Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., Zhang, C., 2021. DeFRCN: Decoupled Faster R-CNN for few-shot object detection, in: IEEE Int. Conf. Comput. Vis. (ICCV), pp. 8681–8690.

[22] Redmon, J., Farhadi, A., 2017. Yolo9000: better, faster, stronger, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 7263–7271.

[23] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks, in: Adv. Neural Inf. Process. Syst. (NIPS).

[24] Rodríguez-Fdez, I., Canosa, A., Mucientes, M., Bugarín, A., 2015. STAC: a web platform for the comparison of algorithms using statistical tests, in: IEEE Int. Conf. Fuzzy Sys. (FUZZ-IEEE).

[25] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115, 211–252.

[26] Saha, S., Singh, G., Sapienza, M., Torr, P., Cuzzolin, F., 2016. Deep learning for detecting multiple space-time action tubes in videos, in: Br. Mach. Vis. Conf. (BMVC).

[27] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision, in: IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 2818–2826.

[28] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, in: Adv. Neural Inf. Process. Syst. (NIPS), pp. 5998–6008.

[29] Wang, K., Liu, M., 2022. Yolo-anti: Yolo-based counterattack model for unseen congested object detection. Pattern Recognition 131, 108814.

[30] Wang, X., Huang, T., Gonzalez, J., Darrell, T., Yu, F., 2020. Frustratingly simple few-shot object detection, in: Int. Conf. Mach. Learn. (ICML), pp. 9919–9928.

[31] Wu, S., Li, X., Wang, X., 2020. IoU-aware single-stage object detector for accurate localization. Image and Vision Computing 97, 103911.

[32] Xu, R., Shao, S., Xing, L., Wang, Y., Liu, B., Liu, W., 2024. Ensembling multi-view discriminative semantic feature for few-shot classification. Engineering Applications of Artificial Intelligence 132, 107915.

[33] Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., Lin, L., 2019. Meta R-CNN: Towards general solver for instance-level low-shot learning, in: IEEE Int. Conf. Comput. Vis. (ICCV), pp. 9577–9586.

[34] Yang, X., Li, Z., Zhong, X., Zhang, C., Ma, H., 2023. Mining graph-based dynamic relationships for object detection. Engineering Applications of Artificial Intelligence 126, 106928.

[35] Zhang, D., Li, J., Li, X., Du, Z., Xiong, L., Ye, M., 2021a. Local–global attentive adaptation for object detection. Engineering Applications of Artificial Intelligence 100, 104208.

[36] Zhang, X., Guo, W., Xing, Y., Wang, W., Yin, H., Zhang, Y., 2023. AugFCOS: Augmented fully convolutional one-stage object detection network. Pattern Recognition 134, 109098.

[37] Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W., 2021b. Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision 129, 3069–3087.

[38] Zhou, X., Koltun, V., Krähenbühl, P., 2020. Tracking objects as points, in: European Conf. Comput. Vis. (ECCV), Springer. pp. 474–490.

Appendix A. Qualitative analysis

Fig. A.10 presents a qualitative analysis of FTFSVid performance on the FSVOD-500 dataset, comparing it with our spatial baseline, DeFRCN. Overall, FTFSVid shows improved classification accuracy over the baseline. In the first example, FTFSVid shows higher confidence for the correct category removing the duplicate detection, while in the third example, it resolves false positives caused by classification errors. The second example illustrates its ability to reduce "missing" errors, as discussed in the experiments section. In the last example, FTFSVid successfully detects an object overlooked by DeFRCN, however it misclassifies the category, highlighting that object classification remains challenging even with spatio-temporal information.

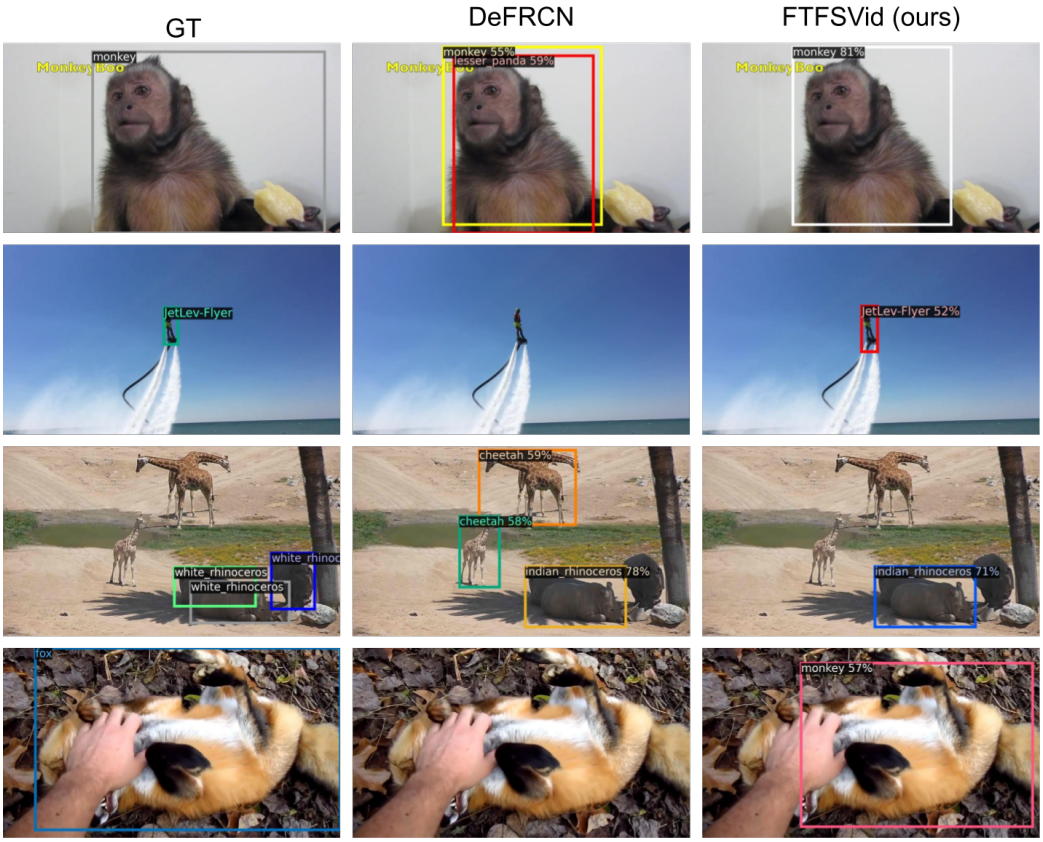


Figure A.10: Qualitative evaluation. From right lo left: ground truth annotations, detections from our spatial baseline (DeFRCN) and detections from our method (FTFSVid) leveraging spatio-temporal information.