MoNet: Deep Motion Exploitation for Video Object Segmentation

Huaxin Xiao[1,2] Jiashi Feng[2] Guosheng Lin[3] Yu Liu[1] Maojun Zhang[1]
[1]National University of Defense Technology [2]National University of Singapore [3]Nanyang Technological University
{xiaohuaxin, jasonyuliu, mjzhang}@nudt.edu.cn elefjia@nus.edu.sg gslin@ntu.edu.sg

Abstract

In this paper, we propose a novel MoNet model to deeply exploit motion cues for boosting video object segmentation performance from two aspects, i.e., *frame representation learning and segmentation refinement. Concretely, MoNet exploits computed motion cue (*i.e.*, optical flow) to reinforce the representation of the target frame by aligning and integrating representations from its neighbors. The new representation provides valuable temporal contexts for segmentation and improves robustness to various common contaminating factors,* e.g., *motion blur, appearance variation and deformation of video objects. Moreover, MoNet exploits motion inconsistency and transforms such motion cue into foreground/background prior to eliminate distraction from confusing instances and noisy regions. By introducing a distance transform layer, MoNet can effectively separate motion-inconstant instances/regions and thoroughly refine segmentation results. Integrating the proposed two motion exploitation components with a standard segmentation network, MoNet provides new state-of-the-art performance on three competitive benchmark datasets.*

1. Introduction

Given the segmentation mask of a target object in the first frame, semi-supervised Video Object Segmentation (VOS) aims to automatically segment the specified object in subsequent video frames. Recently remarkable progress has been made by CNN-based approaches [3, 5, 13, 15, 26] which generally solve the task in two stages: offline training a segmentation model and online fine-tuning it on the test video.

Conventionally, CNN-based methods [3, 26] ignore the temporal information among adjacent frames and cast VOS as a static image segmentation problem. Such frame-by-frame methods suffer a lot from unconstrained video conditions like deformation, scale variation and motion blur, which lead to large appearance changes of the target object from the initial frame to subsequent ones (see results in Fig. 1 by OSVOS [3] which processes each frame independently). Moreover, new instances with confusing appearance appearing in subsequent frames may fail VOS models in distinguishing the target object from distracting ones (see the bottom example in Fig. 1).

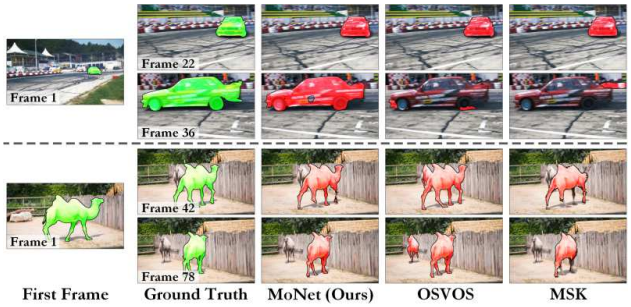

Figure 1. Segmentation results (red masks) of the proposed MoNet, OSVOS [3] and MSK [15] on two video sequences from DAVIS [21], which include several typical challenges for VOS, *e.g.*, appearance change, scale variation (the top example) and confusing instances (the bottom example). MoNet deeply exploits motion cues from adjacent frames and well copes with these challenges, producing better segmentation results than state-of-the-art OSVOS and MSK. Best viewed in color with 4× zoom.

To tackle these challenges, leveraging motion cues (*i.e.*, optical flow in this work) as additional information along the temporal domain becomes necessary for VOS models to enhance segmentation consistency and quality. One simple way is to apply segmentation models to optical flow field [15, 29]. However, in this case, the model performance would be limited by the quality of flow estimation (see results in Fig. 1 by MSK [15] employing RGB image and optical flow as the inputs). To exploit motion cues more effectively, [5, 11] introduce learnable networks to extract motion features from optical flow to complement appearance features, but they learn these two types of features separately, which limits their robustness to various video challenges. Different from simply treating motion as extra inputs or external features, this work attempts to give a new insight into exploiting and utilizing such informative cues better for CNN-based VOS.

First of all, we exploit motion cues to reinforce the learned representation of a target frame. Intense changes in object appearance and scale can bring great difficulties in

segmenting a target object throughout a sequence (see the top example in Fig. 1), as the online fine-tuning only has access to a single labeled frame without foreseeing such variations. An effective way to deal with this challenge is to utilize consecutive motion information about the target object. Thus we propose to integrate the features from adjacent frames into the representation of the target frame. Inspired by the success in video object detection [37, 38], we propose to align the features from adjacent frames, using optical flow to regulate their integration, through a warp layer with bilinear interpolation. Different from directly extracting frame representation from motion domain [5, 11], the motion-aligned representations include necessary appearance information and valuable temporal contexts for normalizing unknown variations, thus benefiting the quality and temporal consistency of VOS results.

Secondly, we exploit motion cues to identify motion-inconsistent instances/regions with confusing appearance, separate the target object from the distractions and improve the segmentation results. This is important for segmenting object in video as new and unexpected similar instances may appear in subsequent frames, which typically confuse and fail existing VOS methods (the bottom example in Fig. 1). To inspect inconsistent motion patterns, we propose a distance transform (DT) layer to separate the target object with notable movement from the background motion. The DT layer measures the connectivity between each location in the optical flow and the background motion using the Minimum Barrier Distance (MBD) [6] and maps the optical flow into a simple foreground/background mask. As an abstract motion prior, the mask is combined with the segmentation prediction to refine the results. Superior to employing fully-supervised CNN-based models to learn motion patterns [29], the DT layer is free of ground truth optical flow to learn a CNN model, and much simpler yet provides comparable performance (see results in Tab. 7).

