2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Meta Learning with Differentiable Closed-form Solver for Fast Video Object Segmentation

Yu Liu[1] Lingqiao Liu[1] Haokui Zhang[2] Hamid Rezatofighi[1] Qingsen Yan[1] Ian Reid[1]

Abstract— Video object segmentation plays a vital role to many robotic tasks, beyond the satisfied accuracy, quickly adapt to the new scenario with very limited annotations and conduct a quick inference are also important. In this paper, we are specifically concerned with the task of fast segmenting all pixels of a target object in all frames, given the annotation mask in the first frame. Even when such annotation is available, this remains a challenging problem because of the changing appearance and shape of the object over time. In this paper, we tackle this task by formulating it as a meta-learning problem, where the base learner grasping the semantic scene understanding for a general type of objects, and the meta learner quickly adapting the appearance of the target object with a few examples. Our proposed meta-learning method uses a closed form optimizer, the so-called "ridge regression", which has been shown to be conducive for fast and better training convergence. Moreover, we propose a mechanism, named "block splitting", to further speed up the training process as well as to reduce the number of learning parameters. In comparison with the state-of-the art methods, our proposed framework achieves significant boost up in processing speed, while having highly comparable performance compared to the best performing methods on the widely used datasets. Video demo can be found here [1].

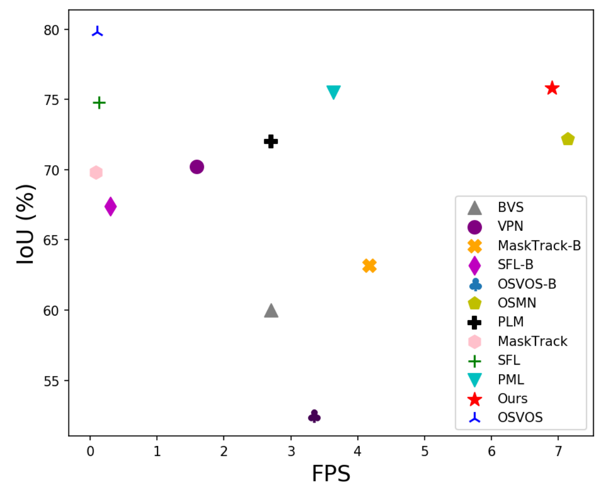

Fig. 1. A comparison of the quality and the speed of previous video object segmentation methods on DAVIS2016 benchmark. We visualize the intersection-over-union (IoU) with respect to the frames-per-second (FPS).

I. Introduction

The goal of video object segmentation is to distinguish an object of interest over video frames from its background at the pixel level. Fast and accurate video object segmentation plays an important role in robotics research and has various applications, including, but not limited to, robotic vision [18], film making [11], public surveillance [39].

In contrast to many vision tasks such as image classification [15], face recognition [26] and object detection [29], [12] for which the performance of the algorithms reach to the point of being suitable for real-world applications, the performance of video object segmentation algorithms are still far beyond the human performance [28]. This is mainly because to acquire the dense pixel-wise labeling is super expensive, thus this problem does not benefit from availability of a massive corpus of training data, unlike the other aforementioned tasks.

Recently, deep learning-based approaches have shown promising progresses on video object segmentation task [3], [37], [22]. However, they still struggle to satisfy both good accuracy and fast processing inference. In this paper, we aim to bridge this gap.

Inspired by the meta-learning method which is proposed for image classification task [2], we propose an intuitive yet powerful algorithm for video object segmentation, in which the reference frame is available with its annotated mask. In addition, we also propose block splitting to speed up the matrix computation, significantly improving the efficiency of the whole framework. Our objective is to train a system that can "adapt" this annotation information to subsequent frames in a fast yet flexible way at inference time. Specifically, at inference time the reference frame (i.e. one with ground-truth annotation) is mapped to vector in a high dimensional embedding space $X = \phi(I)$ using a CNN ϕ.

