2018-12-01

Flow Adaptive Video Object Segmentation

Fanqing Lin
Brigham Young University

Flow Adaptive Video Object Segmentation

Fanqing Lin

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Tony Martinez, Chair
Bryan Morse
Quinn Snell

Department of Computer Science

Brigham Young University

ABSTRACT

Flow Adaptive Video Object Segmentation

Fanqing Lin
Department of Computer Science, BYU
Master of Science

We tackle the task of semi-supervised video object segmentation, i.e, pixel-level object classification of the images in video sequences using very limited ground truth training data of its corresponding video. Recently introduced online adaptation of convolutional neural networks for video object segmentation (OnAVOS) has achieved good results by pretraining the network, fine-tuning on the first frame and training the network at test time using its approximate prediction as newly obtained ground truth. We propose Flow Adaptive Video Object Segmentation (FAVOS) that refines the generated adaptive ground truth for online updates and utilizes temporal consistency between video frames with the help of optical flow. We validate our approach on the DAVIS Challenge and achieve rank 1 results on the DAVIS 2016 Challenge (single-object segmentation) and competitive scores on both DAVIS 2018 Semi-supervised Challenge and Interactive Challenge (multi-object segmentation). While most models tend to have increasing complexity for the challenging task of video object segmentation, FAVOS provides a simple and efficient pipeline that produces accurate predictions.

Keywords: machine learning, video object segmentation, semi-supervised, interactive, convolutional neural network, optical flow, online adaptation

ACKNOWLEDGMENTS

I want to thank the Lord for everything I have. I want to thank my family for their support, giving me the chance to pursue my dreams. I want to thank BYU for the great education that I am receiving. I want to also thank my advisors, Dr. Martinez and Dr. Morse for their mentoring and guidance. Last but not least, I want to thank my lovely wife Nan, for her faith in me and her continous support.

Table of Contents

Chapter 1

Introduction

As Convolutional Neural Networks (CNNs) revolutionize the field of computer vision, there has been a trend to move from tasks such as image classification [15, 23, 43, 53] to object detection [11, 12, 26, 38], and from image segmentation [3, 4, 21, 29] to video object segmentation [18, 25, 41, 48]. Video object segmentation and tracking is vital in computer vision and has many significant applications such as video editing, autonomous vehicles, robotics etc. The segmentation task for video objects involves classifying each pixel as to whether it is part of a specific object in image frames of videos. The capability to successfully segment objects in videos is a key step towards human-level understanding of the surrounding environment for machines.

Recently in video object segmentation (VOS), many have achieved good performance by pretraining on large classification datasets to help the networks learn general objectness [17, 41, 48]. Object proposal is becoming increasingly popular for generating detection boxes for bounded segmentation [14, 24, 28, 50]. Optical flow is commonly used to help networks learn temporal consistency between video frames [17, 18, 25, 42, 46]. Some update the networks online at test time using previous predictions in order to adapt to large changes and keep track of objects during the video sequences [7, 18, 31, 48]. There is also interesting work tackling the task of VOS using unsupervised methods [20, 45].

This paper focuses on the task of semi-supervised video object segmentation, which is the task of pixel-wise classification of object(s) in video sequences given very limited in-domain annotation. The task is extremely challenging due to the scarcity of training data for the target objects in videos during test time. In DAVIS$_{16}$ (single-object) [32] and DAVIS$_{17}$ (multiple-objects)

[35], only the fully annotated ground truth of the first frame is provided for object segmentation in each video sequence. In DAVIS$_{18}$ [34], the Semi-supervised Challenge is the same as DAVIS$_{17}$. DAVIS$_{18}$ also introduces an Interactive Challenge that evaluates both accuracy and the speed of the models, which simulates simple human-like interactive feedback scribbles[1] from the DAVIS server and measures the performance of models over time.

For the task of semi-supervised VOS using only the ground truth of the first frame, we validate our approach on DAVIS$_{16}$ and DAVIS$_{17}$. We propose to improve the recently introduced online adaptation of convolutional neural networks for video object segmentation (OnAVOS) [48]. OnAVOS adopted the architecture proposed by [49], achieved 1st place on DAVIS$_{16}$ and 5th place on DAVIS$_{17}$ with a single online-adaptation pipeline. We include a brief description of OnAVOS in Section 3.0.1. OnAVOS builds on OSVOS [41], which introduces pretraining steps for the network to learn general objectness before fine-tuning on the ground truth mask of the first frame in a specific video at test time. OnAVOS claims that its predecessor lacks the ability to adapt to large changes in video sequences due to its limited knowledge based only on the first frame of videos, and adaptively introduces an approach to update the network during test time, training on previous high-confidence predictions. However, the applied methods for obtaining the high-confidence prediction is rather elementary, simply applying a constant threshold to the foreground logits[2] to extract the confident foreground regions, and a distance transform[3] followed by a very large distance threshold to extract the confident background regions. This approach lacks the ability to adapt across various video sequences due to its handpicked threshold values, which we demonstrate later. More importantly, OnAVOS neglects large portions of effective potential training due to its simplicity in selection of the new masks for online adaptation. Therefore, we propose a new adaptation algorithm that improves the adaptation masks for online training with the utilization of optical flow, which achieves better accuracy without compromising the model complexity. We refer to our approach as Flow Adaptive Video Object Segmentation (FAVOS).

[1] Simulated simple human sketches for feedback that corrects the given segmentation prediction.

[2] Probability map for the output frame.