The proposed two components are integrated into a trainable model, named MoNet, which deeply exploits motion cues in videos and thus addresses the challenging unconstrained conditions better than state-of-the-art VOS methods. We extensively evaluate MoNet on three benchmark datasets, *i.e.*, DAVIS [21], Youtube-Objects [10, 23] and SegTrack-v2 [18], and observe superior performance w.r.t. various metrics.

The main contributions of this paper are three-fold.

- We revive attention to motion cues for solving VOS and advance its exploitation by developing the MoNet model. Results on multiple datasets confirm benefits of more elegantly exploiting motion cues.
- We propose to utilize motion cues to reinforce frame representations by integrating motion-aligned features within the temporal domain, which is shown effective for video object detection but is new to VOS.
- We develop an effective approach to extract segmentation prior directly from motion cues, which highly fits unique requirements of VOS but is ignored by existing solutions. The extracted prior can filter out the distracting instances/regions and purify the segmentation.

2. Related Work

Unsupervised VOS methods aim to segment a primary object without human inputs, by utilizing visual saliency [8, 33] and motion cues [16, 20]. Recently, Tokmakov *et al*. [29] employed synthetic video data to learn a model to segment moving objects from optical flow. Jain *et al*. [11] proposed a two-stream CNN to extract features from input frames and optical flow to jointly segment the object. Based on [29], recurrent units are introduced by [30] to propagate spatial information over time.

This work focuses on semi-supervised VOS where annotation on the first frame is given. Besides some classic methods segmenting objects by minimizing an energy function defined over different constraints [19, 22, 31], recent VOS methods benefit much from adopting CNN. For example, [3] proposed to independently process each frame using CNN without any temporal information. Perazzi *et al*. [15] directly inferred segmentation results from optical flow. Jampani *et al*. [12] proposed a temporal bilateral network to propagate previous masks to the current frame. [13] adopted a three-branch network w.r.t. different segmentation results. In [5], mutual features of object segmentation and optical flow are concatenated at different scales for mutual boosting. Yoon *et al*. [26] formulated video object segmentation as matching a query object at the first frame in subsequent frames. However, the motion cues have not been adequately leveraged. Instead of using motion cues as extra inputs [15, 29] or complementary features [5, 11], we deeply exploit their utilization in frame representation learning and segmentation refinement.

3. The Proposed Model

3.1. Overview

The overall architecture of the proposed MoNet is illustrated in Fig. 2. To learn to exploit motion cues, MoNet receives triple inputs, including the target frame and two adjacent frames. The two adjacent frames are randomly selected within a predefined temporal neighborhood. The triple inputs are passed to a segmentation network [4] and an optical flow estimation network [9], outputting their appearance features and optical flow.

Instead of merging features of the three input frames directly, MoNet aligns features from adjacent frames using their optical flow at first and then integrates them into the target frame feature. Taking in the merged feature, a segmentation model segregates the target frame into fore-

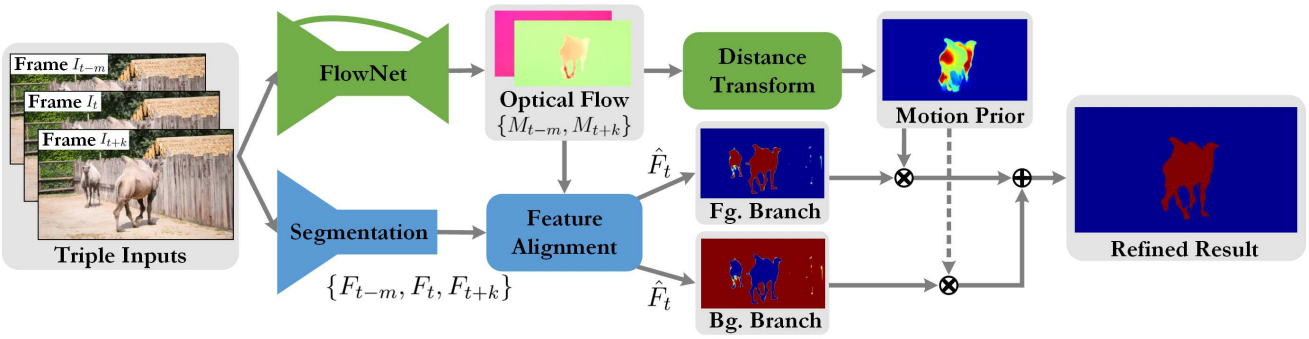

Figure 2. Architecture of the proposed MoNet. The target frame I_t and its two adjacent frames I_{t-m} and I_{t+k} are passed to a segmentation network [4] and a FlowNet [9] respectively. The features F_{t-m} and F_{t+k} from adjacent frames are aligned (by their corresponding optical flow M_{t-m} and M_{t+k}) and combined with the target frame feature F_t, giving a new feature $\hat{F}_t$. Based on $\hat{F}_t$, two separative branches segment the target frame into foreground and background mask. The distance transform layer maps the optical flow to motion prior, which is fused with the foreground/background mask to produce refined object segmentations. Best viewed in color.

ground and background mask. To alleviate susceptibility of the segmentation model to confusing instances (*e.g.*, the target camel on the right and a similar instance on the left in Fig. 2), MoNet exploits motion cues to filter out the distracting instances/regions whose motion is usually in disagreement with the movement of target object. It introduces the distance transform layer to map the estimated optical flow to motion prior, which extracts the moving foreground with distinct motion. MoNet fuses such motion prior to refine its foreground/background segmentation results.

