Video Object Segmentation with Joint Re-identification and Attention-Aware Mask Propagation

Xiaoxiao Li[1][0000−0002−0376−5135] and Chen Change Loy[2][0000−0001−5345−1591]

[1] Department of Information Engineering, The Chinese University of Hong Kong
lx015@ie.cuhk.edu.hk
[2] Nanyang Technological University
ccloy@ieee.org

Abstract. The problem of video object segmentation can become extremely challenging when multiple instances co-exist. While each instance may exhibit large scale and pose variations, the problem is compounded when instances occlude each other causing failures in tracking. In this study, we formulate a deep recurrent network that is capable of segmenting and tracking objects in video simultaneously by their temporal continuity, yet able to re-identify them when they re-appear after a prolonged occlusion. We combine temporal propagation and re-identification functionalities into a single framework that can be trained end-to-end. In particular, we present a re-identification module with template expansion to retrieve missing objects despite their large appearance changes. In addition, we contribute an attention-based recurrent mask propagation approach that is robust to distractors not belonging to the target segment. Our approach achieves a new state-of-the-art $\mathcal{G}$-mean of 68.2 on the challenging DAVIS 2017 benchmark (test-dev set), outperforming the winning solution. Project Page: http://mmlab.ie.cuhk.edu.hk/projects/DyeNet/.

1 Introduction

Video object segmentation aims at segmenting foreground instance object(s) from the background region in a video sequence. Typically, ground-truth masks are assumed to be given in the first frame. The goal is to begin with these masks and track them in the remaining sequence. This paradigm is sometimes known as semi-supervised video object segmentation [24, 3, 27]. A notable and challenging benchmark for this task is 2017 DAVIS Challenge [28]. An example of a sequence is shown in Fig. 1. The DAVIS dataset presents real-world challenges that need to be solved from two key aspects. First, there are multiple instances in a video. It is very likely that they will occlude each other causing partial or even full obstruction of a target instance. Second, instances typically experience substantial variations in both scale and pose across frames.

To address the occlusion problem, notable studies such as [3, 39] adapt generic semantic segmentation deep model to the task of specific object segmentation. These methods follow a notion reminiscent of the template matching based methods that are widely used in visual tracking task [2, 33]. Often, a fixed set of templates such as the masks of target objects in the first frame are used for matching targets. This paradigm fails in some challenging cases in DAVIS (see Fig. 1(a)), as using a fixed set of templates
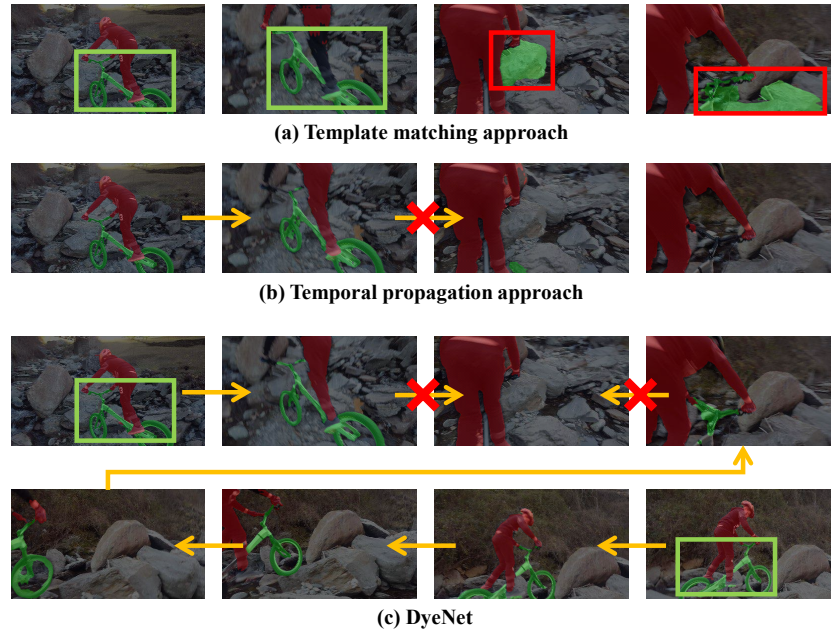
(a) Template matching approach

(b) Temporal propagation approach

(c) DyeNet

Fig. 1. We focus on the bicycle in this example. (a) shows the result of template matching approach, which is affected by large scale and pose variations. As shown in (b), temporal propagation is incapable of handling occlusion. The proposed DyeNet joints them into a unified framework, first retrieves high confidence starting points and then propagates their masks bidirectionally to address those issues. The result of DyeNet is visualized in (c). **Best viewed in color.**

cannot sufficiently cover large scale and pose variations. To mitigate the variations in both scale and pose across frames, existing studies [32, 35, 15, 26, 34, 16] exploit temporal information to maintain continuity of individual segmented regions across frames. On unconstrained videos with severe occlusions, such as that shown in Fig. 1(b), approaches based on temporal continuity are prone to errors since there is no mechanism to re-identify a target when it reappears after missing in a few video frames. In addition, these approaches may fail to track instances in the presence of distractors such as cluttered backgrounds or segments from other objects during temporal propagation.

Solving video object segmentation with multiple instances requires template matching for coping with occlusion and temporal propagation for ensuring temporal continuity. In this study, we bring both approaches into a single unified network. Our network hinges on two main modules, namely a re-identification (Re-ID) module and a recurrent mask propagation (Re-MP) module. The Re-ID module helps to establish confident starting points in non-successive frames and retrieve missing segments caused by occlusions. Based on the segments provided by the Re-ID module, the Re-MP module propagates their masks bidirectionally by a recurrent neural network to the entire video. The process of conducting Re-ID followed by Re-MP may be imagined as dyeing a

fabric with multiple color dots (*i.e.*, choosing starting points with re-identification) and the color disperses from these dots (*i.e.*, propagation). Drawing from this analogy, we name our network as *DyeNet*.