[3] Operation that transforms a binary image into distance values to the closest foreground boundary.

Comparing with the other state of the art approaches [14, 18, 24, 28, 50], our proposed method achieves competitive accuracy with much lower model complexity and higher efficiency. We show our results and provide complexity analysis in Section 4.0.3.

For the task of semi-supervised VOS given interactive feedback, we validate our approach on DAVIS$_{18}$ Interactive Challenge. The DAVIS$_{18}$ server provides simulated interactive feedback on the worst predicted frame after receiving a complete set of predictions per iteration, and evaluates performance in multiple iterations. Despite the fact that FAVOS is relatively time-efficient, the online training time is still unacceptable for the speed evaluation of the DAVIS 2018 Interactive Challenge. As a result, for the purpose of the challenge, we omit the online training step for each frame from FAVOS and only train on the scribbles of the given frames for limited time in each iteration. A detailed description of our approach is included in Section. 3.0.3.

Chapter 2

Related Work

Segmentation in Computer Vision. Research in the task of image segmentation has history long before the era of deep learning. In 2001, Boykov and Jolly [51] presented a widely used interactive graph cut algorithm for N-dimensional image segmentation, which prompts the user to mark certain pixels for foreground and background regions needed for optimal segmentation. In 2006, Criminisi [6] extended the task of image object segmentation to video object segmentation by using a Spatio-temporal Hidden Markov Model, using both the color likelihood and motion likelihoods with the spatial and temporal priors. Bai [1] later introduced SnapCut in 2009, which is capable of segmenting foreground objects more accurately given the ground truth of the first frame in a video by using localized classifiers on foreground boundaries that track the motion of object with the help of optical flow. Similarly, Price [36] introduced LIVEcut, which uses the graph-cut framework and propagates the user-defined foreground region frame-by-frame, taking cues such as color, gradient, shape, temporal coherence into account while learning the feature-weightings interactively. Traditional computer vision approaches depend on the interactive user-defined inputs for the task of video object segmentation in order to track objects accurately. However, consistent user modification can be tedious, which leads to the need for segmentation algorithms in the field of deep learning.

FCNs for Semantic Segmentation. Solving the task of semantic segmentation using fully convolutional networks (FCNs) was initially proposed by Long et al. [27]. They replace the fully-connected layers in classification networks with 1x1 convolutions so the network becomes fully convolutional. In addition, they define skip connections that share features between different levels in the network to help produce detailed segmentation. Similar usage of skip connections introduced in ResNet have also become state of the art for image classification tasks [15], capable of training very deep neural networks without the vanishing gradient issue. Wu [49] and Zagoruyko [53] argued that deep residual networks experimentally act as ensembles of many shallower networks and proposed shallower but wider residual network models that outperform their predecessors in image classification. Additionally, Wu introduced a slightly modified model for semantic segmentation task which also shows competitive results across multiple datasets.

Video Object Segmentation and Tracking. Unlike semantic segmentation, which only requires segmentation on certain classes, target objects vary in different video sequences for the task of VOS. Therefore, models need to learn to generalize on limited annotation and be able to track the objects throughout videos in VOS. Common traditional approaches tackling the task of VOS use superpixels [5, 13], patches [10, 37] or object proposals [33] in order to compensate for the high dimensionality of the original input space and gain efficiency. Various optimization techniques are then applied to best utilize the connections across the video frames for resulting segmentations.

Recently, as large datasets and computational power become more available, convolutional neural network based approaches [14, 18, 19, 24, 27, 28, 41, 48, 50] have become the state of the art in VOS. Pretraining on large image classification datasets has been proven effective for semi-supervised VOS [41, 48]. Alternatively, Khoreva [18] performs extensive data augmentation on specific videos using the provided first frame ground truth, and argues that training on small sets of data of objects related to specific videos is sufficient to produce good results. Le [44] and Li [25] firstly detect the target objects and perform segmentation on the detected bouding boxes, their methods perform well on re-identifying previously missing objects. [24, 30] use deep models to learn similarity between images and propagate the objects found similar to ground truth to achieve object tracking. The usage of optical flow is also common. Khoreva [19, 50] uses optical flow to propagate previous masks and treats the segmentation task as a mask refinement task. In addition, [18, 19, 25] utilize temporal consistency by feeding the optical flow field as additional input to the models. While others have mostly used optical flow as additional input to variational pipelines, hoping that CNNs would automatically learn the temporal connections between frames, we propose to use precomputed optical flow directly to refine the adaptation masks for accurate new ground truth masks and update the network online to adapt to various changes objects may experience throughout the video.

Interactive Video Object Segmentation. Since for the task of VOS, very limited annotation of the target objects is provided for models to learn to segment throughout a video, in which objects change shapes, sizes, positions and experience occlusion or even reappearance, it is common for deep learning approaches to be time-consuming during the training phase. Additionally, taking interactive feedback from user was very common among the traditional computer vision approaches for the task of VOS. As a result, the DAVIS 2018 Interactive Challenge was introduced to evaluate not only the accuracy, but also the speed of the models. The DAVIS server simulates human-like feedback on the worst predicted frame per iteration to interact with the segmentation model and evaluates the results by Area Under Curve (*AUC*) and J@60s, where the area of an accuracy vs time plot is computed and interpolated so the score at the 60-second mark is also evaluated. For faster training, we omit the online adaptation from FAVOS. Similar to FAVOS, we improve the training data by refining feedback scribbles provided by the DAVIS server and show that with a proper training schedule and improved training ground truth, a simple pipeline can achieve competitive results. Since the DAVIS 2018 Interactive Challenge is introduced as a teaser challenge due to the technical challenges behind hosting the server and validating the results, we only compare our results against the baseline approach.