We then determine using ridge regression [23], the coefficients of a matrix W that best maps X to the ground truth, $Y = WX$. W is then the video-specific "adaptor", and it maps the feature vectors for every query image (i.e. every other image in the video sequence) to their predicted segmentation masks. Training comprises the process of learning the mapping $\phi(.)$ by presenting the network with pairs of images (from a variety of videos but with each pair coming from the same video), each with ground-truth annotation, and back-propagating the loss through ϕ. This is illustrated in Fig. 3 and described

[1] Y. Liu, L. Liu, H. Rezatofighi, Q. Yan and I. Reid are with School of Computer Science, The University of Adelaide, 5005, North Terrace, SA yu.liu04@adelaide.edu.au
[2] H. Zhang is with School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, 710072, China hkzhang1991@mail.nwpu.edu.cn

[1]Video demo can be found in the following link: https://www.youtube.com/watch?v=btJlqLj0-nc

Fig. 2. Example result of our technique: The segmentation of the first frame (red) is used to learn the model of the specific object to track, which is segmented in the rest of the frames independently (green). One every 10 frames shown of 50 in total.

in more detail later in the paper.

We observe that a limitation of the proposed approach is that the ridge regression scales poorly with the dimension of the feature feature produced by $\phi(.)$ because the optimization requires an huge matrix inversion. We address this through the use of a "block splitting" method that approximates the matrix in block diagonal form, meaning the inversion can be done much more efficiently.

Our main contributions are three-fold:

- A meta-learning based method for video object segmentation is developed, using a closed form solver (ridge regression) as the internal optimizer. This is capable of performing fast gradient back-propagation and can adapt to previously unseen objects quickly with very few samples. Inference (i.e. segmentation of the video) is a single forward pass per frame with no need for fine-tuning or post-processing.
- Ridge regression in high-dimensional feature spaces can be very slow, because of the need to invert a large matrix. We address this by using a novel block splitting mechanism, which greatly accelerates the training process without damaging the performance.
- We demonstrate the state-of-the-art video segmentation accuracy relative to all others methods of comparable processing time, and even better accuracy than many slower ones (see Fig. I).

II. Related Works

A. Semi-supervised Video Object Segmentation

The goal of video object segmentation is to 'cutout' the target object(s) from the entire input video sequence. For semi-supervised video object segmentation, the annotated mask of the first frame is given, and the algorithm is designed to predict the masks of the rest frames in the video. There are three categories in this spectrum. The first one, which include MSK [27], MPNVOS [33] etc, is to use optical flow to track the mask from the previous frame to the current frame. Similarly, the second category formulates the optical flow and segmentation in two parallel branches, and utilizes the predicted mask from the previous frame as a guidance, some representatives are Segflow [7], OSNM [40] etc. The final class which keeps the state-of-the-art performance on DAVIS benchmark [28] is to try to over-fit the appearance of the target object(s), and expect the method can generalize in the subsequent frames. Specifically, OSVOS [3] uses one-shot learning mechanism to conduct fine-tuning on the first frame of test video to capture the appearance of the target object(s), and conduct inference on the rest frames. The limitations of OSVOS are: (1) it can not adapt to the unseen parts (2) when dramatic changes of appearance happen in subsequent frames, the method's performance significantly degrade. Inspired by the overall design principle of OSVOS, there are some following methods which employ various additional ingredients to improve the segmentation accuracy. Such as OSVOS-S [22], OnVOS [37], but they all suffer the the limitation of super-slow for inference.

In this paper, we mainly target to fast video object segmentation, since no optical flow and fine-tuning processes are used, the proposed method is appropriate for real-world applications.

B. Meta Learning

Meta learning is also named learning to learn [31], [24], it is an alternative to the de-facto solution that has emerged in deep learning of pre-training a network using a large, generic dataset (eg ImageNet [8]) followed by fine-tuning with a problem-specific dataset. Meta-learning aims to replace the fine-tuning stage (which can still be very expensive) by training a network that has a degree of plasticity so that it can adapt rapidly to new tasks. For this reason it has become a very active area recently, especially with regard to one-shot and few-shot learning problems [16], [9].

Recent approaches for meta-learning can be roughly put into three categories: (i) metric learning for acquiring similarities; (ii) learning optimizers for gaining update rules; and (iii) recurrent networks for reserving the memory. In this work, we adopt the meta-learning algorithm that belongs to the category of learning optimizers. Specifically, inspired by [2] which was originally designed for image classification, we adopt ridge regression, which is a closed-form solution to the optimization problem. The reason for using it is because, compared with the widely-used SGD [17] in CNNs, ridge regression can propagate gradient efficiently, which is matched with the goal of fast mapping. Through extensive experiments, we demonstrate that the proposed method is in the first echelon regarding to speed for fast video object segmentation, while obtaining more accurate results without any post-processing.