There are a few methods [17, 21] that improve video object segmentation through both temporal propagation and re-identification. Our approach differs by offering a unified network that allows both tasks to be optimized in an end-to-end network. In addition, unlike existing studies, the Re-ID and Re-MP steps are conducted in an iterative manner. This allows us to identify confidently predicted mask in each iteration and expand the template set. With a dynamic expansion of template set, our Re-ID module can better retrieve missing objects that reappear with different poses and scales. In addition, the Re-MP module is specially designed with attention mechanism to disregard distractors such as background objects or segments from other objects during mask propagation. As shown in Fig. 1(c), DyeNet is capable of segmenting multiple instances across a video with high accuracy through Re-ID and Re-MP. We provide a more detailed discussion against [17, 21] in the related work section.

Our **contributions** are summarized as follows. (1) We propose a novel approach that joints template matching and temporal propagation into a unified deep neural network for addressing video object segmentation with multiple instances. The network can be trained end-to-end. It does not require online training (*i.e.*, fine-tune using the masks of the first frame) to do well but can achieve better results with online training. (2) We present an effective template expansion approach to better retrieve missing targets that reappear with different poses and scales. (3) We present a new attention-based recurrent mask propagation module that is more resilient to distractors.

We use the challenging DAVIS 2017 dataset [28] as our key benchmark. The winner of this challenge [21] achieves a global mean (Region Jaccard and Boundary F measure) of 66.1 on the test-dev partition. Our method obtains a global mean of 68.2 on this partition. Without online training, DyeNet can still achieve a competitive $\mathcal{G}$-mean of 62.5 while the speed is an order of magnitude faster. Our method also achieves state-of-the-art results on DAVIS 2016 [27], SegTrack$_{v2}$ [19] and YouTubeObjects [29] datasets.

2   Related Work

Image segmentation. The goal of semi-supervised video object segmentation is different to semantic image segmentation [4, 41, 23, 20, 40] and instance segmentation [8, 9, 22, 10] that perform pixel-wise class labeling. In video object segmentation, the class type is always assumed to be undefined. Thus, the challenge lies in performing accurate object-agnostic mask propagation. Our network leverages semantic image segmentation task to learn generic representation that encompasses semantic level information. The representation learned is strong, allowing our model to be applied in a dataset-agnostic manner, *i.e.*, it is not trained with any first frame annotation of each video in the target dataset as training/tuning set, but it can also be optionally fine-tuned and adapted into the targeted video domain as practiced in [16] to obtain better results. We will examine both possibilities in the experimental section.

Visual tracking. While semi-supervised video object segmentation can be seen as a pixel-level tracking task, video object segmentation differs in its more challenging na-

ture in terms of object scale variation across video frames and inter-object scale differences. In addition, the pose of objects is relatively stable in the tracking datasets, and there are few prolonged occlusions. Importantly, the problem differs in that conventional tracking tasks only need bounding box level tracking results, and concern about causality (*i.e.*, tracker does not use any future frames for estimation). In contrast, semi-supervised video object segmentation expects precise pixel-level tracking results, and typically does not assume causality.

Video object segmentation. Prior to the prevalence of deep learning, most approaches to semantic video segmentation are graph based [7, 18, 37, 25]. Contemporary methods are mostly based on deep learning. A useful technique reminiscent of template matching is commonly applied. In particular, templates are typically formed by the ground-truth masks in the first frame. For instance, Caelles *et al* [3] adapt a generic semantic image segmentation network to the templates for each testing video individually. Yoon *et al* [39] distinguish the foreground objects based on the pixel-level similarity between candidates and templates, which is measured by a matching deep network. Another useful technique is to exploit temporal continuity for establishing spatiotemporal correlation. Tsai *et al* [32] estimate object segmentation and optical flow synergistically using an iterative scheme. Jampani *et al* [15] propagate structured information through a video sequence by a bilateral network that performs learnable bilateral filtering operations cross video frames. Perazzi *et al* [26] and Jang *et al* [34] estimate the segmentation mask of the current frame by using the mask from the previous frame as a guidance.

Differences against existing methods that combine template matching and temporal continuity. There are a few studies that combine the merits of the two aforementioned techniques. Khoreva *et al* [16] show that a training set closer to the target domain is more effective. They improve [3] by synthesizing more training data from the first frame of testing videos and employ mask propagation during the inference. Instance Re-Identification Flow (IRIF) [17] divides foreground objects into human and non-human object instances, and then apply person re-identification network [36] to retrieve missing human during mask propagation. For non-human object instances, IRIF degenerates to a conventional mask propagation method. Our method differs to these studies in that we do not synthesize training data from the first frames and do not explicitly divide foreground objects into human and non-human object instances.