3.2. Aligning Feature with Motion Cues

The features learned from a single frame cannot represent the temporal variation of the target object, which limits VOS performance due to loss of important temporal information. To model short temporal dynamics of a given frame I_t, we propose to include features from two randomly selected adjacent frames I_{t-m} and I_{t+k} within a predefined neighborhood of I_t. The features from I_{t-m} and I_{t+k} complement and enhance the feature of I_t by embedding temporal contexts. However, directly aggregating these features cannot improve VOS performance as expected (see results in Tab. 6). Because the spatial locations of temporal contexts in these features always disagree with the locations of I_t. Inspired by the success in video object detection [37, 38], we propose to align the features from adjacent frames I_{t-m} and I_{t+k} to I_t by exploiting motion cues, before combining them.

Formally, let F_{t-m} denote the feature of frame I_{t-m} output by the segmentation network, and $\hat{F}_{t-m}$ denote its aligned feature w.r.t. frame I_t. Aligning $F_{t-m}(x', y')$ to $\hat{F}_{t-m}(x, y)$ needs correspondence between the location (x, y) in I_t and (x', y') in I_{t-m}. The optical flow map M_{t-m} provides the needed displacement (u, v) pointing from (x, y) in I_t to (x', y') in I_{t-m}. With (u, v), the aligned feature $\hat{F}_{t-m}(x, y)$ can be computed by bilinear interpolation:

$$\hat{F}_{t-m}(x,y) = \theta_1 F_{t-m}(\lfloor x' \rfloor, \lfloor y' \rfloor) + \theta_2 F_{t-m}(\lceil x' \rceil, \lfloor y' \rfloor) + \theta_3 F_{t-m}(\lfloor x' \rfloor, \lceil y' \rceil) + \theta_4 F_{t-m}(\lceil x' \rceil, \lceil y' \rceil),$$

where $(x', y') = (x + u, y + v)$, $\theta_1 = (1 - x' + \lfloor x' \rfloor)(1 - y' + \lfloor y' \rfloor)$, $\theta_2 = (x' - \lfloor x' \rfloor)(1 - y' + \lfloor y' \rfloor)$, $\theta_3 = (1 - x' + \lfloor x' \rfloor)(y' - \lfloor y' \rfloor)$, and $\theta_4 = (x' - \lfloor x' \rfloor)(y' - \lfloor y' \rfloor)$.

The above equation is implemented as a warp layer in MoNet. After feature alignment, three channel-wise weighting vectors are learned to merge the feature F_t from I_t with $\hat{F}_{t-m}$ and $\hat{F}_{t+k}$ as follows:

$$\hat{F}_t = \boldsymbol{w}_{t-m} \otimes \hat{F}_{t-m} + \boldsymbol{w}_t \otimes F_t + \boldsymbol{w}_{t+k} \otimes \hat{F}_{t+k}, \quad (1)$$

where $\otimes$ denotes channel-wise scalar-matrix multiplication.

Eqn. (1) dynamically combines the features along the channel dimension, assigning suitable weights to different channels. After alignment and aggregation, $\hat{F}_t$ includes various tailored temporal information, provides enriched representation of I_t and effectively extends temporal-domain receptive field of the segmentation classifier.

3.3. Distance Transform Layer

When segmenting the target object in a video sequence, the segmentation model may be distracted by some confusing factors (*e.g.*, instances from the same category, similar instances the model had seen during its offline training and visually similar regions) and produces false positive predictions. Usually, the motion of such confusing instances/regions is inconsistent with the movement of target object. To utilize such motion cue to eliminate negative effects of these distractions, we propose to perform MBD-based distance transform [28] on the estimated optical flow map M to obtain relatively clean and robust motion prior. Such prior helps identify the moving object with distinct motion and remove the instances/regions with inconsistent motion patterns as the identified movement.

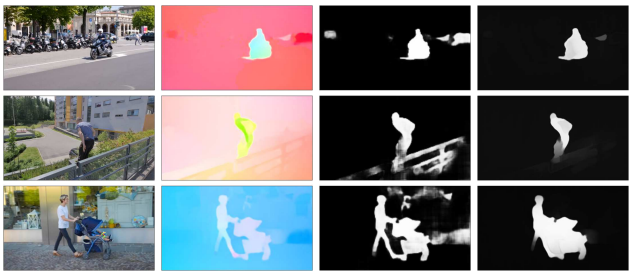

Figure 3. Comparison on moving object extraction from optical flow between MPNet [29] and the proposed DT layer. Although the MPNet provides sharper details about the moving object, it is susceptible to minor motion in optical flow. In contrast, the DT layer is more robust to complicated motions and gives smoother and better extracted objects. Best viewed in color.

Formally, for each spatial location (*i.e.*, pixel) l in M, a path $\pi_l = \langle \pi_l(1), \cdots, \pi_l(n) \rangle$ connecting to l is defined as a sequential collection of its spatially neighboring locations, where n is the number of considered neighbors and takes a typical value of 4. The distance transform $D(\cdot)$ on M is to map each location l to a distance value w.r.t. a target set S with the minimum path cost:

$$D(l) = \min_{\pi_l \in \prod_{S,l}} \mathcal{P}(\pi_l), \tag{2}$$

where $\mathcal{P}(\cdot)$ denotes a path cost function and S denotes a set of predefined locations. $\prod_{S,l}$ denotes the set of all paths that connect S and l.

In particular, we calculate the path cost function $\mathcal{P}(\cdot)$ in DT using the minimum barrier distance [28], for its appealing robustness [6, 36]. The MBD-based path cost function at location l is defined as

$$\mathcal{P}(\pi_l) = \max_{i \in \{1,\ldots,n\}} \|M(\pi_l(i))\| - \min_{i \in \{1,\ldots,n\}} \|M(\pi_l(i))\|, \tag{3}$$

where $\|M(\cdot)\|$ is the flow magnitude of a location in M.