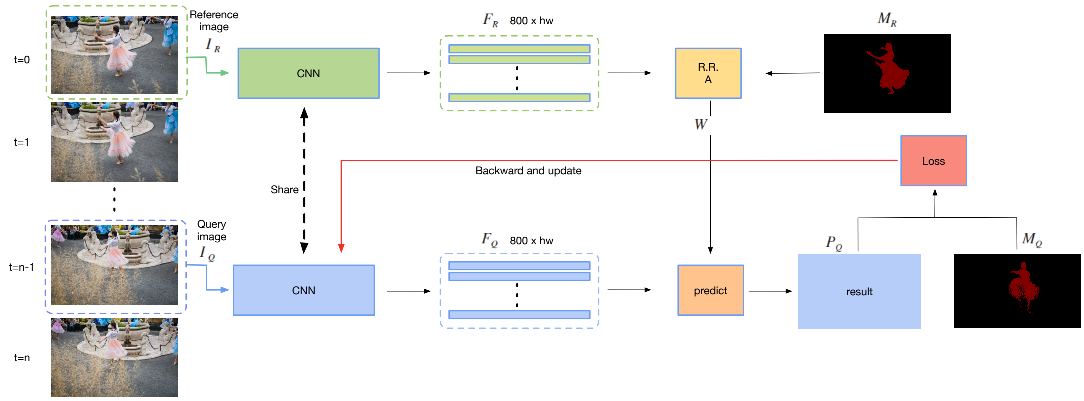

Fig. 3. Workflow of the proposed method. An image pair sampled from the same video as the input to the network. The first image I_R and its annotation M_R as the reference frame, and the second image I_Q and its annotation M_Q (or prediction P_Q during inference) as the query frame. The image pair first passes through the feature extractor (DeepLabv2 [4] with ResNet101 [13]) to compute a 800D embedding tensor F_R, F_Q. Then a mapping matrix W between F_R and M_R is calculated in the reference frame (Eq. 1) using ridge regression. After that, the prediction result P_Q in the query frame is acquired by multiplying F_Q and W (Eq. 2). During training, the loss error between P_Q and M_Q is back-propagated to enhance the network' adaptation ability between the reference frame and the query frame. During inference, the reference frame (I_R and M_R) is always the first frame, and the query image I_Q is the rest sequence from the same video. Through iterative meta-learned, our network is capable of quickly adapting to unseen target object(s) with a few examples.

C. Fast Video Object Segmentation

A few previous methods proposed to tackle fast video object segmentation. In particular, FAVOS [6] first tracks the part-based detection. Then, based on the tracked box, it generates the part-based segments and merges those parts according to a similarity score to form the final segmentation results. The limitation of FAVOS is that it can not be learned in an end-to-end manner, and heavily relies on the part-based detection performance. OSNM [40] proposes a model which is composed of a modulator and a segmentation network. Through encoding the mask prior, the modular can help the segmentation network quickly adapt to the target object. RGMP [38] shares the same spirit with OSNM. Specifically, it employs a Siamese encoder-decoder structure to utilize the mask propagation, and further boosts the performance with synthetic data. The most similar work to ours is PML [5], which formulates the problem as a pixel-wise metric learning problem. Through the FCN [21], it maps the pixels to high-dimensional space, and utilizes a revised triplet loss to encourage pixels belonging to the same object much closer than those belonging to different objects. Nearest neighbor (NN) is required for retrieval during inference. In contrast our meta-learning approach acquires a mapping matrix between the high-dimensional feature and annotated mask in reference image using ridge regression, and then can be adapted rapidly to generate the prediction mask. Compared to baseline method PML [5], our method is twice faster and achieves 3.8 percent gains regarding to segmentation accuracy. And with the same efficiency, the J mean of our method is 3.4 percent better than OSNM [40] on the DAVIS2016 [28] validation set.

III. Methodology

A. Overview

We formulate the video object segmentation as a meta-learning problem. For each image pair which comes from a same video, ridge regression is used as the optimizer to learn the base learner. Meta learner is naturally built through the training process. Once the meta learner is learned, it possesses the ability of fast mapping between the image features and object masks, and can be adapted to unseen objects quickly with the help of the reference image.

According to the phase that user input involved in the training loop, the current existing methods can be classified into three categories.

User input outside the network training loop This category utilizes the user input to fine-tune the network to over-fit the appearance cues of target object(s) during inference. The representatives are OSVOS [3] and its following works [22], [1], [37]. Since online fine-tuning is required during inference, the limitation of these algorithms is time-consuming, which usually take seconds per image, thus is not practical for the real-world applications.