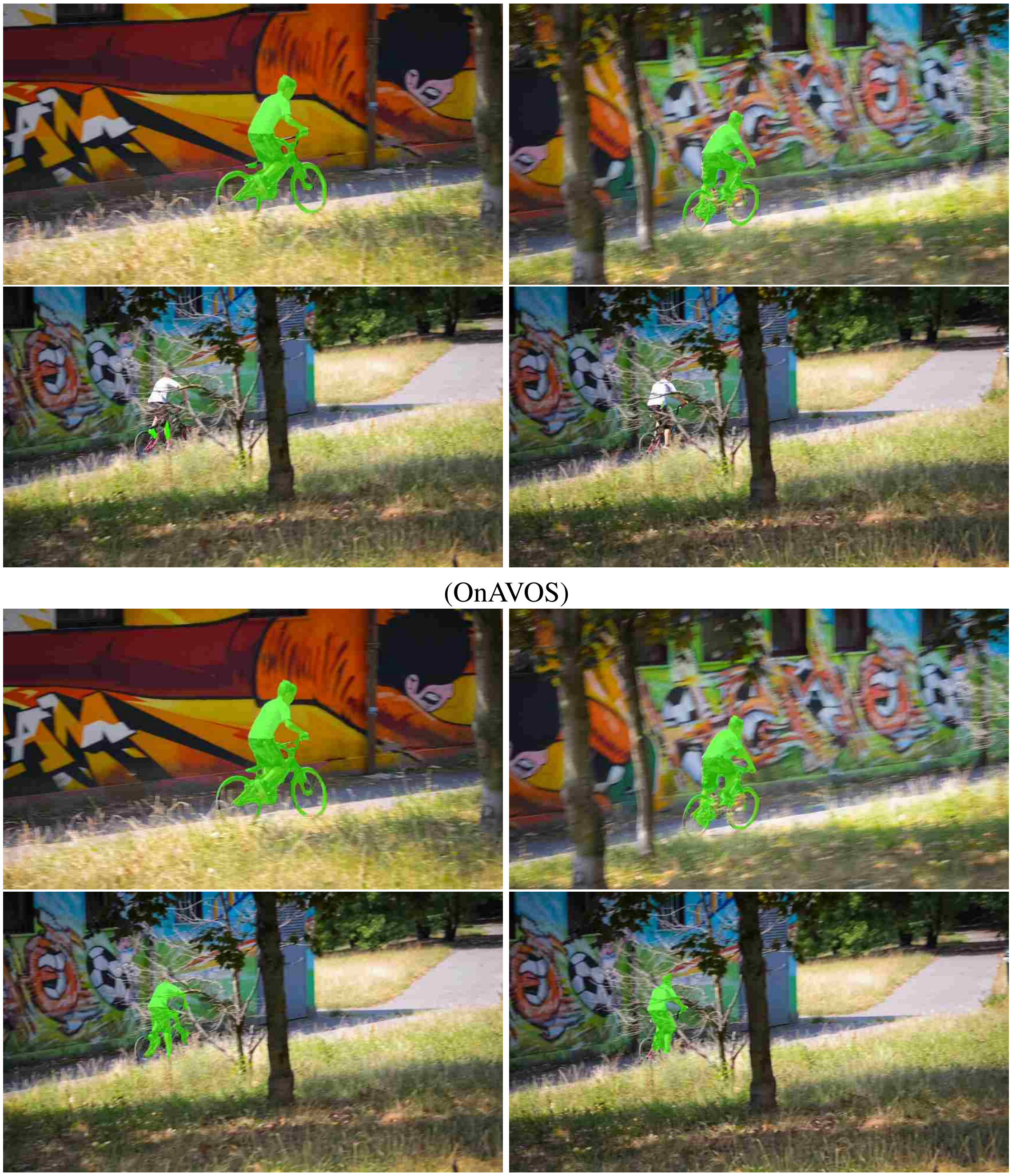
Chapter 3

Approach

3.0.1 OnAVOS

We propose to build upon OnAVOS [48], which uses a base model (following OSVOS [41]) that was pretrained on ImageNet [8] and PASCAL [9] by mapping all 20 annotated classes to foreground and other regions to background. In addition, to gain higher resolutional and domain-specific knowledge on the target dataset, OSVOS and OnAVOS fine-tune on the DAVIS training set. Both approaches demonstrate that the pretraining steps are necessary for promising results. At test time, OSVOS and OnAVOS perform data augmentation and finetune on the ground truth masks of the first frames of targeted video sequences. However, OnAVOS argues that OSVOS is unable to adapt to large changes across the entire sequence, and hence introduces an online adaptation step that continues the training of the network based on its high-confidence predictions. By updating the network online, OnAVOS shows an excellent improvement of score from a mIoU (mean Intersection-over-Union) of 81.7% to 85.7% on DAVIS 2016, and from a mIoU of 76.6% to 77.4% on the Youtube-Objects dataset.