Li *et al* [21] adapt person re-identification approach [36] to a generic object re-identification model and employ a two-stream mask propagation model [26]. Their method (VS-ReID) achieved the highest performance in the 2017 DAVIS Challenge [21], however, its shortcomings are also obvious: (1) VS-ReID only uses the masks of target objects in the first frame as templates. It is thus more susceptible to pose variations. (2) Their method is much slower compared to ours due to its redundant feature extraction steps and less efficient inference method. Specifically, the inference of VS-ReID takes ∼3 seconds per frame on DAVIS dataset. The speed is 7 times slower than DyeNet. (3) VS-ReID does not have any attention mechanism in its mask propagation. Its robustness to distractors and background clutters is thus inferior to DyeNet. (4) VS-ReID cannot perform end-to-end training. By contrast, DyeNet performs joint learning of re-identification and temporal propagation.
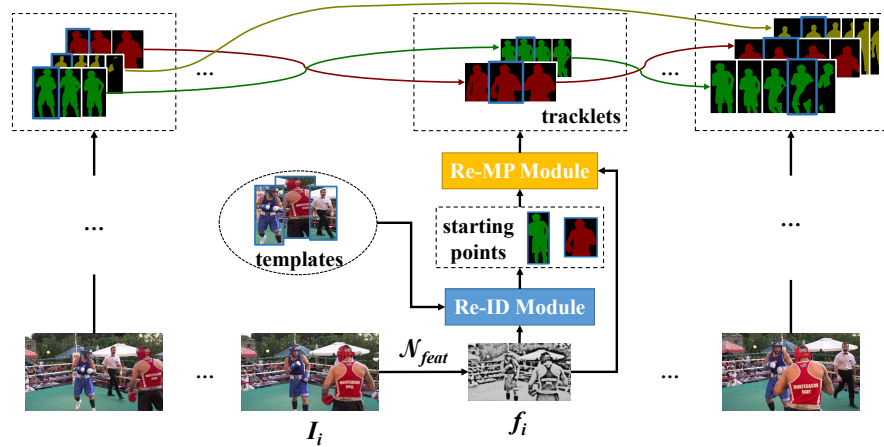
Fig. 2. The pipeline of DyeNet. The network hinges on two main modules, namely a re-identification (Re-ID) module and a recurrent mask propagation (Re-MP) module. **Best viewed in color.**

3   Methodology

We provide an overview of the proposed approach. Figure 2 depicts the architecture of DyeNet. It consists of two modules, namely the re-identification (Re-ID) module and the recurrent mask propagation (Re-MP) module. The network first performs feature extraction, which will be detailed next.

Feature extraction. Given a video sequence with N frames $\{I_1, ..., I_N\}$, for each frame I_i, we first extract a feature f_i by a convolutional feature network $\mathcal{N}_{feat}$, *i.e.*, $f_i = \mathcal{N}_{feat}(I_i)$. Both Re-ID and Re-MP modules employ the same set of features in order to save computation in feature extraction. Considering model capacity and speed, we use ResNet-101 [11] as the backbone of $\mathcal{N}_{feat}$. More specifically, ResNet-101 consists of five blocks named as 'conv1', 'conv2_x' to 'conv5_x'. We employ 'conv1' to 'conv4_x' as our feature network. To increase the resolution of features, we decrease the convolutional strides in 'conv4_x' block and replace convolutions in 'conv4_x' by dilated convolutions similar to [4]. Consequently, the resolution of feature maps is $1/8$ of the input frame.

Iterative inference with template expansion. After feature extraction, DyeNet runs Re-ID and Re-MP in an iterative manner to obtain segmentation masks of all instances across the whole video sequence. We assume the availability of masks given in the first frame and use them as templates. This is the standard protocol of the benchmarks considered in Sec. 4.

   In the first iteration, the Re-ID module generates a set of masks from object proposals and compares them with templates. Masks with a high similarity to templates are chosen as the starting points for Re-MP. Subsequently, Re-MP propagates each selected mask (*i.e.*, starting point) bidirectionally, and generates a sequence of segmentation masks, which we call tracklet. After Re-MP, we can additionally consider post-
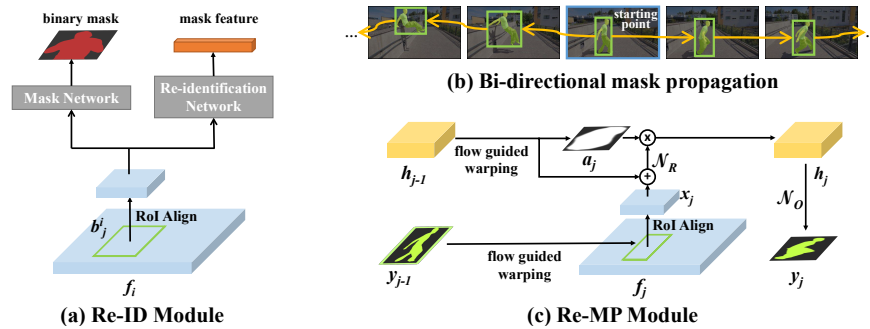
Fig. 3. (a) The network architecture of the re-identification (Re-ID) module. (b) Illustration of bi-direction mask propagation. (c) The network architecture of the recurrent mask propagation (Re-MP) module. **Best viewed in color.**

processing steps to link the tracklets. In subsequent iterations, DyeNet chooses confidently predicted masks to expand the template set and reapplies Re-ID and Re-MP. Template expansion avoids heavy reliance on the masks provided by the first frame, which may not capture sufficient pose variations of targets.

Note that we do not expect to retrieve all the masks of target objects in a given sequence. In the first iteration, it is sufficient to obtain several high-quality starting points for the mask propagation step. After each iteration of DyeNet, we select predictions with high confidence to augment the template set. In practice, the first iteration can retrieve nearly 25% masks as starting points on DAVIS 2017 dataset. After three iterations, this rate will increase to 33%. In this work, DyeNet stops the iterative process when no more high-confident masks can be found by the Re-ID module. Next, we present the Re-ID and Re-MP modules.

3.1 Re-identification

We introduce the Re-ID module to search for targets in the video sequences. The module has several unique features that allow it to retrieve a missing object that reappears in different scales and poses. First, as discussed previously, we expand the template set in every iteration we apply Re-ID and Re-MP. Template expansion enriches the template set for more robust matching. Second, we employ the object proposal method to estimate the location of target objects. Since these proposals are generated based on anchors of various sizes, which cover objects of various scales, the Re-ID module can handle large scale variations.