User input within the network training loop This category of work injects the user input as the additional input for training the network. Through this way, no online fine-tuning is needed. These algorithms incorporate the user input either by using a parallel network or concatenating the image with the user input [38], [40]. One limiation of this kind of methods is that the model needs to be recalculated once the user input changes, thus it is not practical for adaptation especially for long videos.

User input is detached from the network training loop In contrast to the previous methods, our algorithm shares the same spirit with PML [5] in design. The network and user input are detached, and the user input can be much more flexible. Moreover, once the user input is given (for example, the annotation in the reference image), the network can quickly adapt to the target objects without any extra operations.

B. Segmentation as Meta-Learning

For simplicity, we assume single-object segmentation case, and the annotation of first frame is given as the user input. Note that our method can also be applied for multi-objects and easily extended to other types of user input, e.g., scribble, clicks etc.

We adopt the following notation:

- C: the number of feature channels (in our case 800);
- w, h: the spatial resolution of the extracted features (in our case 1/8th of the orginal image size);
- F_R, F_Q: the feature tensors of size $C \times h \times w$ produced by ϕ;
- X: a flattened tensor of F_R or F_Q, with shape $h \cdot w \times C$;
- Y: the flattened tensor of annotation mask M_R or M_Q, with shape $h \cdot w \times 1$;
- W: the mapping matrix of size $C \times 1$ between the feature space and annotation mask.

As noted above, there are two components to the learner: (i) $\phi(.)$ an embedding model that maps images to a high-dimensional feature space, $C \times h \times w$; and (ii) an adaptor W of size $C \times 1$, found using ridge regression, that maps the embedded features to a (flattened) segmentation mask (of size $h \cdot w \times 1$).

Embedding Model We adopt DeeplabV2 [4] built on the ResNet-101 [13] backbone structure as our feature extractor ϕ. This choice allows a direct comparison of our method with the baseline, PML [5]. First, we use the pretrained model on COCO [20] dataset as the initialization for semantic segmentation. Then the ASPP [4] layer for classification is removed and replaced by our video-specific mapping W.

Ridge Regression Ridge regression is a closed form solver and widely-used in machine learning community [30], [25]. The learner seeks W that minimizes Λ as follows:

$$\begin{aligned} \Lambda(X, Y) &= \arg\min_{W} ||XW - Y||^2 + \lambda ||W||^2 \\ &= (X^T X + \lambda I)^{-1} X^T Y \end{aligned} \tag{1}$$

where, X, Y and W are as defined above, and λ is a regularization parameter, and set to 5.0 in all of our experiments. As can be seen in Fig. 3, during training, an image pair as well as their annotations are sampled from the same video sequence. The feature F_R extracted from the reference image I_R (in the Fig. 3 this is the first image) and its annotation M_R will be used to calculate the mapping matrix W.

$$P_Q = F_Q \times W \tag{2}$$

(where we abuse notation and use the unflattened feature tensors for clarity).

For the query image I_Q, likewise we compute the feature F_Q, map these to the predicted segmentation mask P_Q using Equation 2 in which W is the matrix computed from the reference image and its ground truth. The loss between the prediction mask P_Q and the annotation M_Q for the query provides the back-propagation signal to improve ϕ's ability to produce adaptable features.

During inference in our case, the reference frame I_R will be always the first frame, for which the annotation mask is provided, and the query frames I_Q will be the rest of frames in the same video.

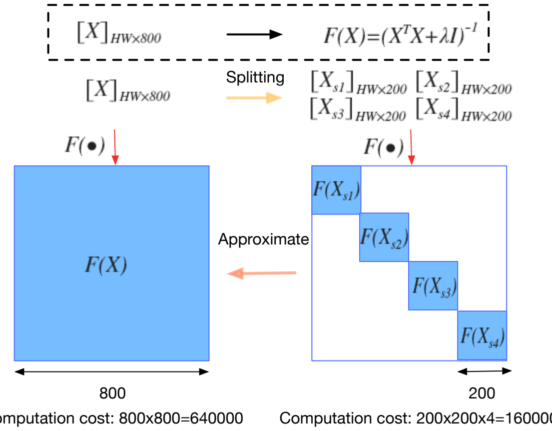

Fig. 4. Illustration of the proposed block splitting: during matrix inverse calculation of ridge regression, the computation of the higher dimensional feature is approximated by the sum of computation of that lower dimensional features. Which can effectively speed up the training process as well as reducing the parameters and memory.