The online adaptation method that OnAVOS uses demonstrates its effectiveness in helping the network to adapt. However, we noticed that the adaptation scheme is rather elementary. First, OnAVOS applies a handpicked probability threshold of $\alpha = 0.97$ to the foreground logits and extracts the high-confidence regions for the foreground adaptation masks. This thresholding method performs well in simple cases where the network outputs very high probabilities on correct pixels; but in difficult cases, it loses the foreground region entirely when the network is not very confident of its predictions (Figure 3.1). Secondly, to extract the confident background masks, OnAVOS



(OnAVOS)

(FAVOS)

Figure 3.1: Comparison between OnAVOS and FAVOS. With adaptation mask refinement, we maintain improved segmentations while OnAVOS loses the target object. **Best viewed in color.**

applies a simple erosion[4] on the selected foreground region, inverts the values and performs a distance transform followed by a thresholding with a constant threshold value of $d = 150$ (treating

[4] Morphological operator that decreases the area of the foreground selection from its boundaries.

pixels far away from the foreground boundary as the background). The unknown regions between the foreground and the background adaptation masks are then not used for training. The intent of this selection method for adaptation masks is so the network will always train on correctly annotated pixels. Experimentally, both OnAVOS and we have observed that if we decrease the thresholding values on OnAVOS to gain more training information, the network starts learning incorrect information about the target pixels and soon makes more incorrect predictions lead to unrecoverable drifts in predictions. Therefore, the adaptation scheme used by OnAVOS is low-risk to apply, yet it neglects much potential information useful for training and sometimes avoids training entirely when the predictions fail to meet the thresholding standards.

3.0.2 FAVOS

We introduce Flow Adaptive Video Object Segmentation (FAVOS) that extracts its high-confidence regions for the online training with the guide of optical flow[5] [16, 39]. Originally, OSVOS states that methods using temporal consistency (including optical flow) work well when the target objects transition very gradually, but fail in cases such as occlusions and abrupt motion. Building on OSVOS, OnAVOS does not utilize any temporal information either. We have also observed that the current top-performing approach [16] for obtaining the optical flow field is far from perfect and its usage could possibly lead to degradation in performance in video object segmentation. The question becomes: how can we utilize the rough optical flow field estimation correctly to improve performance?

The main idea of our approach is to obtain the confident foreground and background regions by utilizing the current prediction and the previous prediction remapped (flow warped) by the optical flow field. The adaptation algorithm used by OnAVOS [48] utilizes the temporal connection between previous and current predictions by using regions too far away from previous foreground prediction as a mask for current background predictions, which is simple yet produces good results. The intuition is that new objects entering the scenes are particularly troublesome to predict since the

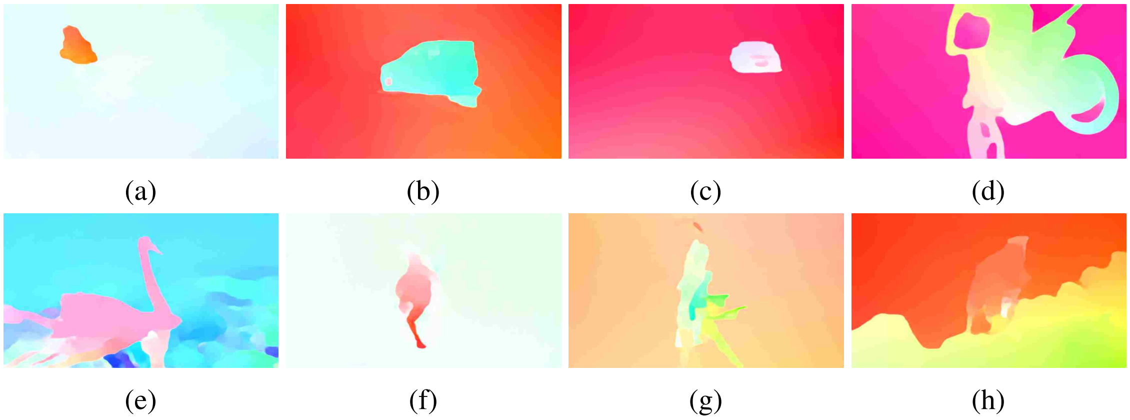
[5] 2-channel vector field that represents the motion of pixels from $frame_t$ to $frame_{t+1}$.

Algorithm 1: Flow-guided Online Adaptation for VOS

```
Input :Pretrained network $N$, confidence percentage $\rho$, initial finetuning steps $n_{init}$, total
       online-training steps $n_{total} = 15$, first frame online-training steps $n_{first} = 2$,
       current adaptation mask online-training steps $n_{cur} = 1$ pre-computed optical flow
       fields $F$, size threshold for noise removal $s = 0.05$.
Fine-tune $N$ for $n_{init}$ steps on $frame(1)$
$lastmask$ ← ground_truth(1)
For $t$ in total_frame_num, do
    For $i$ in total_object_num, do
        $cur\_fg_i$ ← fg_predict($t$) on $object_i$
        $cur\_fg_i$ ← confident_fg($cur\_fg_i$, $\rho$)        #confident foreground
        $flow\_fg_i$ ← flow_warp($lastmask$, $F(t$-1)) on $object_i$
        $confident\_fg_i$ ← create_confident_fg($cur\_fg_i$, $flow\_fg_i$)
        $unsure\_region_i$ ← create_unsure($confident\_fg_i$, $F(t)$)        #unsure region
        $confident\_bg_i$ ← $\sim(confident\_fg_i \cup unsure\_region_i)$        #confident background
    $confident\_adaptation\_mask$ ← combine confident regions from all objects
    If $confident\_adaptation\_mask \neq \emptyset$, then
        interleaved for $n_{total}$ steps:
        train($N$, $n_{cur}$, $confident\_adaptation\_mask$)
        train($N$, $n_{first}$, ground_truth(1))
    End If
    $final\_prediction$ ← noise_removal(fg_predict($t$), $s$)
    $lastmask$ ← $final\_prediction$
```

network has not trained on them as negative examples and therefore outputs high probabilities. By using the previous mask to help determine an approximate foreground region, the false positives far away from previous foreground can be set as background for training before the final prediction. However, OnAVOS had to set the foreground logits threshold value α and background distance threshold value d very high for safe adaptations, therefore leaving out much blank area between the confident foreground and background for useful training.