Figure 3(a) illustrates the Re-ID module. For the i-th frame, besides the feature f_i, the Re-ID module also requires the object proposals $\{b_1^i, ..., b_M^i\}$ as input where M indicates the number of proposal bounding boxes on this frame. We employ a Region Proposal Network (RPN) [30] to propose candidate object bounding boxes on every frame. For convenience, our RPN is trained separately from DyeNet, but their backbone networks are shareable. For each candidate bounding box b_j^i, we first extract its feature from f_i, and resize the feature into a fixed size $m \times m$ (*e.g.*, 28×28) by RoIAlign [10],

which is an improved form of RoIPool that removes harsh quantization. The extracted features are fed into two shallow sub-networks. The first sub-network is a mask network that predicts a $m \times m$ binary mask that represents the segmentation mask of the main instance in candidate bounding box b_j^i. The second sub-network is a re-identification network that projects the extracted features into an L2-normalized 256-dimensional subspace to obtain the mask features. The templates are also projected onto the same subspace for feature extraction.

By computing the cosine similarities between the mask and template features, we can measure the similarity between candidate bounding boxes and templates. If a candidate bounding box is sufficiently similar to any template, that is, the cosine similarity is larger than a threshold ρ_{reid}, we will keep its mask as a starting point for mask propagation. In practice, we set ρ_{reid} with a high value to establish high-quality starting points for our next step.

We employ 'conv5_x' block of ResNet-101 as the backbone of the sub-networks. However, some modifications are necessary to adapt them to the respective tasks. In particular, we decrease the convolutional strides in the mask network to capture more details of prediction. For the re-identification network, we keep the original strides and append a global average pooling layer and a fully connected layer to project the features into the target subspace.

3.2   Recurrent Mask Propagation

As shown in Fig. 3(b), we bi-directionally extend the retrieved masks (*i.e.*, starting points) to form tracklets by using the Re-MP module. By incorporating short-term memory, the module is capable of handling large pose variations, which complements the re-identification module. We formulate the Re-MP module as a Recurrent Neural Network (RNN). Figure 3(c) illustrates the mask propagation process between adjacent frames. For brevity, we only describe the forward propagation. A backward propagation can be conducted with the same approach.

Suppose $\hat{y}$ is a retrieved segmentation mask for instance k in the i-th frame, and we have propagated $\hat{y}$ from i-th frame to $(j-1)$-th frame, $\{y_{i+1}, y_{i+2}, ..., y_{j-1}\}$ is the sequence of binary masks that we obtain. We now aim to predict y_j, *i.e.*, the mask for instance k in the j-th frame. In a RNN framework, the prediction of y_j can be solved as

$$h_j = \mathcal{N}_R(h_{(j-1)\to j}, x_j), \tag{1}$$

$$y_j = \mathcal{N}_O(h_j), \tag{2}$$

where $\mathcal{N}_R$ and $\mathcal{N}_O$ are the recurrent function and output function, respectively.

We first explain Eq. (1). We begin with estimating the location, *i.e.*, the bounding box, of instance k in the j-th frame from y_{j-1} by flow guided warping. More specifically, we use FlowNet 2.0 [13] to extract the optical flow $F_{(j-1)\to j}$ between $(j-1)$-th and j-th frames. Other flow estimation methods [12, 31] are applicable too. The binary mask y_{j-1} is warped to $y_{(j-1)\to j}$ according to $F_{(j-1)\to j}$ by a bilinear warping function. After that, we obtain the bounding box of $y_{(j-1)\to j}$ as the location of instance k in the j-th frame. Similar to the Re-ID module, we extract the feature map according to this bounding box from f_j by RoIAlign operation. The feature of this bounding box
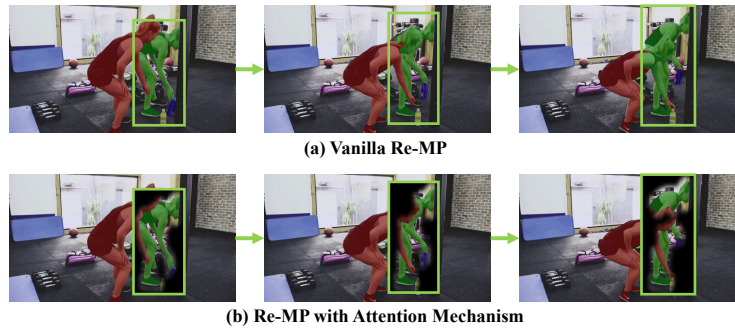
(a) Vanilla Re-MP

(b) Re-MP with Attention Mechanism

Fig. 4. Region attention in mask propagation.

is denoted as x_j. The historical information of instance k from i-th frame to $(j-1)$-th frame is expressed by a hidden state or memory $h_{j-1} \in \mathbb{R}^{m \times m \times d}$, where $m \times m$ denotes the feature size and d represents the number of channels. We warp h_{j-1} to $h_{(j-1) \to j}$ by optical flow for spatial consistency. With both x_j and $h_{(j-1) \to j}$ we can estimate h_j by Eq. (1). Similar to the mask network described in Sec. 3.1, we employ 'conv5_x' block of ResNet-101 as our recurrent function $\mathcal{N}_R$. The mask for the instance k in the j-th frame, y_j, can then be obtained by using the output function in Eq. (2). The output function $\mathcal{N}_O$ is modeled by three convolutional layers.

Region attention. The quality of propagation to obtain y_j relies on how accurate the model in capturing the shape of target instance. In many cases, a bounding box may contain distractors that can jeopardize the quality of mask propagated. As shown in Fig 4(a), if we directly generate y_j from h_j, a model is likely to be confused by distractors that appear in the bounding box. To overcome this issue, we leverage the attention mechanism to filter out potentially noisy regions. It is worth pointing out that attention mechanism has been used in various computer vision tasks [1, 38] but not mask propagation. Our work presents the first attempt to incorporate attention mechanism in mask propagation.