C. Block Splitting

Thanks to ridge regression, the computation of the mapping matrix and gradient back-propagation are already very fast compared with other algorithms, which also focus on video object segmentation.

$$F(X) = (X^T X + \lambda I)^{-1} \tag{3}$$

During the experiments, we found the higher dimension of the feature used as the input for meta-learning module, the more accurate segmentation results likely be achieved. However, we also observed that the higher dimension of the feature being utilized, the slower of the training process. Specifically, during the computation of mapping matrix W, it involves a matrix inverse calculation. as denoted by Equation 3, which will become the bottleneck of fast propagation when the very high dimensional feature is used.

In order to further speed up the training process of the proposed network, we deliver a block splitting mechanism, and its work principle as shown in Fig. 4. In particular, our motivation is that the matrix inverse computation $F(X)$ for much high-dimensional feature (eg. 800D) can be approximated by the sum of the computations of that relative low-dimensional features (eg. 200D $\times$ 4). From the work principle, it can be viewed that a $n \times n$ matrix can be approximated by four $n/4 \times n/4$ irrelevant diagonal matrix.

The advantages of using the proposed block splitting mechanism are: Firstly, it can largely speed up the matrix inverse process involved in ridge regression, thus it saves the training time to some extent. Secondly, through the matrix approximation step as aforementioned, the network parameters involved in the ridge regression as well as memory utilized in our network are reduced. The experimental evidence can be found in Ablation Study (Section V).

D. Training

Training Strategy For training, optimizer is SGD with momentum 0.9, with weight decay 5e-4. We use the DeepLabV2 [4] with backbone network ResNet-101 [13] as the feature extractor, and the constant learning rate, i.e. 1.0e-5, is used during the whole training process. The dimension of extracted feature is 800 outputed by the feature extractor, which is used as the input for the meta-learning module.

Loss BCEWithLogitsLoss[2] is employed for training the proposed network, it essentially is a combination of the Sigmoid layer and binary cross entropy (BCE) loss, it benefits from the log-sum-exp trick for numerical stability. And compared to BCE loss, it is more robust and less likely to cause numerical problem when computing the inverse matrix in the ridge regression step.

$$\begin{gathered}\ell(x,y) = L = \{l_1, ..., l_N\}^T \\ l_n = -w_n[y_n \cdot \log \delta(x_n) + (1-y_n) \cdot \log(1-\delta(x_n))]\end{gathered} \tag{4}$$

where N is the batch size. x_n is the input of the loss calculation, and y_n ($y_n \in [0,1]$) is the ground truth label. w_n is a rescaling weight given to the loss of each batch element.

IV. Experiments

A. Dataset

We verify the proposed method both on DAVIS2016 [28] and SegTrack v2 [19] datasets.

On DAVIS2016, which contains 50 pixel-level annotated video sequences, and each video only contains one target object for segmenting. Among these 50 video sequences, 30 video sequences as the training set with which the annotated mask is provided for every frame. And another 20 video sequences as the validation set, and only the annotation of the first frame is allowed to access.

SegTrack v2 [19] is extended from SegTrack [34] dataset. Both of them contain the dense pixel-level annotation for each frame within each video. For segtrack v2 dataset, we test our algorithm on all the sequences which contain one target object.

[2]https://pytorch.org/docs/stable/nn.html

TABLE I
Performance comparison of our approach with recent approaches on DAVIS 2016 Performance measured in mean IoU. PML* denotes PML without spatial-temporal and online adaptation which is the same case with our method.

Method	DAVIS	Online Tuning	OptFlw	CRF	BS	Speed(s)
OFL	68.0	-	✗	✓	✗	42.2
BVS	60.0	-	✗	✗	✗	0.37
ConvGRU	70.1	✗	✓	✗	✗	20
VPN	70.2	✗	✗	✗	✗	0.63
MaskTrack-B	63.2	-	✗	✗	✗	0.24
SFL-B	67.4	✗	✓	✗	✗	0.30
OSVOS-B	52.5	✗	✗	✗	✗	0.14
OSNM	72.2	✗	✗	✗	✗	0.14
PML*	72.0	✗	✗	✗	✗	0.28
Ours	75.8	✗	✗	✗	✗	0.145
PLM	70.0	✓	✗	✗	✗	0.50
SFL	74.8	✓	✗	✗	✗	7.9
MaskTrack	69.8	✓	✗	✗	✗	12
OSVOS	79.8	✓	✓	✗	✓	10