In order to obtain more informative and still accurate adaptation masks, we have performed numerous experiments which utilize additional information other than the current and previous predictions, particularly the optical flow field (obtained using FlowNet2.0 [16]). Initially, we used the optical flow field in a similar way to [19], which intends to extract a rough segmentation mask of the primary object using the flow field, based on the assumption that objects tend to have consistent motion. This approach works well on videos where the background motion is consistent and the


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.2: Optical flow fields with helpful segmentation information, consistent motion in foreground and background (a,b,c,d). Optical flow fields that have various motion in foreground and background, therefore the target object's segmentation information cannot be extracted (e,f,g,h).

primary target object has a different motion than the background, but fails in most other more general cases (see Figure 3.2). As a result, instead of using the optical flow field for additional segmentation information, we use it as a mapping tool that warps the previous mask to produce a current estimation [18], which we refer to as flow estimation. Ideally, accurate flow fields could potentially enable us to output perfect segmentations throughout the video given the ground truth of the first frame without any deep networks. Unfortunately, from experiments, we observe that the flow estimations are very approximate and cannot output accurate segmentations independently. We conclude that both flow estimation and current prediction are necessary for obtaining better adaptation masks. Algorithm 1 demonstrates our approach, which we also describe in detail in the following section.

Finetuning on the first frame. We use model pretrained on image classification datasets [8, 9] and the DAVIS datasets [32, 35], and finetune the model on specific video sequences using the provided first frame of ground truth [48] (Algorithm 1: $line_1$). For DAVIS 2016 Challenge, since the task of single object segmentation is rather simple, we follow [48] by training on the first frame for n_{init} steps. As for DAVIS 2017 Challenge, the task of multiple-object segmentation is much more difficult, specifically due to a significant amount of data skew which leads to inbalanced training for

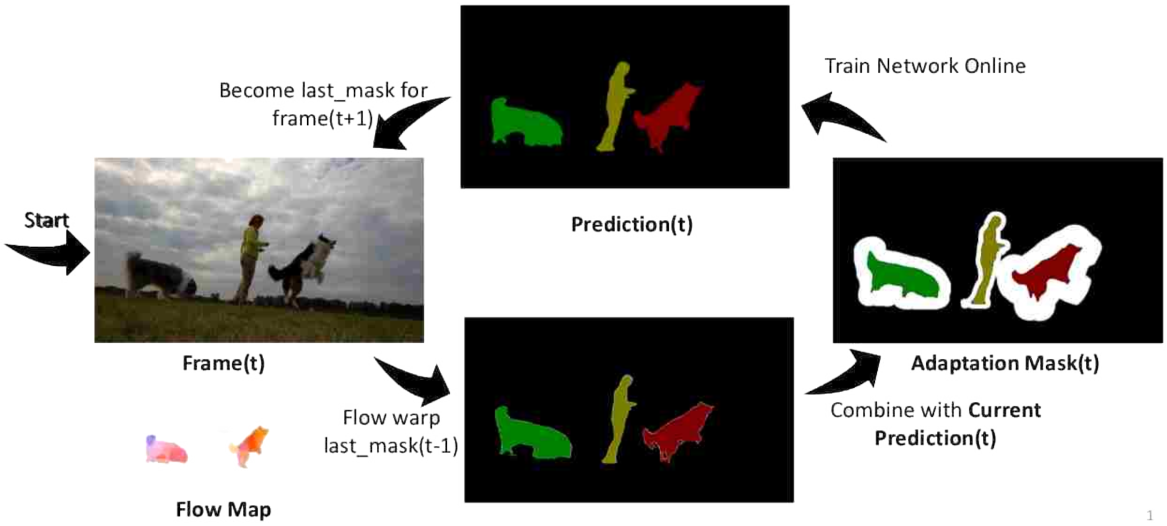


Figure 3.3: FAVOS online adaptation pipeline. We combine the current prediction and flow warpped estimation to create adaptation mask, which we use to finetune the network online for final prediction on each frame. **Best viewed in color.**

objects with different sizes. We address this issue by introducing a two-phase finetuning method. In phase one, we train the model for a maximum of n_{init1} steps on the first frame. If the mIoU score exceeds (1.0/I)*(I-1) where I is the number of target objects or in cases that the maximum step is reached, we will proceed to the next phase. At the beginning of phase two, our model is either capable of recognizing all the target objects, or has trained extensively yet is still unable to recognize all target objects (note that the smallest objects can be only a few pixels in size). We then train the model for a fixed n_{init2} steps in phase two so the model is guaranteed to have domain-specific knowledge on the target video sequence. Our two-phase finetuning method enables the model to adapt to video sequences with various level of difficulty.