We use the FastMBD algorithm [36] to solve the MBD-based DT in Eqns. (2) and (3) approximately, which visits each location l of M in a raster scan or inverse raster scan order. At each scanning step , only half of adjacent locations of l contribute to updating the distance map $D(l)$:

$$D(l) \leftarrow \min\{D(l), \mathcal{P}(\pi_{h,l})\}, \tag{4}$$

where h denotes an adjacent location of l and $\pi_{h,l}$ denotes the path combining π_h with connections from h to l. The path cost function $\mathcal{P}(\pi_{h,l})$ is defined as

$$\mathcal{P}(\pi_{h,l}) = \max\{U(h), \|M(l)\|\} - \min\{V(h), \|M(l)\|\}, \tag{5}$$

where $U(h)$ and $V(h)$ denote the largest and smallest value of $\|M(\pi_h)\|$ respectively. The computation of Eqn. (5) is efficient by caching U and V of all locations.

We implement the FastMBD algorithm as a distance transform layer in MoNet. The DT layer takes the flow map M as input and outputs the distance map D, which measures connectivity of each location in M to the predefined S. As we aim to separate the target object motion from cluttered background motion, with the common assumption that background motion is usually connected to optical flow borders, we define S to include locations along the borders of optical flow M.

For each M, the DT layer visits all the locations of M twice, *i.e.*, one raster and the other inverse raster scanning, which are sufficient to perform well without significant computation overhead. For a given frame I_t, we average and normalize the distance maps D_{t-m} and D_{t+k} from M_{t-m} and M_{t+k} into the final D_t. Path cost in Eqn. (5) is calculated by the flow magnitude, where larger path cost indicates more inconsistent motion between the location l and the locations in S. Thus a larger value of $D_t(l)$ demonstrates a lower probability for l corresponding to background motion. Namely, the distance map D provides an abstract motion prior for the foreground object.

Considering the complex nature of optical flow, MPNet [29] learns a complicated encoder-decoder network to extract the moving object from the optical flow. In this paper, we use DT layer to solve this problem in a much simpler way. Fig. 3 qualitatively compares the DT layer with the CNN-based MPNet [29] on segmenting foreground moving objects. The MPNet is trained from synthetic sequences with *ground truth* optical flow. Thus it provides sharper details about the moving object but is susceptible to minor motion. In contrast, the DT layer is more robust to various motion complexities in optical flow and provides smoother and better motion prior. Moreover, the DT layer is fully unsupervised.

3.4. Object Segmentation

Based on the aligned feature, two segmentation branches [2] are designed to predict the mask of foreground and background respectively. To remedy the possible unreliabilities in the motion prior (as it completely derives from estimated optical flow without context information), we employ two complementary classifiers to integrate the motion prior from the DT layer and they respond to normal and inverse motion prior respectively.

Formally, the final prediction is made upon the aligned feature $\hat{F}$ and motion prior D as follows:

$$\mathcal{C}_s(\hat{F}, D) = D \otimes \mathcal{C}_f(\hat{F}, W_f) + D \otimes (1 - \mathcal{C}_b(\hat{F}, W_b)),$$

where $\mathcal{C}_s(\hat{F}, D)$ is the segmentation classifier, $\mathcal{C}_f(\hat{F}, W_f)$ and $\mathcal{C}_b(\hat{F}, W_b)$ denote the foreground and background prediction branch respectively, with parameters W_f and W_b. The above segmentation classifier $\mathcal{C}_s$ is trained by minimiz-

	Metric	Semi-supervised									Unsupervised				
		MoNet	OSVOS	MSK	SFL	CTN	VPN	PLM	OFL	FCP	LVO	ARP	FSEG	MPNet	SFL
$\mathcal{J}$	Mean $\mathcal{M}\uparrow$	**84.7**	79.8	79.7	76.1	75.5	70.2	70.0	68.0	58.4	75.9	**76.2**	70.7	70.0	67.4
	Recall $\mathcal{O}\uparrow$	**96.8**	93.6	93.1	88.2	89.0	82.3	-	75.6	71.5	89.1	**91.1**	83.5	85.0	81.4
	Decay $\mathcal{D}\downarrow$	6.4	14.9	8.9	12.1	14.4	12.4	-	26.4	**-2.0**	**0.0**	7.0	1.5	1.3	6.2
$\mathcal{F}$	Mean $\mathcal{M}\uparrow$	**84.8**	80.6	75.4	76.0	71.4	65.6	62.0	63.4	49.2	**72.1**	70.6	65.3	65.9	66.7
	Recall $\mathcal{O}\uparrow$	**94.7**	92.6	87.1	85.5	84.8	69.0	-	70.4	49.5	83.4	**83.5**	73.8	79.2	77.1
	Decay $\mathcal{D}\downarrow$	8.6	15.0	9.0	10.4	14.0	14.4	-	27.2	**-1.1**	**1.3**	7.9	1.8	2.5	5.1
$\mathcal{G}$	Mean $\mathcal{M}\uparrow$	**84.7**	80.2	77.6	76.1	73.5	67.8	66.0	65.7	53.8	**74.0**	73.4	68.0	67.9	67.1

Table 1. Quantitative comparison of the unsupervised and semi-supervised models on DAVIS validation set. The up-arrow ↑ means larger is better while the down-arrow ↓ means smaller is better.

ing the following balanced binary cross-entropy loss [35]:

$$\begin{aligned}\mathcal{O}(W) = &-\beta \sum_{j\in Y_+} \log \mathcal{C}_s(Y_j = 1|\hat{F}, D, W) \\ &-(1-\beta)\sum_{j\in Y_-} \log \mathcal{C}_s(Y_j = 0|\hat{F}, D, W),\end{aligned} \tag{6}$$

where Y is the ground truth and divided into the background label map Y_- and the foreground label map Y_+. $\beta = |Y_-|/(|Y_-| + |Y_+|)$. $|Y_-|$ and $|Y_+|$ denote the number of labels in Y_- and Y_+ respectively. W denotes the parameters of the whole network, including W_f, W_b and the parameters of the segmentation network in Fig. 2.