B. Results on DAVIS2016

Quantitative Results Table I shows the experimental results on DAVIS2016 [28] on different methods. Apart from the performance (measured by J mean), switches for online-fining, using optical-flow, dense CRF (CRF) and boundary snapping (BS) are also described. Meanwhile, the inference time is also shown. In particular, compared with most of the competitors, our algorithm shares the same or much faster processing time with superior performance regarding the segmentation accuracy. Compared with OSVOS [3], for which the online fine-tuning is necessary, our method just takes a smaller fraction of time to do inference. Compared to the baseline method PML [5] which use the same feature extractor, our method is twice faster and achieve 3.8 percent gains with the same settings. Compared OSNM [40], with the same efficiency, our method achieve 3.4 percent improvements regarding to the segmentation accuracy.

Qualitative Results Fig. 5 demonstrates some visualized results of our method. As shown in Fig. 5, our method is not only good at recovering object details (e.g., the results on the sequence of blackswan), but also robust against heavy occlusions (eg. the results on the sequences bmx-bumps and libby, dramatic movement as well as abrupt rotation (eg. the results on the sequence motocross-bumps). However, there are very few scenarios which may lead to failure cases (denoted by the red box), and mainly caused by the (noisy) objects which have not appeared at the first frame of the video, and can be easily cured by some post-processing steps, including tracking [6], online adaptation [37], [5].

Fig. 5. Qualitative results: Homogeneous sample of DAVIS sequences with our result overlaid.

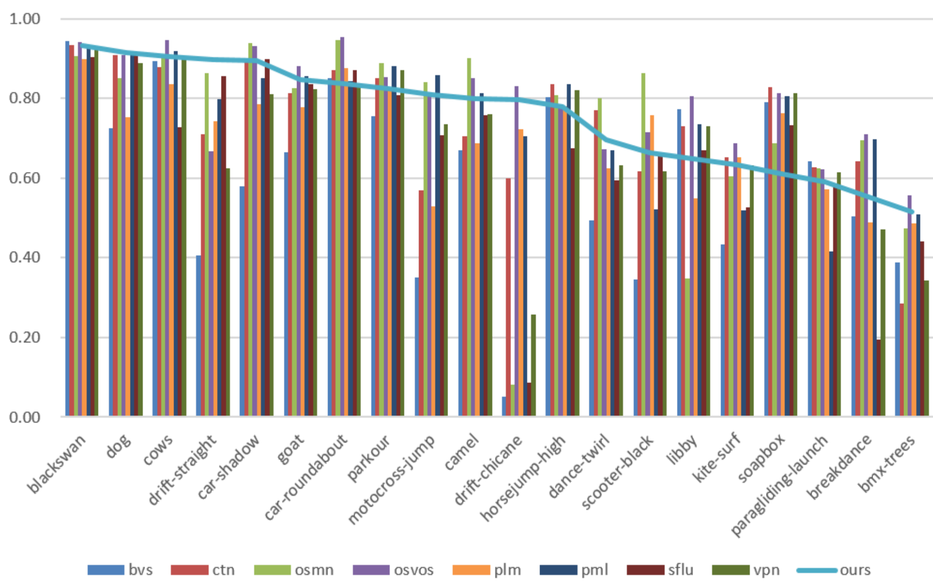

Fig. 6. Per-sequence results of mean region similarity (J) . Sequences are sorted by the performance of our algorithm.

In Fig. 7, we show some visualized results compared with OSVOS [3] and PML [5]. For the breakdance, scooter-black and dance-jump sequences, which contain fast moving and abrupt rotation, OSVOS [3] performs worse than PML [5]. And for the dog sequence, PML [5] can not achieve a satisfied result due to the dramatic change of the light conditions. However, on both of these two scenarios, the proposed method performs better than both of OSVOS and PML, which is benefit from robust adaptation ability of our network.

C. Results on SegTrack Dataset

In Fig. 8, some visualized results in the segTrack [34] dataset are shown. Which are acquired by direcly utlized the model trained on DAVIS2016 dataset. As can be seen, in most cases, our model maintain a good segmentation accuracy, and with a few case fails (as denoted by the red box), which mainly due to the dramatically changes of the light conditions and exact same appearance between the background and the target object. These results prove our method has a better generalization ability and can be quickly adapted to other unseen objects with very few examples (here, only the annotation in the first frame is provided).