$$
\begin{aligned}
ratio_i &= \begin{cases} 1.0 - size_i/size_{fg}, & foreground\ ratio \\ size_{fg}/size_{total}, & background\ ratio \end{cases} \\
scale_i &= max(min(ratio_i, 0.55), 0.45) \\
balanced\ loss &= \sum_{i=0}^{I} scale_i * loss_i \\
final\ loss &= bootstrap\_cross\_entropy(balanced\ loss,\ b)
\end{aligned}
\tag{3.1}
$$

Algorithm 2: Confident foreground selection for adaptation mask

```
Input :Probability logit map for current prediction $L$, foreground mask from current
       prediction $C$, foreground mask from flow warpped previous prediction $P$, $\tau = 0.8$.
$IoU \leftarrow intersection\_over\_union(C, P)$
If $IoU \geq \tau$ Then
    $confident\_fg = IoU$
Else
    $\mu_c = \text{mean}(L[C])$
    $\mu_p = \text{mean}(L[P])$
    If $\mu_c \geq \mu_p$ Then
        $confident\_fg = C$
    Else
        $confident\_fg = P$
```

In addition, we introduce a class-balanced loss function to balance the training between objects with different sizes. As shown in Equation 3.1, we first compute the foreground ratio and background ratio separately. The foreground ratio is determined by the proportion between the size of $object_i$ and the size of the total foreground. The background ratio is then balanced with the total amount of training foreground objects receive. In the case of $I = 1$, we balance the losses by treating background as another foreground object. After obtaining the balanced ratios for the target objects, we limit the ratios to be between 45% and 55% to avoid over-balancing, which can result in excessive training for small objects and insufficient training for large objects respectively. For the last step, we apply a bootstrapped cross-entropy loss [52] on the balanced loss, which selects the hardest $b\%$ pixels for training ($b = 35$). Experimentally, we have discovered that applying bootstrapped cross-entropy loss without class-balancing [48] performs well in general cases, but fails to detect relatively small objects.

Online training using adaptation masks. Figure 3.3 illustrates our approach (Algorithm 1: $line_{4-11}$). To obtain the confident adaptation mask in each frame of the video sequence, our first step is to obtain the current foreground prediction for each object. We use distance transform on the predicted foreground object and select the region with distance values larger than an adaptive threshold determined by a percentile value ρ, such that the inner $\rho\%$ of the initial foreground is selected as confident foreground region. Figure 3.1 shows the improvement of our approach compared with applying a constant threshold on the foreground probability map [48]. The second step is to warp the previous foreground prediction using the optical flow field to produce flow estimation for each object. We then generate the confident foreground region using both the current prediction and the flow estimation as described in Algorithm 2, which checks the agreement between the current prediction and the flow estimation by using the IoU and selecting the confident foreground region accordingly. To avoid training on incorrect pixels, which can immediately lead to escalated errors in future predictions, we insert an unsure layer where no training takes place. The unsure layer is generated by applying a distance transform and a distance threshold on the confident foreground region, so pixels that are within the range of d pixels from the foreground are selected as unsure. The value of d is adaptively determined by the optical flow magnitude for the target object. Finally, the confident background region is simply the region outside of the unsure region. After obtaining the confident regions for all objects, we combine them to produce the final adaptation mask for the current frame. In general, we want to maximize the confident foreground/background region and minimize the number of incorrectly labeled pixels for the adaptation mask. As for online adaptation (Algorithm 1: $line_{12-16}$), we iteratively fine-tune the network by training on the adaptation mask of the current frame (for n_{cur} steps) and the ground truth of the first frame (for n_{first} steps). Re-training on the first frame is significant since the current adaptation masks can be inaccurate and the network needs to retain the knowledge of the target object. After the online adaptation, we produce a final prediction mask for the current frame by averaging the pixel-wise posterior probabilities over 10 runs for more stability [47]. For DAVIS 2016 dataset, we perform DenseCRF [22] in the same fashion as OnAVOS. For DAVIS 2017/2018 dataset, the only post-processing required for the final prediction is using connected-components labeling[6] for noise removal. We remove small components which fail to exceed a size threshold of $s = 5\%$ of the largest component for the corresponding object class.

[6] Technique used to detect connected regions in binary digital images.

3.0.3 Interactive Segmentation

The DAVIS 2018 Interactive Challenge introduces a new way of evaluating segmentation models, which measures both accuracy and speed. During the testing phase, the server allocates fixed time intervals available for each video, where $T = \#Iterations \times \#Objects \times 30s$ (*#Iterations* = 8, *#Objects* is the number of target objects in a specific video sequence). For each new video sequence, the server provides the model with 1 of 3 predefined human-annotated scribbles, which only consists of the data points that belong to the corresponding objects of a particular frame. The first scribble is special since it provides the most information for the target objects. For each new iteration, the server generates a new simulated scribble that consists of data points in key misclassified regions of the worst predicted frame from previous predictions and waits for a new set of predictions on all video frames. Testing for one video is finished if either the model runs out of time or has finished all 8 iterations. The accuracies achieved in iterations are recorded for video sequences. The overall accuracy vs time plot is generated by interpolating and averaging scores from all videos, after which *AUC* and J@60s are computed and compared between models.

Given the number of objects that range from 1 to 10 in videos of the DAVIS 2018 Interactive Challenge Set, the time available for models to train and predict on a video can range from 4 to 40 minutes. In order to take advantage of the maximum number of simulated scribbles generated from the server in the allocated time, a model needs to aim for an average time of $\#Objects \times 30s$ per iteration. In the case of $\#Objects = 1$, the model has 30 seconds for each iteration and 4 minutes in total for training and predicting the entire video sequence, which is an extremely short amount of time for deep learning models. As a result, we omit the online-training part of FAVOS for faster response time. We are the only team that participated in both the DAVIS 2018 Semi-supervised Challenge and the DAVIS 2018 Interactive Challenge.