Specifically, given the warped hidden state $h_{(j-1) \to j}$, we first feed it into a single convolutional layer and then a softmax function, to generate the attention distribution $a_j \in \mathbb{R}^{m \times m \times 1}$ over the bounding box. Figure 4(b) shows the attention distributions we learned. Then we multiply the current hidden state h_j by a_j across all channels to focus on the regions we interested. And the mask y_j is generated from enhanced h_j by using Eq. (2). As shown in Fig. 4, the Re-MP module concentrates on the tracked object thanks to the attention mechanism. The mask propagation of an object aborts when its size is too small, indicating a high possibility of occlusion. Finally, $\hat{y}$ is extended to a tracklet $\{y_{k1}, ..., y_{i+1}, \hat{y}, y_{i+1}, ..., y_{k2}\}$ after the forward and backward propagation. This process is applied to all the starting points to generate a set of tracklets. However, in some cases, different starting points may produce the same tracklet, which leads to redundant computation. To speed up the algorithm, we sort all starting points descendingly by their cosine similarities against templates. We extend the starting points according to the sorted order. If a starting point's mask highly overlaps with a mask in

existing tracklets, we skip this starting point. This step does not affect the results; on the contrary, it greatly accelerates the inference speed.

Linking the tracklets. The previous mask propagation step generates potentially segmented tracklets. We introduce a greedy approach to link those tracklets into consistent mask tubes. It sorts all tracklets descendingly by cosine similarities between their respective starting point and templates. Given the sorted order, tracklets with the highest similarities are assigned to the respective templates. The method then examines the remaining tracklets in turn. A tracklet is merged with a tracklet of higher order if there is no contradiction between them. In practice, this simple mechanism works well. We will investigate other plausible linking approaches (*e.g.*, conditional random field) in the future.

3.3   Inference and Training

Iterative inference. During inference, we are given a video sequence $\{I_1, ..., I_N\}$, and the masks of target objects in the first frame. As mentioned, we employ those masks as the initial templates. DyeNet is iteratively applied to the whole video sequence until no more high confidence instances can be found. The set of templates will be augmented by the predictions with high confidences after each iteration.

Training details. The overall loss function of DyeNet is formulated as: $L = L_{reid} + \lambda(L_{mask} + L_{remp})$, where L_{reid} is the re-identification loss of re-identification network in Sec. 3.1, which follows Online Instance Matching (OIM) loss in [36]. L_{mask} and L_{remp} indicate the pixel-wise segmentation losses of the mask network in Sec. 3.1 and recurrent mask propagation module in Sec. 3.2. The overall loss is a linear combination of those three losses, where λ is a weight that balances the scale of those lose terms. Following [21, 16], the feature network is pre-trained by the semantic segmentation task. The DyeNet is then jointly trained on the DAVIS training sets using 24k iterations. We fix a mini-batch size of 32 images (from 8 videos, 4 frames for each video), momentum 0.9 and weight decay of 5^{-4}. The initial learning rate is 10^{-3} and dropped by a factor of 10 after every 8k iterations.

4   Experiments

Datasets. To demonstrate the effectiveness and generalization ability of DyeNet, we evaluate our method on DAVIS 2016 [27], DAVIS 2017 [28], SegTrack$_{v2}$ [19] and YouTubeObjects [29] datasets. DAVIS 2016 (DAVIS$_{16}$) dataset contains 50 high-quality video sequences (3455 frames) with all frames annotated with pixel-wise object masks. Since DAVIS$_{16}$ focuses on single-object video segmentation, each video has only one foreground object. There are 30 training and 20 validation videos. DAVIS 2017 (DAVIS$_{17}$) supplements the training and validation sets of DAVIS$_{16}$ with 30 and 10 high-quality video sequences, respectively. It also introduces another 30 development test videos and 30 challenge testing videos, which makes DAVIS$_{17}$ three times larger than its predecessor. Besides that, DAVIS$_{17}$ re-annotates all video sequences with multiple objects. All of these differences make it more challenging than DAVIS$_{16}$. SegTrack$_{v2}$ dataset contains 14 low resolution video sequences (947 frames) with 24 generic foreground

Table 1. Ablation study on Re-MP with DAVIS$_{17}$ *val*.

	Variant	$\mathcal{J}$-mean	$\mathcal{F}$-mean	$\mathcal{G}$-mean
MSK [26]	ResNet-101	63.3	67.2	65.3
Re-MP	no attention	65.3	69.7	67.5
	full	**67.3**	**71.0**	**69.1**

objects. For YouTubeObjects [29] dataset, we consider a subset of 126 videos with around 20000 frames, and the pixel-level annotation are provided by [14].

Evaluation metric. For DAVIS$_{17}$ dataset, we follow [28] that adopts region ($\mathcal{J}$), boundary ($\mathcal{F}$) and their average ($\mathcal{G}$) measures for evaluation. To be consistent with existing studies [34, 16, 26, 3], we use mean intersection over union (mIoU) averaged across all instances to evaluate the performance in DAVIS$_{16}$, SegTrack$_{v2}$ and YouTubeObjects.

Training modalities. In existing studies [26, 16], training modalities can be divided into **offline training** and **online training**. In offline training a model is only trained on the training set without any annotations from the test set. Since the first frame annotations are provided in the testing stage, we can use them for tuning the model, namely online training. Online training can be further divided into **per-dataset** and **per-video** training. In per-dataset online training, we fine-tune a model based on all the first frame annotations from the test set, to obtain a dataset-specific model. Per-video online training adapts the model weights to each testing video, *i.e.*, there will be as many video specific models as the testing videos during the testing stage.