V. Ablation Study

A. Comparison with PML with different adds-on

TABLE II
Comparison with basedline method PML [5] under different settings.

Method	Spat.-Temp.	Online Adapt.	MaskIoU
PML-Abal1	✗	✗	72.0
PML-Abal2	✗	✓	73.2
PML-Abal3	✓	✗	74.3
PML	✓	✓	75.5
Ours	✗	✗	75.8

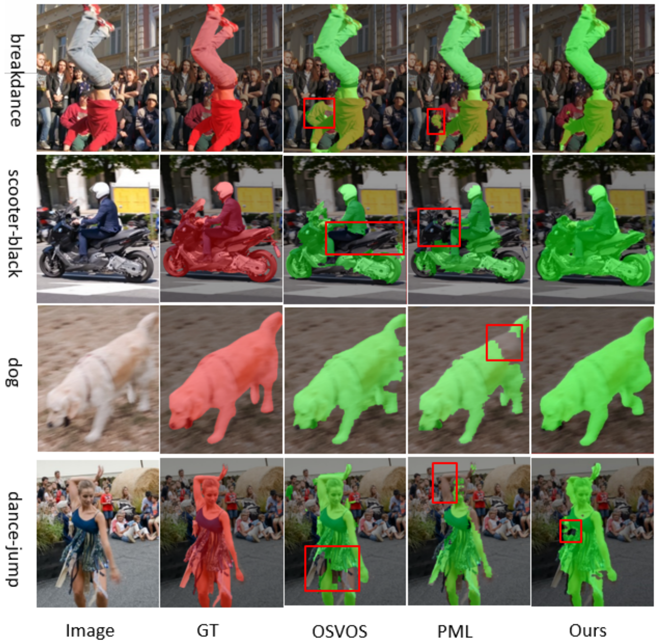

Fig. 7. Visualized comparison between the proposed method and other methods. With the red box to denote the error region.

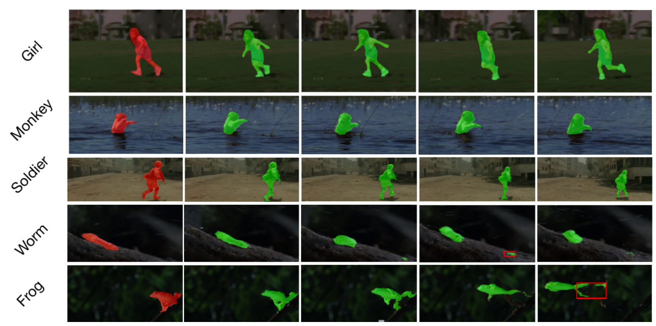

Fig. 8. Qualitative results: Homogeneous sample of SegTrack sequences with our result overlaid.

Compared to baseline method PML [5] which with same backbone network (DeepLabV2) under the same settings, our method achieved 3.8 percent improvement regarding to MaskIoU accuracy as shown in Table II. And compared with baseline method adding spatial temporal attention and online adaptation, our method is still slightly better and twice faster.

B. Feature Dimension and Block Splitting

As mention in Section III-C, since our meta learning module (ridge regression) requires the computation of matrix inverse, the training speed will varies significantly regrading the features with various dimensions utilized for this step. And based on the fact that low dimensional features usually have the faster speed but lose some details of image information. On the contrary, high dimensional features are time-consuming but carry much rich information. We propose a block splitting mechanism to train the meta learner. In Table III, the splitting number (of feature), feature dimension, running speed (per iteration), memory cost (of the whole network), as well as computation cost (of the computation of matrix inverse) with different settings are listed out. As can be seen, with the feature dimension decreasing, the overall trend are running speed increasing, computation cost decreasing, dramatically. However the memory cost reduce slightly, which mainly because of the backbone feature extractor take up most of the memory usage. All the numbers are tested on the single GPU card (with type of GTX 1080). Please note, the performance of using different splits change slightly during the preliminary experiments. The reason for using feature with 800D is based on the observation that: The higher dimension of the feature, the stronger representation ability and the slower training speed. 800D feature is somehow a compromise between the good performance and fast training speed.

