In defense of OSVOS

Yu Liu *
yu.liu@robotvision.org
Yutong Dai *
yutong.dai@adelaide.edu.au
Anh-Dzung Doan
dung.doan@adelaide.edu.au
Lingqiao Liu
lingqiao.liu@adelaide.edu.au
Ian Reid
ian.reid@adelaide.edu.au

Abstract

As a milestone for video object segmentation, one-shot video object segmentation (OSVOS) has achieved a large margin compared to the conventional optical-flow based methods regarding to the segmentation accuracy. Its excellent performance mainly benefit from the three-step training mechanism, that are: (1) acquiring object features on the base dataset (i.e. ImageNet), (2) training the parent network on the training set of the target dataset (i.e. DAVIS-2016) to be capable of differentiating the object of interest from the background. (3) online fine-tuning the interested object on the first frame of the target test set to overfit its appearance, then the model can be utilized to segment the same object in the rest frames of that video. In this paper, we argue that for the step (2), OSVOS has the limitation to ‘overemphasize’ the generic semantic object information while ‘dilute’ the instance cues of the object(s), which largely block the whole training process. Through adding a common module, video loss, which we formulate with various forms of constraints (including weighted BCE loss, high-dimensional triplet loss, as well as a novel mixed instance-aware video loss), to train the parent network in the step (2), the network is then better prepared for the step (3), i.e. online fine-tuning on the target instance. Through extensive experiments using different network structures as the backbone, we show that the proposed video loss module can improve the segmentation performance significantly, compared to that of OSVOS. Meanwhile, since video loss is a common module, it can be generalized to other fine-tuning based methods and similar vision tasks such as depth estimation and saliency detection.

1 Introduction

With the popularity of all kinds of mobile device and sensors, countless video clips are uploaded and shared through the social media platforms and video websites every day. Smartly analysing these video clips are very useful yet quite challenging. The revival of deep learning boosts the performance of many recognition tasks on static images to a level that can be
matched with human beings, including object classification [10, 13, 19], semantic segmentation [4, 12, 14] and object tracking [3, 22]. Compared to static images, video clips contain much more rich information, and the temporal correlations among inter-frame, if being used appropriately, it can significantly improve the performance of the corresponding tasks on static images. As one of the most active fields in computer vision community, video object segmentation aims to distinguish the foreground objects(s) from the background in pixel level. In 2017, one-shot video object segmentation (OSVOS) is proposed by Caelles et al [2], as a milestone in this research field, which achieves over 10% improvements compared to the previous conventional methods regarding the segmentation accuracy.

Motivation and principle of OSVOS The design of OSVOS is inspired by the perception process of human beings. Specifically, when we recognize an object, the first thing come into our view are the image features such as corners and textures of the scene, then we can distinguish the object(s) from background through shape and edge cues, which also named objectness. Finally, based on the rough localization from the above two steps, we will pay attention on the details of the target instance.

In particular, OSVOS utilizes a fully-convolutional neural network (FCN) [14] to conduct video object segmentation, and the three phases are:

- **Acquire object features**: to acquire the generic semantic features from ImageNet.
- **Train parent network**: to train a network on DAVIS-2016 training set, which is capable of distinguishing the foreground object(s) from the background.
- **Online fine-tuning**: based on the parent network, to train the network which is over-fitting the appearance of the target instance on the first frame.

The Pros and Cons of OSVOS
- **Pros:** The online fine-tuning process of OSVOS wishes to fully acquire the appearance of the target object in the first frame. Hence, it is capable of handling the fast moving, abrupt rotation as well as heavy occlusion, which are the limitations of the conventional optical flow based methods.

- **Cons:** i) when similar (noisy) objects appear in the subsequent frames of the video sequence, they will be wrongly segmented as foreground objects. ii) when the appearance of the target object changes dramatically in the later phase of the video sequence, the algorithm fails to segment the new appearance parts.

**The motivation of video loss**  We propose the video loss in defense of OSVOS based on two observations:

- For CNN, the low-level layers have relatively large spatial resolutions, and carry more details about the object instance, while the high-level layers have more stronger abstract and generalization ability, leading to carry more category information. Especially, in the second phase of OSVOS, i.e. training the parent network, it actually tries to fine-tune the network to acquire the ability of distinguishing the objects from the background. However, it dilutes the ‘instance’ information. And quickly adapts to the specific (target) instance, which is exactly the need of third phase (i.e. online fine-tuning). Video loss can effectively ‘rectify’ the training process of parent network, and make it be better prepared for the online fine-tuning.