4.1   Ablation Study

In this section, we investigate the effectiveness of each component in DyeNet. Unless otherwise indicated we employ the *train set* of DAVIS$_{17}$ for training. All performance are reported on the *val set* of DAVIS$_{17}$. Offline training modality is used.

Effectiveness of Re-MP module. To demonstrate the effectiveness of Re-MP module, we do not involve the Re-ID module in this experiment. Re-MP module is directly applied to extend the annotations in the first frame to form mask tubes. This variant degenerates our method to a conventional mask propagation pipeline but with an attention-aware recurrent structure. We compare Re-MP module with the state-of-the-art mask propagation method, MSK [26]. To ensure a fair comparison, we re-implement MSK to have the same backbone ResNet-101 as DyeNet. We do not use online training and any post-processing in MSK either. The re-implemented MSK achieves 78.7 $\mathcal{J}$-mean on DAVIS$_{16}$ *val set*, which is much higher than the original result 69.9 reported in [26].

As shown in Table 1, MSK achieves 65.3 $\mathcal{G}$-mean on DAVIS$_{17}$ *val set*. Unlike MSK that propagates predicted masks only, the proposed Re-MP propagates all historical information by the recurrent architecture, and RoIAlign operation allows our network to focus on foreground regions and produce high-resolution masks, which makes Re-MP outperform MSK. The Re-MP with attention mechanism is more focused on foreground regions, which further improves $\mathcal{G}$-mean by 1.6.

Figure 5 shows propagation results of different methods. In this video, a dog passes in front of a woman and another dog. MSK dyes the woman and the back dog with the
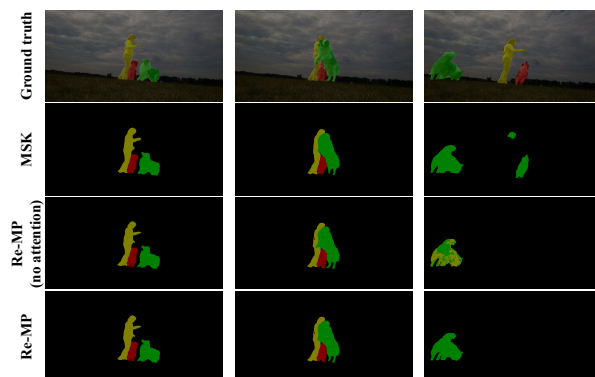
Fig. 5. Examples of mask propagation. **Best viewed in color.**

Table 2. Ablation study on Re-ID with DAVIS$_{17}$ *val*. The improvement of $\mathcal{G}$-mean between rows is because of template expansion.

| ρ_{reid}	0.9			0.8			0.7			0.6		
	preci.	recall	$\mathcal{G}$-mean	preci.	recall	$\mathcal{G}$-mean	preci.	recall	$\mathcal{G}$-mean	preci.	recall	$\mathcal{G}$-mean
Iter. 1	97.0	16.0	72.3	87.1	22.2	73.2	78.9	26.2	73.2	76.5	29.2	73.4
Iter. 2	90.3	29.3	73.3	75.6	32.5	73.7	68.9	33.5	**74.1**	65.5	34.1	74.0
Iter. 3$^+$	90.7	30.1	73.6	74.6	32.6	73.7	68.8	33.5	**74.1**	65.3	34.2	73.9

instance id of the front dog. The plain Re-MP does not dye other instances, but it is still confused during the crossing and assigns the front dog with two instance ids. Thanks to the attention mechanism, our full Re-MP is not distracted by other instances. Due to occlusion, the masks of other instances are lost, and they will be retrieved by the Re-ID module in the complete DyeNet.

Effectiveness of Re-ID module with template expansion. In DyeNet, we employ the Re-ID module to search for target objects in the video sequence. By choosing an appropriate similarity threshold ρ_{reid}, we can establish high-quality starting points for the Re-MP module. The threshold ρ_{reid} controls the trade-off between precision and recall of retrieved objects. Table 2 lists the precision and recall of retrieved starting points in each iteration as ρ_{reid} varies, and corresponding overall performance. Tracklets are linked by greedy algorithm in this experiment.

Overall, the $\mathcal{G}$-mean is increased after each iteration due to the template expansion. When ρ_{reid} decreases, more instances are retrieved in the first iteration, which leads to high recall and $\mathcal{G}$-mean. It also produces some imprecise starting points and further affects the quality of templates in subsequent iterations, so the increase of performance between each iteration is limited. In contrast, Re-ID module with high ρ_{reid} is stricter. As the template set expands, it can still achieve satisfying recall rate gradually. In practice, the iterative process stops in about three rounds. Due to our greedy algorithm, the overall performance is less sensitive to ρ_{reid}. When $\rho_{reid} = 0.7$, DyeNet achieves the best $\mathcal{G}$-mean. This value is used in all the following experiments.

Table 3. Ablation study of each module in DyeNet with DAVIS$_{17}$ *test-dev*.

	Variant	$\mathcal{J}$-mean	$\mathcal{F}$-mean	$\mathcal{G}$-mean	$\Delta\mathcal{G}$-mean
MSK[26]	ResNet-101	50.9	52.6	51.7	-
Re-MP	no attention	55.4	60.5	58.0	+ 6.2
	full	59.1	62.8	61.0	+ 9.2
+ Re-ID		**65.8**	**70.5**	**68.2**	+ 7.2
offline	offline only	60.2	64.8	62.5	- 5.6
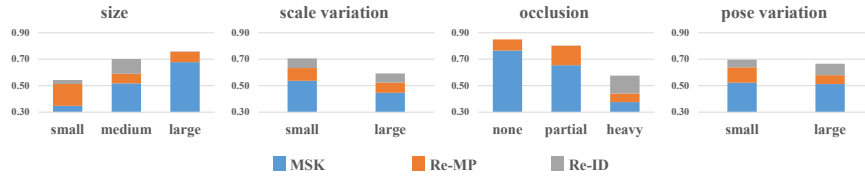
Fig. 6. Stage-wise performance increment according to specific attributes. **Best viewed in color.**

Effectiveness of each component in DyeNet. Table 3 summarizes how performance gets improved by adding each component step-by-step into our DyeNet on the *test-dev set* of DAVIS$_{17}$. Our re-implemented MSK is chosen as the baseline. All models in this experiment are first offline trained on the *train* and *val set*, and then per-dataset online trained on the *test-dev set*.