3.5. Implementation Details

We focus on exploiting motion cues to improve VOS performance. Therefore, extensive engineering on the segmentation architecture is out of the scope of this work. We use the well established VGG16 [27] based DeepLab architecture [4] as the backbone segmentation network without any further modification. Each segmentation branch adopts the structure of atrous spatial pyramid pooling [4]. The CNN-based FlowNet2[1] [9] is employed to online estimate the optical flow. The sampling neighborhood of a given frame is set to 3 frames. For each triple input, the `conv5_3` feature is extracted and aligned by Eqn. (1).

Before training on video sequences, we pretrain the segmentation network with static images from PASCAL VOC 2012 dataset [7]. At the stage of offline training on video sequences, we first fine-tune the pretrained model with the feature alignment. The $\boldsymbol{w}_{t-m}$, $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t+k}$ in Eqn. (1) are initialized as 0, 1 and 0. The segmentation network with Fg/Bg branches are trained together on the training set of DAVIS, by SGD with learning rate 5×10^{-8} for 20K iterations. Then the motion prior estimated by Eqn. (4) is used to train the final offline model. The learning rate is set to 1×10^{-8} for 10K iterations.

[1]To balance the accuracy and running speed, we adopt the thin version of FlowNet2, *i.e.*, *FlowNet2-css-ft-sd*, for the estimation of optical flow.

When performing inference on a specific video sequence, the model is online fine-tuned on the first frame, from the offline pretrained model, and directly applied to subsequent frames. Considering randomness of selecting adjacent frames, we repeat inference for a specific target frame multiple times, with equal neighborhood range. Then we average the predictions into the final segmentation for the target frame. The segmentation results are post-processed by a fully-connected CRF [17].

Our proposed MoNet is implemented by the publicly available Caffe library [14]. All the experiments and analyses are conducted on a Nvidia Titan X GPU and a 6 core Intel i7-4930K CPU 3.4GHz.

4. Experiments

4.1. Experimental Setup

Datasets The proposed MoNet is evaluated on three video object segmentation datasets, *i.e.*, DAVIS [21], Youtube-Objects [10, 23] and SegTrack-v2 [18]. DAVIS consists of 50 high-resolution video sequences with 3,455 frames in total. Each video is annotated with multiple attributes such as deformation, fast motion and scale variation, for comprehensively analyzing model performance. In Youtube-Objects, there are 126 video sequences with more than 20,000 frames in total, divided into 10 common object categories. SegTrack-v2 is a relatively small dataset composed of 14 video sequences.

Evaluation Metrics We adopt two conventional evaluation metrics from [21], region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$. The region similarity $\mathcal{J}$ is calculated by the intersection-over-union between the predicted segmentation and the ground truth. The contour accuracy $\mathcal{F}$ is defined as the F-measure between the contour points of the predicted segmentation and the ground truth. Each metric is quantized by three different statistics: mean $\mathcal{M}$, recall $\mathcal{R}$ and decay $\mathcal{D}$ as described in [21]. In addition, an overall measure (global mean $\mathcal{G}$) of the performance is defined as the average of $\mathcal{M}_\mathcal{J}$ and $\mathcal{M}_\mathcal{F}$.

Category	MoNet	OSVOS	MSK	OFL	JFS	BVS
Aeroplane	89.2	88.2	86.0	**89.9**	89.0	86.8
Bird	**88.8**	85.7	85.6	84.2	81.6	80.9
Boat	**81.1**	77.5	78.8	74.0	74.2	65.1
Car	**81.9**	79.6	78.8	80.9	70.9	68.7
Cat	**76.7**	70.8	70.1	68.3	67.7	55.9
Cow	**82.0**	77.8	77.7	79.8	79.1	69.9
Dog	81.1	**81.3**	79.2	76.6	70.3	68.5
Horse	**74.4**	72.8	71.7	72.6	67.8	58.9
Motorbike	**77.2**	73.5	65.6	73.7	61.5	60.5
Train	**85.2**	75.7	83.5	76.3	78.2	65.2
Mean $\mathcal{J}\uparrow$	**81.7**	78.3	77.7	77.6	74.0	68.0

Table 2. Quantitative comparison of per-category region similarity $\mathcal{J}$ on Youtube-Objects dataset.

Metric	MoNet	MSK	OFL	OSVOS	BVS
Mean $\mathcal{J}\uparrow$	**72.4**	70.3	67.5	65.4	58.4

Table 3. Quantitative comparison of region similarity $\mathcal{J}$ on SegTrack-v2 dataset.

Baselines We compare the proposed MoNet with 6 latest and state-of-the-art CNN-based models: OSVOS [3], MSK [15], SFL [5], CTN [13], VPN [12] and PLM [26], and 4 non-CNN-based methods: OFL [31], FCP [22], JFS [25] and BVS [19]. We also compare with unsupervised models: LVO [30], ARP [16], FSEG [11] and MPNet [29].