TABLE III

Ablation study on block splitting: feature dimension, running speed, memory and computation cost with different settings are listed out.

Split No	Feature	Speed(s)	Memory	Computation
1	800	1.50	11590	640k
2	400	1.23	11720	320k
4	200	0.75	11580	160k
8	100	0.86	11584	80k

C. Per Sequence Performance Analysis

In Fig. 6, J mean of per sequence of different methods are outlined. It is sorted according our algorithm's performance in each sub-sequence, which provides a more intuitive understanding for the proposed algorithm. Firstly, the proposed method achieve a better video segmentation accuracy when compared to many other methods. Secondly, our algorithm works quite well on most of sequences, even on the most challenging sequences, e.g., breakdance and bmx-tree, the J mean is above 0.5. Thirdly, benefit from the quick adaption ability of meta-learning, around half of sequence achieve J mean over 0.8. Moreover, our method can well recover the object details as well as robust against fast movement and heavy occlusion, which are aligned with our conclusion in Section IV-B

VI. Conclusion

In this paper, we explore applying meta-learning into video object segmentation system. A closed form optimizer, i.e., ridge regression, is utilized to update the meta learner, which achieves fast speed while maintains the superior accuracy. Through iteratively meta-learned, the network is capable of conducting fast mapping on unseen objects with a few examples available. Compared to the fine-tuning methods, our algorithm with similar performance but just a smaller fraction time is required, which is appeal to the real-world applications. In addition, a block splitting mechanism is delivered to speed up the training process, which also has the benefits of reducing parameters and saving memory. In future work, we would like to use other basic optimizers, such as, Newton's methods and logistic regression. Meanwhile, based on the flexible design of our meta-learner, instead of inferring the rest frames from the given whole annotation of the first frame. Inferring whole object from only part of annotation or user feedback is also worth to investigate.

References

[1] L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5977–5986, 2018.

[2] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. arXiv preprint arXiv:1805.08136, 2018.

[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. IEEE, 2017.

[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. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2018.

[5] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1189–1198, 2018.

[6] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. arXiv preprint arXiv:1806.02323, 2018.

[7] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In Computer Vision (ICCV), 2017 IEEE International Conference on, pages 686–695. IEEE, 2017.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence, 28(4):594–611, 2006.

[10] V. Garcia and J. Bruna. Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043, 2017.

[11] J. P. Gee. Deep learning properties of good digital games: how far can they go? In Serious Games, pages 89–104. Routledge, 2009.

[12] R. Girshick. Fast r-cnn. In The IEEE International Conference on Computer Vision (ICCV), December 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[14] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In Advances in neural information processing systems, pages 109–117, 2011.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[16] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[18] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. The International Journal of Robotics Research, 34(4-5):705–724, 2015.

[19] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In Proceedings of the IEEE International Conference on Computer Vision, pages 2192–2199, 2013.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[22] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. arXiv preprint arXiv:1709.06031, 2017.

[23] R. H. Myers and R. H. Myers. Classical and modern regression with applications, volume 2. Duxbury Press Belmont, CA, 1990.

[24] D. K. Naik and R. Mammone. Meta-neural networks that learn by learning. In Neural Networks, 1992. IJCNN., International Joint Conference on, volume 1, pages 437–442. IEEE, 1992.

[25] I. Nouretdinov, T. Melluish, and V. Vovk. Ridge regression confidence machine. In ICML, pages 385–392, 2001.

[26] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In BMVC, volume 1, page 6, 2015.

[27] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In Computer Vision and Pattern Recognition, volume 2, 2017.

[28] 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 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[30] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. 1998.

[31] J. Schmidhuber. Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook. PhD thesis, Technische Universität München, 1987.

[32] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems, pages 4077–4087, 2017.

[33] J. Sun, D. Yu, Y. Li, and C. Wang. Mask propagation network for video object segmentation. arXiv preprint arXiv:1810.10289, 2018.

[34] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. International journal of computer vision, 100(2):190–202, 2012.

[35] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[36] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In Advances in Neural Information Processing Systems, pages 3630–3638, 2016.

[37] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. arXiv preprint arXiv:1706.09364, 2017.

[38] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7376–7385, 2018.

[39] Z. Xu, C. Hu, and L. Mei. Video structured description technology based intelligence analysis of surveillance videos for public security applications. Multimedia Tools and Applications, 75(19):12155–12172, 2016.

[40] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. algorithms, 29:15, 2018.