Improved training on the first scribble. Unlike simulated scribbles for the following iterations, the first scribbles are predefined by human annotators. For each video sequence, there are 3 first scribbles on different frames that are proper representations of the entire sequence, each manually

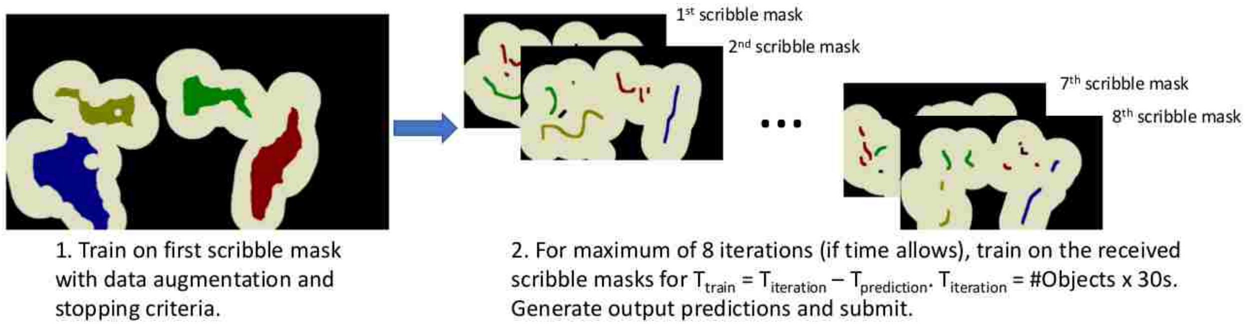

1st scribble mask
2nd scribble mask
7th scribble mask
8th scribble mask
...
1. Train on first scribble mask with data augmentation and stopping criteria.
2. For maximum of 8 iterations (if time allows), train on the received scribble masks for $T_{train} = T_{iteration} - T_{prediction}$. $T_{iteration}$ = #Objects x 30s. Generate output predictions and submit.

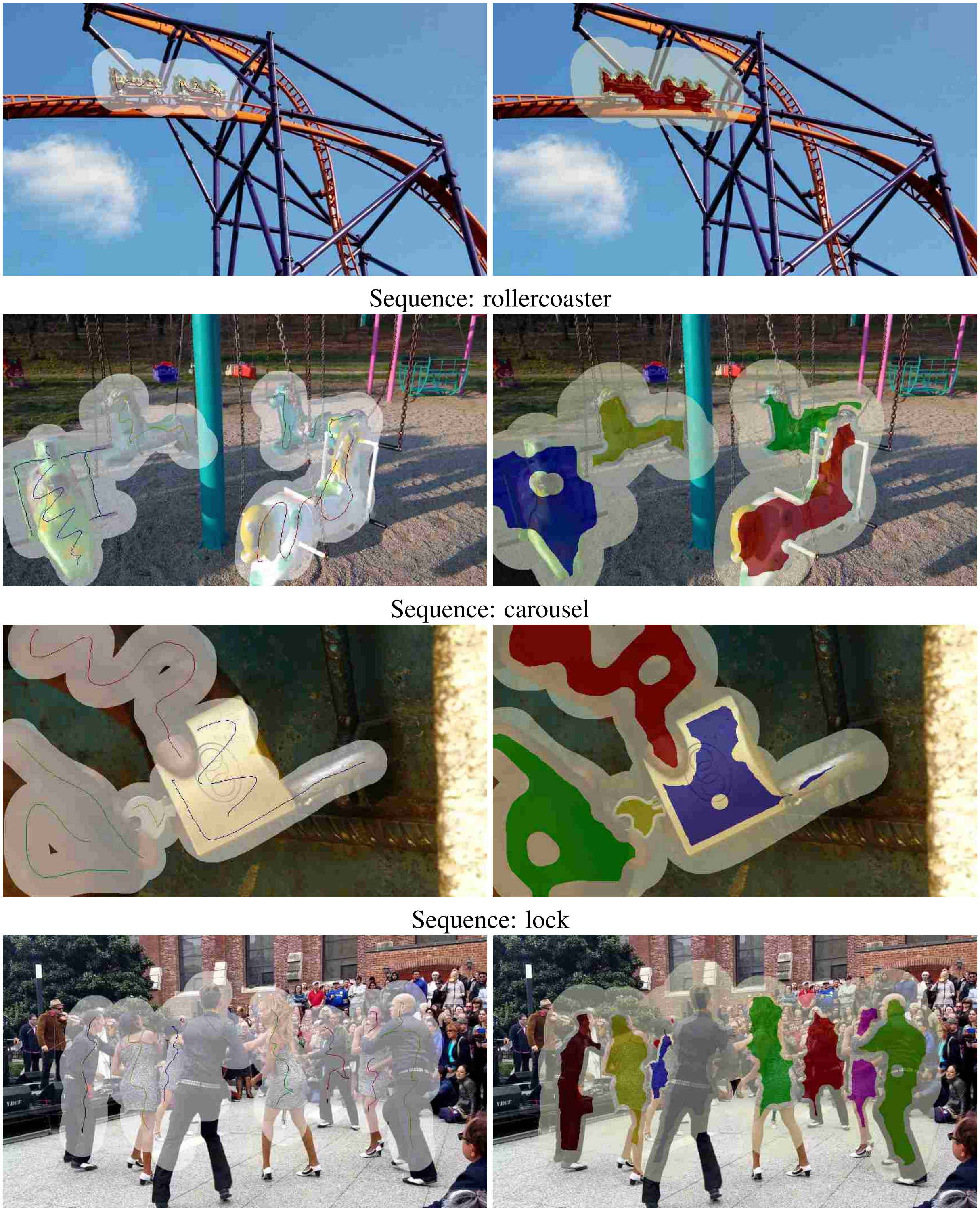
Figure 3.4: Scheduled training procedure for DAVIS 2018 Interactive Challenge. We train on drastically improved first scribble mask for faster and better first submission. For the following iterations, we train on the pool of received scribble masks for a training time that allows maximum number of iterations.