4.2. Comparison with State-of-the-arts

DAVIS Tab. 1 shows the results of compared methods on the DAVIS validation set [21]. Overall, the proposed MoNet performs the best. In terms of $\mathcal{M}_\mathcal{G}$, $\mathcal{M}_\mathcal{J}$ and $\mathcal{M}_\mathcal{F}$, the proposed MoNet improves the state-of-the-art OSVOS [3] by 5.6%, 6.1% and 5.2% respectively. MSK [15] and SFL [5] adopt motion cues as extra inputs and complementary features respectively. MoNet outperforms them by 9.3% and 11.3% respectively w.r.t. $\mathcal{M}_\mathcal{G}$. This proves MoNet can better exploit motion cues. Very recently, OnAVOS [32] improves the OSVOS using online adaption and achieves 85.5% w.r.t. $\mathcal{M}_\mathcal{G}$. However, OnAVOS uses a much better segmentation architecture [34] than the VGG16-based network [4] used in our method. The baseline performance of OnAVOS is 80.3% on $\mathcal{M}_\mathcal{G}$. In contrast, our baseline model performance is only 75.7%, as shown in Tab. 6. Our proposed MoNet improves its baseline by a margin of 11.9%, which is more significant than OnAVOS over its baseline (6.5%). As extensive network architecture engineering is out of the scope of this work, we will update MoNet with a stronger baseline in the future.

Youtube-Objects and SegTrack-v2 Tab. 2 reports per-category mean $\mathcal{J}$ on the Youtube-Objects dataset [10, 23]. The proposed MoNet achieves the best performance in 8 out of 10 categories, except for *Aeroplane* and *Dog*. In particular, MoNet outperforms the state-of-the-art OSVOS [3] and MSK [15] by a margin as large as 4.3% and 5.2% respectively for mean $\mathcal{J}$. On the SegTrack-v2 dataset [18], MoNet gives the best mean $\mathcal{J}$ of 72.4%, as shown in Tab. 3. Overall, the proposed MoNet provides new state-of-the-art for CNN-based VOS in terms of region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$ and global mean $\mathcal{G}$ consistently.

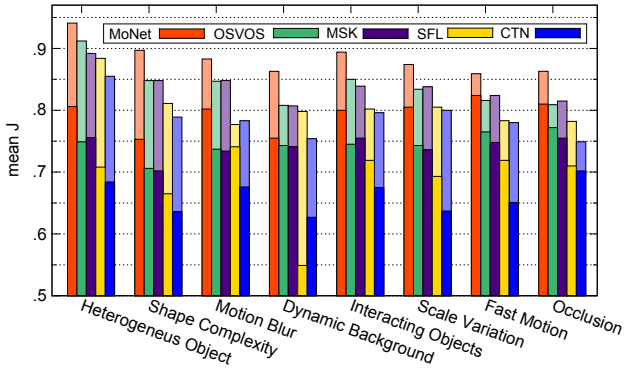

Figure 4. Attribute-based analysis on DAVIS validation set. We compare the proposed MoNet with 4 top-performing CNN-based methods, *i.e.*, OSVOS, MSK, SFL and CTN. For each method, the dark color bin quantizes the mean $\mathcal{J}$ over all sequences with specified attribute labeled in x-axis, and the light color bin illustrates the performance gain on video sequences without the specified challenging attribute. Best viewed in color.

Attribute-based Performance Analysis To more comprehensively analyze model performance under difference video challenges, we perform attribute-based analysis on the DAVIS validation set. Each video is annotated with one or more attributes and each attribute features a specific challenging condition. Based on the results in Tab. 1, we select 4 top-performing semi-supervised approaches for comparison, *i.e.*, OSVOS [3], MSK [15], SFL [5] and CTN [13]. The results are plotted in Fig. 4. For each approach, the dark color bin corresponds to the mean $\mathcal{J}$ over all sequences with the specific attribute (*e.g.*, *Shape Complexity*), and the light color bin quantizes the performance gain on the video sequences without that attribute. Fig. 4 presents performance with the most influential 8 attributes, including *Heterogeneus Object*, *Shape Complexity*, *Motion Blur*, *etc*. The proposed MoNet has the best performance (79.4%) on the video sequences with these 8 attributes, while the mean $\mathcal{J}$ of OSVOS and MSK is only 74.5% and 74.1% respectively. MoNet presents the most stable performance—when discarding these attributes, it has the smallest performance difference in mean $\mathcal{J}$. Namely, MoNet is more robust to various video challenges.

Running Time Tab. 4 compares the per-frame running time of different CNN-based models. For each model, we

Method	MoNet	OSVOS	MSK	SFL	CTN
Per-frame (s)	14.1	∼**5.0**	12.0	7.9	30.0
Mean $\mathcal{G}$ ↑	**84.7**	80.2	77.5	76.1	73.5

Table 4. Average per-frame running time of fine-tuning and inferring a DAVIS sequence. Pre- and post-process are considered.

NBHD	Variant	Mean $\mathcal{J}$ ↑	Mean $\mathcal{F}$ ↑	Mean $\mathcal{G}$ ↑
Present	Baseline	75.3	76.2	75.7
Only Past	+ FA	77.9 +2.6	82.1 **+5.9**	79.9 **+4.2**
	+ FA&MP	81.5 **+3.6**	**84.9** +2.8	83.2 +3.3
	+ CRF [17]	**84.3** +2.8	84.6 -0.3	**84.5** +1.3
Past & Future	+ FA	78.2 +2.9	82.3 **+6.1**	80.2 **+4.5**
	+ FA&MP	82.0 **+3.8**	**85.5** +3.2	83.8 +3.6
	+ CRF [17]	**84.7** +2.7	84.8 -0.7	**84.7** +0.9

Table 5. Ablation study on DAVIS validation set. NBHD denotes the temporal neighborhood of a target frame. Present, Past, Future denotes the NBHD from target frame, preceding frame and subsequent frame respectively. FA denotes the feature alignment while MP denotes the motion prior.

report the average time of fine-tuning and inferring on a DAVIS sequence with the resolution of 480×854 pixels. The proposed MoNet has similar running time as the MSK while providing better performance than the MSK. The DT layer is conducted on the $1/4$ down-sampled optical flow map and takes about 0.1 seconds to estimate the motion prior. It totally takes about 0.6 seconds for MoNet to infer a 480×854 frame.