- Each video is supposed to maintain an average object, and through mapping, we expect that the objects from a same video are close to each other in the embedding space, while the objects from different videos are far away from each other. By this way, video loss can help the network to maintain an average object for each video sequence.

## 2 Related Works

For the task of semi-supervised video object segmentation, the annotations of first frame is given, and the algorithm is expected to infer the object(s) of interest in the rest frames of the video. According to design principle, the existing algorithms which achieve the state-of-the-art performance on DAVIS benchmark [17] for semi-supervised video object segmentation can be roughly classified into three categories:

### 2.1 Tracking based Methods

In this category, one stream of methods employ the optical flow to track the mask from the previous frame to the current frame, including MSK [18], MPNVOS [20] etc, one limitation of those methods is that they can not handle heavy occlusion and fast moving. Most recently, there are an emergence of methods which use the ReID technique to conduct the video object segmentation, including PReMVOS [21] and FAVOS [3]. Specifically, FAVOS using ReID to tackle the part-based detection box first, and through merging the (box) region based segments to form the final segmentation. PReMVOS firstly generate instance segmentation proposals in each frame, and then take use of the ReID technique to do data association to pick the correct segments in temporal domain, which can largely reduce the background noises brought by other nearby or overlapped object(s).
2.2 Adaptation based Methods

For this category of methods, the core idea is utilizing the mask priors acquired from the previous frame(s) to be the guidance, to supervise the prediction in the current frame. Specifically, Segflow \cite{segflow} takes use of a parallel two-branch network to predict the segmentation as well as optical flow, through the bidirectional propagation between two frames, calculating optical flow and segmentation together and expecting them to benefit from each other. RGMP \cite{rgmp} takes both annotations of the first frame and predicted mask of the previous frame as guidance, employs a Siamese encoder-decoder to conduct the mask propagation as well as detection, and with synthetic data to further boost the segmentation performance. OSMN \cite{osmn} shares the similar design principle with RGMP, while the difference is that it uses an modulator to quickly adapt the first annotation to the previous frame, which can then be used by the segmentation network as the spatial prior.

2.3 Fine-tuning based Methods

Besides the aforementioned two categories of methods, there are some fine-tuning based methods which achieve the top performance in video object segmentation benchmark are OSVOS-S \cite{osvos-s}, OnVOS \cite{onvos}, CINM \cite{cinm} etc, and all of them are derived from OSVOS \cite{osvos}. Specifically, OSVOS-S \cite{osvos-s} aims to solve the problem of removing noisy object(s) with the help of instance segmentation. While OnAVOS \cite{onavos} tries to enhance the network’s ability for recognizing the new appearance of the target object(s) as well as suppressing the similar appearance carried by the noisy object(s). CINM \cite{cinm} is also initialized with the fine-tuning model, and employ a CNN to infer the markov random field (MRF) in spatial domain, and with optical flow to track the segmented object(s) in temporal domain.

2.4 Video Loss

In this paper, targeting to improve the fine-tuning methods, with OSVOS as an entry, we deliver a tiny head video loss. As aforementioned, our observation is based on the ‘delayed’ learning process for target instance(s) between the parent network and online fine-tuning. Through incorporating a basic component, i.e. video loss, we achieve better performance regarding to segmentation accuracy compared to OSVOS using exactly same backbone network structure(s). Furthermore, considering that it may not always be easy to distinguish the object(s) from background in 2D image coordinate, we further utilize metric learning and the proposed mixed instance-aware video loss to enforce the pixels, after mapping through a FCN in high-dimensional space, which belong to target object(s) or background are supposed to be closed with each other, while any two pixels with one belong to the target object(s) and the other belong to background are supposed to has a relatively far distance with each other. Through employing the proposed video losses, the performance has been significantly improved regarding to the segmentation accuracy, and some noisy objects have been effectively removed. Moreover, since video loss is a common building block, it can be generalized to all kinds of fine-tuning based methods including, but not limit to OnVOS \cite{onvos}, OSVOS-S \cite{osvos-s}, CINM \cite{cinm} etc.