Compared with MSK, our Re-MP module with attention mechanism significantly improves $\mathcal{G}$-mean by 9.2. The full DyeNet that contains both Re-ID and Re-MP modules achieves 68.2 by using greedy algorithm to link the tracklets. More remarkably, without online training, our DyeNet achieves a competitive $\mathcal{G}$-mean of 62.5.

To further investigate the contribution of each module in DyeNet, we categorize instances in *test-dev set* by specific attributes, including:

- **Size:** Instances are categorized into 'small', 'medium', and 'large' according to their size in the first frames' annotations.
- **Scale Variation:** The area ratio among any pair of bounding boxes enclosing the target object is smaller than 0.5. The bounding boxes are obtained from our best prediction.
- **Occlusion:** An object is not, partially, or heavily occluded.
- **Pose Variation:** Noticeable pose variation, due to object motion or relative camera-object rotation.

We choose the best version of DyeNet in Table 3, and visualize its stage-wise performance according to specific attributes in Fig. 6. We find that object's size and occlusion are most important factors that affect the performance, and scale variation has more influence on the performance than pose variation. By inspecting closer, we observe that our Re-MP module can well track those small objects, which is the shortcoming of conventional mask propagation methods. It also avoids the distraction from other objects in partial occlusion cases. Complementary to Re-MP, Re-ID module retrieves missing
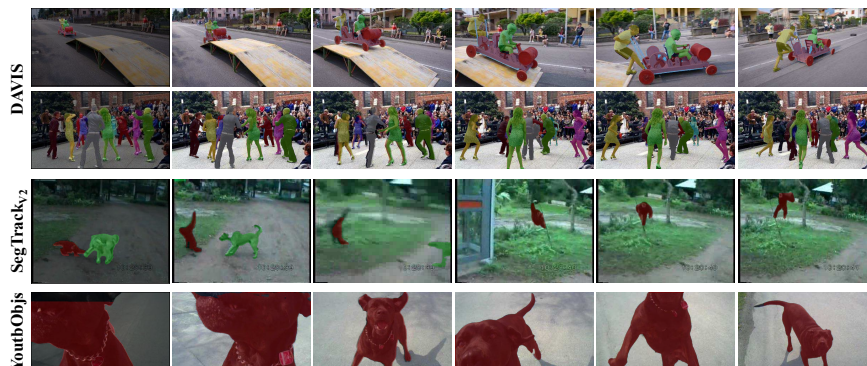
Fig. 7. Visualization of DyeNet's prediction. The first column shows the first frame of each video sequence with ground truth masks. The frames are chosen at equal interval. **Best viewed in color.**

Table 4. Results on DAVIS$_{17}$ *test-dev*.

	online training		$\mathcal{J}$-mean	$\mathcal{F}$-mean	$\mathcal{G}$-mean
	dataset	video			
OnAVOS[34]	√	√	53.4	59.6	56.5
LucidTracker[16]	√	√	60.1	68.3	64.2
VS-ReID[21]	√	×	64.4	67.8	66.1
LucidTracker[16]	√	√	63.4	69.9	66.6
DyeNet (offline)	×	×	60.2	64.8	62.5
DyeNet	√	×	**65.8**	**70.5**	**68.2**

instances due to heavy occlusions, greatly improves the performance in heavy occlusion cases. Even with large pose variations, template expansion ensures Re-ID works well, .

4.2 Benchmark

In this section, we compare our DyeNet with other existing methods and show that it can achieve the state-of-the-art performance on standard benchmarks, including DAVIS$_{16}$, DAVIS$_{17}$, SegTrack$_{v2}$ and YouTubeObjects datasets. In this section, DyeNet is tested on a single scale without any post-processing. Table 4 lists the $\mathcal{J}$, $\mathcal{F}$ and $\mathcal{G}$-means on DAVIS$_{17}$ *test-dev*. Approaches with ensemble are marked with [†]. DyeNet is trained on *train* and *val sets* of DAVIS$_{17}$ and achieves a competitive $\mathcal{G}$-mean of 62.5. It further improves $\mathcal{G}$-mean to 68.2 through online fine-tuning, which is the best-performing method on DAVIS$_{17}$ benchmark.