4.3. Ablation Study

Tab. 5 summarizes the contributions of feature alignment, motion prior and fully-connected CRF [17] to the performance of MoNet. The baseline in Tab. 5 is the DeepLab [4] network trained on the PASCAL VOC 2012 dataset [7] and the DAVIS [21]. The baseline only uses the present frame as input, and the predictions of its foreground and background branches are averaged to generate the segmentation result. We also evaluate two variants of MoNet. One only samples the adjacent frames from preceding ones (Only Past), and the other samples from both preceding and subsequent frames (Past & Future). For both variants, *+FA* denotes training the baseline with the component of aligning features from sampled adjacent frames, which brings 4.0% improvement over the baseline in terms of $\mathcal{M}_\mathcal{G}$. *+FA&MP* means employing the DT-based motion prior to refine the results of *+FA*, which brings another 3.0% enhancement. Sampling from subsequent frames only gives 0.6% improvement, which indicates the strong generalization ability of MoNet to exploit both historical and subsequent motion. The CRF post-process [17] increases the performance by another 1% in terms of $\mathcal{M}_\mathcal{G}$.

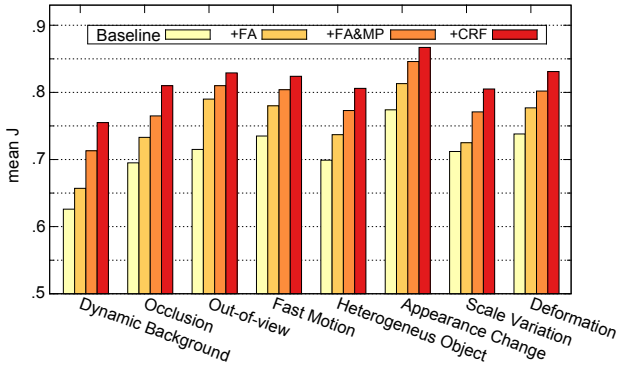

Figure 5. Attribute-based component analysis on the DAVIS validation set. We compare the mean $\mathcal{J}$ performance of different components, *i.e.*, feature alignment (*+FA*), motion prior (*+FA&MP*) and fully-connected CRF (*+CRF*), under various video attributes.

Attribute-based Component Analysis To understand which component in Tab. 5 (Past & Future variant) is helpful to providing robustness to various video challenging conditions, we evaluate and summarize attribute-based performance of these components in Fig. 5. We present the top 8 attributes which present greatest performance improvement over the baseline. The component *+FA* is most beneficial for addressing attributes including *Out-of-view*, *Fast Motion*, *Heterogeneous Object*, *Appearance Change* and *Deformation*. We attribute this benefit of feature alignment to incorporating valuable temporal information. The motion prior eliminates motion-inconsistent regions and offers great robustness to the long temporal range variation, thus it helps a lot in addressing the attributes of *Scale Variation* and *Dynamic Background*.

Feature Alignment Tab. 6 reports ablation studies on effects of aligning different features (*i.e.*, `conv4_3` and `conv5_3`) and effects of varying temporal neighborhood ranges (denoted as #NBHD). From the results for the second and third variants in Tab. 6, one can observe the aligning feature `conv5_3` is better than `conv4_3`. Without refinement using motion prior, #NBHD=5 with the feature `conv5_3` performs the best. Increasing #NBHD (to 10 frames) leads to performance drop. The fourth variant directly combines the triple features without motion alignment, which decreases the performance significantly, proving motion-based feature alignment is effective and necessary. The quality of the motion prior extracted from the optical flow depends on #NBHD. Therefore, from the last variant in Tab. 6, we observe that using motion prior improves the performance most when using 3 frames. It is understandable the motion prior is more sensitive to the sampling range than feature alignment, as feature alignment can be stabilized by the weights in Eqn. (1) while motion prior is directly estimated from the optical flow.

Variant	Baseline	conv4_3 with FA				conv5_3 with FA				conv5_3 w/o FA				conv5_3 with FA&MP			
#NBHD	0	1	3	5	10	1	3	5	10	1	3	5	10	1	3	5	10
Mean $\mathcal{J}\uparrow$	75.3	**77.6**	77.1	77.2	76.4	77.2	78.2	**79.2**	77.5	76.2	**76.6**	76.6	74.8	81.6	**82.0**	81.9	79.7
Mean $\mathcal{F}\uparrow$	76.2	79.9	81.8	**82.0**	80.1	80.3	82.3	**82.7**	81.6	79.4	**79.6**	78.3	77.4	83.8	**85.5**	84.8	82.0
Mean $\mathcal{G}\uparrow$	75.7	78.7	79.5	**79.6**	78.3	79.0	80.2	**80.9**	79.6	77.8	**78.1**	77.4	76.1	82.7	**83.8**	83.4	80.9

Table 6. Ablation study on the feature alignment. #NBHD denotes the range of temporal neighborhood. FA denotes the feature alignment while MP denotes the motion prior. All experiments are conducted on DAVIS validation set.

Method	Flow Used	Mean $\mathcal{J}\uparrow$
MPNet	LDOF [1]	52.4
DT Layer	LDOF [1]	50.8
MPNet	EpicFlow [24]	56.9
DT Layer	EpicFlow [24]	55.4
MPNet + FA (#NBHD=1)	FlowNet2 [9]	81.8
DT Layer + FA (#NBHD=1)	FlowNet2 [9]	81.6

Table 7. Comparison of the proposed DT Layer with the MPNet [29] on DAVIS validation set.