3 Methodology

The motivation, design and key implementation of video loss will be illustrated in detail.
Figure 2: The workflow of OSVOS-VL. Compared to OSVOS, only a tiny head, video loss block, is added.

3.1 Overview

The assumption for video loss is that, different objects are linear separable in high-dimentional space, i.e. the feature space. Meanwhile, the euclidean distances of the features of the same object are supposed to be smaller than that of different objects. The workflow of OSVOS-VL is shown in Figure 2. As can be seen, video loss just like a light-weight head being parallel with the prediction part, thus the extra time cost is insignificant. Once the better features are obtained after training of parent network, it would ease the learning processing of online fine-tuning stage and is much prone to achieve accurate segmentation results compared to that of OSVOS. This design can be viewed as maintaining an average target object for each video, and expecting the objects from different videos are much more far away than that of from the same video, which effectively prevents the background noise from other objects (of no interest). We deliver three types of video loss (VL) in this paper. The first one is the two dimensional video loss (2D-VL), which make the parent network to push away different objects in image coordinates. The second and third are the high-dimensional video loss (HD-VL). Established on 2D-VL, the HD-VL further maps 2D features to high dimensional space, and clusters pixels which belong to the same instance together, and utilize object centers in HD-space as constraints.

3.2 Two Dimensional Video Loss

In OSVOS [2], considering the sample imbalance between the target object(s) and the background, weight cross entropy loss is employed to conduct the pixel-wise segmentation task. The expression of weighted cross entropy loss as follows:

\[
L_{2d} = -\frac{Y-}{Y} \sum_{j \in Y_+} \log P(y_j = 1 | X) - \left( 1 - \frac{Y_-}{Y} \right) \sum_{j \in Y_-} \log P(y_j = 0 | X)
\]  

(1)

Where X is the input image, \( y_j \in 0, 1, j = 1, \ldots, |X| \) is the pixelwise binary label of X, and Y and Y_- are the positive and negative labeled pixels. P(·) is obtained by applying a sigmoid to the activation of the final layer. \(|Y_-| / |Y|\) is employed for the purpose of training the imbalanced binary task as in [23].

OSVOS [2] only rely on weighted cross entropy loss to fine-tune parent network, but we argue that it will mix up all of the objectness features in DAVIS dataset, without classifying...
which kind of objects the foreground belongs to. It may make the online fine-tuning process more harder to recognize which instance the object is. Therefore, we propose 2D-VL to force the network to learn features of different instances during the training of parent network.

For this purpose, we add the identity of each video \(v_{id}\) into the training process as input. After recognizing the \(v_{id}\) of the training (image) data, the network can update each specific (video) category through back propagation. Please note, 2D-VL share the same expression with Equation 1, but different from the prediction branch, our 2D-VL only updates corresponding (video) category directly, as illustrated in Figure 3.

### 3.3 High Dimensional Video Loss

With the observation that objects in 2D dimension usually have similar appearances or shapes, which brings too much confusion to the network to distinguish an object accurately, we propose to map the prediction to a high-dimensional space firstly, and expecting that after mapping, the distance among different objects is enlarged. The mapping process in high dimensional space is shown in Fig. 4. In HD space, we look forward to clustering embeddings from different objects into different groups.

In PML [5], a modified triplet loss is utlized to pull samples with same identity close to each other, and only constrain the smallest negative points and smallest positive points. Inspired but different from that, we randomly sample 256 points in both foreground parts and background parts, pulling points from the same part together and push points from different parts away. The triplet loss is defined as

\[
L_{hd, tl} = y \| f(x_1) - f(x_2) \| + (1 - y) \max(0, \lambda - \| f(x_1) - f(x_2) \|) \tag{2}
\]

where \(y = 1\) when point \(x_1, x_2\) belong to a same cluster (foreground or background) and \(y = 0\) when point \(x_1, x_2\) belong to different clusters.

### 3.4 Mixed Instance-aware Video Loss

**Contrastive Center Loss** Inspired by the work [6], we define a contrastive center loss for the purpose of pulling embeddings with the same label close to each other while pushing embeddings with different labels far away from each other. To restrict the entire foreground

![Figure 3](https://via.placeholder.com/150)

**Figure 3:** The illustration of two-dimensional video loss.
area and background area, we first calculate the center point of each part. Then we use the contrastive center loss function to penalize the distance between these two center points. The motivation behind this is to restrict distribution of foreground embeddings and reduce the amount of computation.

\[
L_{hd_{cl}} = \max(0, \lambda - \|\mu_+ - \mu_-\|)
\]  

(3)

where \(\mu_+\) represents the center point of foreground cluster, and \(\mu_-\) represents the center point of background cluster, both in high-dimensional space.