To show the generalization ability and transferability of DyeNet, we next evaluate DyeNet on three other benchmarks, DAVIS$_{16}$, SegTrack$_{v2}$ and YouTubeObjects, which contain diverse videos. For DAVIS$_{16}$, DyeNet is trained on its *train set*. Since there is no video for offline training in SegTrack$_{v2}$ and YouTubeObjects, we directly employ the model of DAVIS$_{17}$ as their offline model. As summarized in Table 5, offline DyeNet obtains promising performance, and after online fine-tuning, our model

Table 5. Results (mIoU) across three datasets.

	online training		DAVIS$_{16}$	SegTrack$_{v2}$	YoutbObjs
	dataset	video			
VPN[15]	×	×	75.0	-	-
SegFlow[5]	√	√	76.1	-	-
OSVOS[3]	√	√	79.8	65.4	72.5
MSK[26]	√	√	80.3	70.3	72.6
LucidTracker[16]	√	√	84.8	77.6	76.2
OnAVOS[34]	√	√	85.7	-	77.4
DyeNet (offline)	×	×	84.7	**78.3**	74.9
DyeNet	√	×	**86.2**	**78.7**	**79.6**

achieves state-of-the-art performance on all three datasets. Note that although the videos in SegTrack$_{v2}$ and YouTubeObjects are very different from videos in DAVIS$_{17}$, DyeNet trained on DAVIS$_{17}$ still gains outstanding performance on those datasets without any fine-tuning, which shows its great generalization ability and transferability to diverse videos. We also find that our offline predictions on YouTubeObjects are even better than most ground-truth annotations, and performance losses are mainly caused by annotation bias. In Fig. 7, we demonstrate some examples of DyeNet's predictions.

Speed Analysis. Most of existing methods require online training with post-processing to achieve a competitive performance. Because of those time consuming processes, their speed of inference is slow. For example, the full OnAVOS [34] takes roughly 13 seconds per frame to achieve 85.7 mIoU on DAVIS$_{16}$ *val set*. LucidTracker [16] that achieves 84.8 mIoU requires 40k iterations per-dataset, 2k per-video online training and post-processing[6]. Our offline DyeNet is capable of obtaining similar performance (84.7 mIoU) at 2.4 FPS on a single Titan Xp GPU. After 2k per-dataset online training, our DyeNet achieves 86.2 mIoU, and the corresponding running time is 0.43 FPS.

5 Conclusion

We have presented DyeNet, which joints re-identification and attention-based recurrent temporal propagation into a unified framework to address challenging video object segmentation with multiple instances. This is the first end-to-end framework for this problem with a few compelling components. First, to cope with pose variations of targets, we relaxed the reliance of template set in the first frame by performing template expansion in our iterative algorithm. Second, to achieve robust video segmentation against distractors and background clutters, we proposed attention mechanism for recurrent temporal propagation. DyeNet does not require online training to obtain competitive accuracies at a faster speed than many existing methods. With online training, DyeNet achieves state-of-the-art performance on a wide range of standard benchmarks (including DAVIS, SegTrack$_{v2}$ and YouTubeObjects).

References

1. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. In: ICLR (2015)
2. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: CVPR (2010)
3. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR (2017)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: ICLR (2015)
5. Cheng, J., Tsai, Y.H., Wang, S., Yang, M.H.: Segflow: Joint learning for video object segmentation and optical flow. In: ICCV (2017)
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. IJCV **70**(1), 41–54 (2006)
7. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: CVPR (2010)
8. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV (2014)
9. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR (2015)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
12. Hui, T.W., Tang, X., Loy, C.C.: LiteFlowNet: A lightweight convolutional neural network for optical flow estimation. In: CVPR (2018)
13. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
14. Jain, S.D., Grauman, K.: Supervoxel-consistent foreground propagation in video. In: ECCV (2014)
15. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. In: CVPR (2017)
16. Khoreva, A., Benenson, R., Ilg, E., Brox, T., Schiele, B.: Lucid data dreaming for object tracking. In: CVPRW (2017)
17. Le, T.N., Nguyen, K.T., Nguyen-Phan, M.H., Ton, T.V., Nguyen, T.A., Trinh, X.S., Dinh, Q.H., Nguyen, V.T., Duong, A.D., Sugimoto, A., Nguyen, T.V., Tran, M.T.: Instance re-identification flow for video object segmentation. In: CVPRW (2017)
18. Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: ICCV (2011)
19. Li, F., Kim, T., Humayun, A., Tsai, D., Rehg, J.M.: Video segmentation by tracking many figure-ground segments. In: ICCV (2013)
20. Li, X., Liu, Z., Luo, P., Loy, C.C., Tang, X.: Not all pixels are equal: difficulty-aware semantic segmentation via deep layer cascade. In: CVPR (2017)
21. Li, X., Qi, Y., Wang, Z., Chen, K., Liu, Z., Shi, J., Luo, P., Tang, X., Loy, C.C.: Video object segmentation with re-identification. In: CVPRW (2017)
22. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: CVPR (2017)
23. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Deep learning markov random field for semantic segmentation. TPAMI (2017)
24. Märki, N., Perazzi, F., Wang, O., Sorkine-Hornung, A.: Bilateral space video segmentation. In: CVPR (2016)

25. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: ICCV (2013)
26. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR (2017)
27. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
28. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv:1704.00675 (2017)
29. Prest, A., Leistner, C., Civera, J., Schmid, C., Ferrari, V.: Learning object class detectors from weakly annotated video. In: CVPR (2012)
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
31. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In: CVPR (2018)
32. Tsai, Y.H., Yang, M.H., Black, M.J.: Video segmentation via object flow. In: CVPR (2016)
33. Valmadre, J., Bertinetto, L., Henriques, J.F., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking (2017)
34. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017)
35. Xiao, F., Jae Lee, Y.: Track and segment: An iterative unsupervised approach for video object proposals. In: CVPR (2016)
36. Xiao, T., Li, S., Wang, B., Lin, L., Wang, X.: Joint detection and identification feature learning for person search. In: CVPR (2017)
37. Xu, C., Xiong, C., Corso, J.J.: Streaming hierarchical video segmentation. In: ECCV (2012)
38. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: CVPR (2016)
39. Yoon, J.S., Rameau, F., Kim, J., Lee, S., Shin, S., Kweon, I.S.: Pixel-level matching for video object segmentation using convolutional neural networks. In: CVPR (2017)
40. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
41. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Chang, H., Torr, P.: Conditional random fields as recurrent neural networks. In: ICCV (2015)