Distance Transform Layer To investigate effectiveness of the proposed DT layer in MoNet, we compare it with the CNN-based MPNet [21], which is trained on the synthetic videos with *ground truth* optical flow and applied on the real-world videos with estimated optical flow. The first four rows of Tab. 7 report their results on the DAVIS validation set with different optical flow computation methods, *i.e.*, LDOF [1] and EpicFlow [24]. The proposed DT layer gives comparable performance as the MPNet although the DT layer is *unsupervised* and much simpler. Furthermore, we adopt the MPNet to estimate the motion prior for the proposed MoNet. Due to the limit of memory capacity, we cannot online estimate the motion prior with the MPNet. Thus we set the temporal neighborhood to be 1 frame (#NBHD=1) and offline estimate the motion prior by MPNet. As shown in the last two rows of Tab. 7, the motion prior by DT layer has similar performance as the one by MPNet, which indicates the DT layer works sufficiently well for MoNet. Besides, the DT layer can be easily extended to larger temporal neighborhood.

4.4. Qualitative Results

Fig. 6 shows example segmentation results of the proposed MoNet on DAVIS [21], Youtube-Objects [10, 23] and SegTrack-v2 [18]. In the figure, the first column shows the first frame of the video sequence along with segmentation annotation (green masks). The other columns show the segmentation results (red masks) by MoNet. The example sequences feature typical video challenges, *e.g.*, object deformation, fast motion, scale variation and appearance change. The proposed MoNet can cope with these challenges well and produce robust and accurate segmentation results.

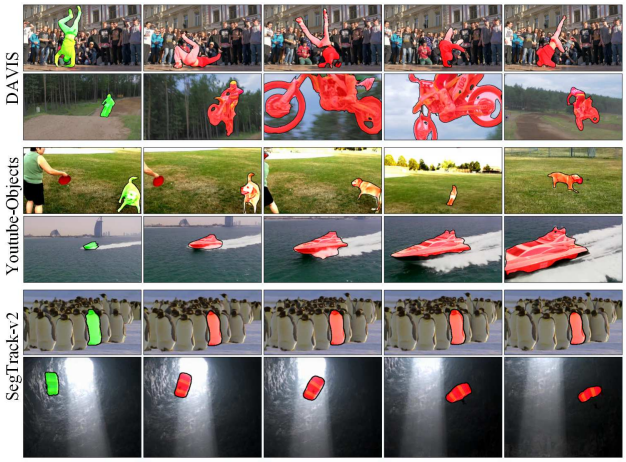

Figure 6. Qualitative results of the proposed MoNet on the DAVIS, Youtube-Objects and SegTrack-v2. The first column is the first frame of a specific sequence with its corresponding annotation (green masks). The other columns are the segmentation results (red masks) by our MoNet. Best viewed in color with 3× zoom.

5. Conclusion

This paper presents a novel trainable network *MoNet* to effectively utilize motion cues to reinforce video frame representation and refine segmentation results. Extensive experiments on various datasets demonstrate that these two exploitation strategies of motion cues are effective and offer superior performance over existing motion utilization, *e.g.*, taking motion cues as extra input [15] or supportive features [5, 11]. A distance transform layer is adopted to filter out the motion-inconsistent instances/regions, which has not been considered in existing works. We also validate the effectiveness of the DT layer with comparison to a CNN-based moving object segmentation method [29].

Acknowledgments

Huaxin Xiao was supported by the China Scholarship Council under Grant 201603170287. Jiashi Feng was partially supported by NUS startup R-263-000-C08-133, MOE Tier-I R-263-000-C21-112, NUS IDS R-263-000-C67-646 and ECRA R-263-000-C87-133.

References

[1] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *TPAMI*, 33(3):500–513, 2011. 8

[2] S. Caelles, Y. Chen, J. Pont-Tuset, and L. Van Gool. Semantically-guided video object segmentation. *arXiv:1704.01926*, 2017. 4

[3] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 2, 6

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 2, 3, 5, 6, 7

[5] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017. 1, 2, 6, 8

[6] K. C. Ciesielski, R. Strand, F. Malmberg, and P. K. Saha. Efficient algorithm for finding the exact minimum barrier distance. *CVIU*, 123:53–64, 2014. 2, 4

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2014. 5, 7

[8] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 2

[9] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 2, 3, 5, 8

[10] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. 2, 5, 6, 8

[11] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017. 1, 2, 6, 8

[12] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *CVPR*, 2017. 2, 6

[13] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident network. In *CVPR*, 2017. 1, 2, 6

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014. 5

[15] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 1, 2, 6, 8

[16] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017. 2, 6

[17] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 5, 7

[18] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 2, 5, 6, 8

[19] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 2, 6

[20] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 2

[21] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 1, 2, 5, 6, 7, 8

[22] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015. 2, 6

[23] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2, 5, 6, 8

[24] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 8

[25] N. Shankar Nagaraja, F. R. Schmidt, and T. Brox. Video segmentation with just a few strokes. In *ICCV*. 6

[26] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017. 1, 2, 6

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 5

[28] R. Strand, K. C. Ciesielski, F. Malmberg, and P. K. Saha. The minimum barrier distance. *CVIU*, 117(4):429–437, 2013. 3, 4

[29] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 1, 2, 4, 6, 8

[30] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 2, 6

[31] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 2, 6

[32] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 6

[33] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 2

[34] Z. Wu, C. Shen, and A. v. d. Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv:1611.10080*, 2016. 6

[35] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 5

[36] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 fps. In *ICCV*, 2015. 4

[37] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, 2017. 2, 3

[38] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *CVPR*, 2017. 2, 3