**Mixed Loss** Contrastive center loss is a loss restricting the overall distribution of examples, while triplet loss considers the constraints in pixel level. In order to combine two types of constraints, here we define a mixed loss as

\[
L_{hd_{mix}} = \beta_1 L_{hd_{cl}} + \beta_2 L_{hd_{tl}}
\]  

(4)

Where \(\beta_1\) and \(\beta_2\) are the coefficients for balancing two loss terms.

### 3.5 Training

In order to form a fair comparison with OSVOS [2], we adopt the same settings for the training of parent network and online fine-tuning except the training epochs. Specifically, SGD solver with momentum 0.9 is used, learning rate is 1e-8, the weight decay is 5e-4. Batch size is 1 for VGG16 based experiments and is 2 for MobileNet [11] based experiments, respectively. For training the parent network, fine-tuning of 240 epochs is conducted based on the initialization of ImageNet [9] features. For online fine-tuning, 10k iterations of fine-tuning is applied for all of the experiments for the fair comparison.
4 Experimental Results

4.1 Dataset

DAVIS-2016 \[17\] is the most widely used dataset for video object segmentation, which is composed of 50 videos with pixel-wise annotations for single-object. Among them, 30 video sequences are chosen as training set, and the other 20 video sequences are utilized as test set.

4.2 Quantitative Results

In Table 1, J Mean for both parent network and online fine-tuning with different structures as backbones are listed out. As can be seen, for both of the experiments which based on VGG16 and MobileNet, OSVOS+ video loss achieve the better performance during online fine-tuning phase, while with comparable performance with OSVOS during parent network training phase, which proves our assumption that video loss, as a common module, is effective in helping the (FCN) network to recognize the target instance. In Table 2, compared to OSVOS with video loss utilized in 2D (OSVOS-V2d), OSVOS with high-dimensional loss (OSVOS-Vhd, OSVOS-Vmixed) performs better, which is matched with our observation that sometimes it is much more easier for similar features to be distinguished in high-dimensional space than that of in two-dimensional space. Please note, all of our experiments trained with 10k iterations and without any post-processing for the purpose of fair comparison and saving training time, which is slightly different from \[2\], and the preliminary experiments we tested show that as the training iterations increasing (around 20k iterations), which can replicate the numbers that the paper \[2\] report.

4.3 Qualitative Results

We also provide some visualized comparisons in Figure 5 between OSVOS and OSVOS-V2d, as can be seen, among the results acquired by OSVOS, wrong segments are accompanied in the surroundings of the target object, we suspect that is because only rely on prediction loss in OSVOS can not distinguish instance information between the target object

| Method        | Parent Network | Finetuning | Backbone |
|---------------|----------------|------------|----------|
| OSVOS         | 52.5           | 75.0       | VGG16    |
| OSVOS-V2d     | 50.8           | 76.2       | VGG16    |
| OSVOS         | 53.1           | 65.7       | MobileNet|
| OSVOS-V2d     | 54.1           | 66.2       | MobileNet|

Table 1: J Mean of OSVOS and OSVOS-V2d with different backbone network structures.

| Method     | Parent Network | Finetuning |
|------------|----------------|------------|
| OSVOS      | 53.1           | 65.7       |
| OSVOS-V2d  | 54.1           | 66.2       |
| OSVOS-Vhd  | 53.7           | 66.9       |
| OSVOS-Vmixed | 58.6       | 67.5       |

Table 2: J Mean of OSVOS, OSVOS-V2d, OSVOS-Vhd, OSVOS-Vmixed. With MobileNet as backbone, and 20 is dimensions for the embedding of OSVOS-Vhd and OSVOS-Vmixed.
and background (noisy) objects. In contrast, OSVOS-VL effectively remove the noisy parts compared to that of OSVOS.

4.4 Performance on per sequence

In order to have a better understanding of the work principle of the proposed video loss block, we illustrate performance comparison of OSVOS and OSVOS-V2d on per sequence, as shown in Table 3, for both VGG16 and MobileNet based experiments, in 12 out of 20 sequences, OSVOS-V2d achieve better performance than OSVOS, in some sequences such as bmx-trees, camel, dance-twirl, dog, drift-chicane, drift-straight, motocross-jump, paragliding-launch. OSVOS-V2d achieves consistent improvements on both backbones, and these sequences usually contain abrupt motions or noisy objects which share the similar appearance with the target object.

5 Conclusion

In this paper, we deliver a common module, video loss, for video object segmentation, which is tailored to overcome the limitation of fine-tuning methods, during the phase of training parent network, dilute the instance information, hence delay the overall training process. Considering in CNN, the shallow layers usually contain much rich details of object(s) which are the key cues to specify different instances, while the deeper layers have more stronger generalization ability to recognize generic objects. Various video losses are proposed as the constraints to supervise the training process of parent network, which is effective in removing the noisy objects. Once the training process is finished, the parent network is well prepared to adapt to the instance quickly during online fine-tuning. One of our future interests will be extending the video loss into other fine-tuning methods such OSVOS-S, OnVOS. Another one will be with the help of the network search technique to automatically decide the training epochs and learning rate.
Table 3: J Mean of OSVOS and OSVOS-V2d on per sequence. VGG denotes VGG16 and MN denotes MobileNet as the feature extractor.

| Sequence       | VGG+OSVOS | VGG+OSVOS-V2d | MN+OSVOS | MN+OSVOS-V2d |
|----------------|-----------|---------------|----------|--------------|
| Blackswan      | 94.1      | 93.6          | 93.3     | 94.0         |
| bmx-trees      | 52.8      | 58.5          | 42.0     | 42.8         |
| breakdance     | 67.6      | 67.7          | 75.1     | 70.0         |
| camel          | 83.7      | 85.8          | 70.2     | 75.1         |
| car-roundabout | 88.3      | 75.6          | 83.0     | 74.2         |
| car-shadow     | 88.6      | 83.5          | 74.3     | 74.2         |
| cows           | 95.2      | 94.9          | 90.7     | 87.9         |
| dance-twirl    | 60.7      | 64.9          | 63.4     | 66.2         |
| dog            | 72.6      | 88.3          | 88.7     | 90.8         |
| drift-chicane  | 61.3      | 73.9          | 26.9     | 36.0         |
| drift-straight | 56.4      | 61.8          | 35.5     | 35.7         |
| goat           | 88.1      | 87.9          | 82.0     | 85.5         |
| horsejump-high | 84.5      | 81.5          | 69.8     | 69.1         |
| kite-surf      | 75.3      | 73.9          | 54.9     | 55.3         |
| libby          | 75.4      | 77.2          | 69.4     | 68.3         |
| motocross-jump | 60.2      | 67.3          | 49.6     | 52.7         |
| paragliding-launch | 63.9 | 64.0          | 56.3     | 58.5         |
| parkour        | 89.0      | 89.2          | 81.5     | 73.6         |
| scooter-black  | 58.2      | 35.4          | 57.6     | 62.3         |
| soapbox        | 84.3      | 86.1          | 49.5     | 46.3         |

References

[1] Linchao Bao, Baoyuan Wu, and Wei 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] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 221–230, 2017.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008, 2018.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan 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] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc 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] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In Proceedings of the IEEE international conference on computer vision, pages 686–695, 2017.

[7] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In Pro-
ceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7415–7424, 2018.

[8] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. arXiv preprint arXiv:1708.02551, 2017.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[10] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[12] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8759–8768, 2018.

[13] Yu Liu, Lingqiao Liu, Hamid Rezatofighi, Thanh-Toan Do, Qinfeng Shi, and Ian Reid. Learning pairwise relationship for multi-object detection in crowded scenes. arXiv preprint arXiv:1901.03796, 2019.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[15] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. arXiv preprint arXiv:1807.09190, 2018.

[16] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. arXiv preprint arXiv:1709.06031, 2017.

[17] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander 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.

[18] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2663–2672, 2017.

[19] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[20] Jia Sun, Dongdong Yu, Yinghong Li, and Changhu Wang. Mask propagation network for video object segmentation. arXiv preprint arXiv:1810.10289, 2018.
[21] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017.

[22] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. *arXiv preprint arXiv:1812.05050*, 2018.

[23] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon 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.

[24] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[25] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018.