sketched by a human annotator with a certain style. For segmentation, each ground truth pixel contributes to the weight updates. So we improve the first scribbles to help the model train on more annotated pixels, achieving better and faster predictions for its first submission. As illustrated in Figure 3.5, the original first scribbles contain very limited information, yet have great potential for improvements. For all scribbles, we apply a simple dilation with size 7 on the original target regions and the same technique as Section 3.0.2 to generate unsure regions. Additionally, for first scribbles, we apply component-based GrabCut[7] [40] to select the best object-like regions that contain the original scribbles. We treat the given scribbles as confident foreground, and regions outside of the unsure region as confident background. Finally, the inner $\rho\%$ of the target region is extracted for safe training. Due to the simplicity of the given scribbles and imperfection of the GrabCut results, we only apply this technique on the first scribbles for faster first submission. Although simulated scribbles generated by the DAVIS server provide limited information, they are guaranteed to be accurate and thus are safer for future training. We refer to the improved scribble images for training as scribble masks.

Scheduled training on scribble masks. Figure 3.4 illustrates our training procedure. For each video sequence, in order to have a fast and accurate first submission, which defines the be-

[7] Technique used to select refined foreground and background regions on images given user-defined confident foreground, background and unsure region.

ginning score in the accuracy vs time plot and has a significant impact on the *AUC*, we train the model on the improved first scribble mask with a stopping criteria. We stop training on the first scribble mask once the average training loss ceases to decrease for 20 steps, then generate the output predictions. This stopping criteria allows the model to adapt to video sequences with various levels of difficulty. Maximum prediction time is recorded so that we can compute the training time available for future iterations by subtracting the prediction time from the iteration time. During the training phase of each iteration, we randomly select from the pool of received scribble masks and train for as many steps as possible for the given time. Data augmentation is applied in the same fashion as Section 3.0.2. This scheduled training procedure aims to generate a decent first submission, and maximize the number of iterations under limited time, which allows the server to provide more simulated feedback scribbles for improvement.



Sequence: rollercoaster

Sequence: carousel

Sequence: lock

Sequence: salsa

Figure 3.5: Comparison between original first scribbles (left column) and our improved first scribbles (right column). Colors shown are the original coloring in the annotations of DAVIS dataset for different target objects. Given simple scribbles, we drastically increase the total number of annotated pixels for training. Unsure regions and annotated regions are labeled as white and different targets' corresponding colors with 0.50 and 0.70 opacity respectively for better visualization. **Best viewed in color.**

Chapter 4

Experiments

4.0.1 Datasets

This paper focuses on the task of semi-supervised video object segmentation. Specifically, we test on the recently introduced DAVIS (Densely Annotated VIdeo Segmentation [32, 34, 35]) datasets. DAVIS$_{16}$ (single-object segmentation) [32] contains 30 fully annotated video sequences for training and 20 video sequences that only provide the first frame of annotation for validation. DAVIS$_{17}$ [35] introduces multiple-object segmentation. It contains 60 fully annotated video sequences for training, and 30 video sequences for each of the validation, test-development and test-challenge sets. DAVIS$_{18}$ [34] repeats the semi-supervised challenge in DAVIS$_{17}$ and additionally introduces interactive video object segmentation by providing simulated human-like interaction from a server and measuring both the accuracy and the speed of the model. The DAVIS 2017 Test-development set is used for the interactive track.

4.0.2 Experimental Setup

For DAVIS$_{16}$ and DAVIS$_{17}$, we use pretrained models provided by OnAVOS and apply the same learning rates ($\lambda = 3 \cdot 10^{-6}$ for finetuning phase, $\lambda = 10^{-5}$ for online adaptation). During training, we also augment the training data by random scaling with factors uniformly distributed in [0.7, 1.3]. However, we omitted the random flipping from data augmentation since the location of instances at a particular time is important information for multiple-object segmentation. On DAVIS$_{16}$, we finetune the model on the first frame for $n_{init} = 50$ steps [48]. The confidence percentage value ρ is set to 85%. On DAVIS$_{17}$, we use two-phase finetuning with $n_{init1} = 2000$,

Measure	FAVOS	OnAVOS	OSVOS-S	CINM	RGMP	FAOVOS	OSVOS	MSK	PML	SFL
Region J (Mean)	**0.869**	0.861	0.856	0.834	0.815	0.824	0.798	0.797	0.755	0.761
Region J (Recall)	**0.970**	0.961	0.968	0.949	0.917	0.965	0.936	0.931	0.896	0.906
Region J (Decay)	**0.041**	0.052	0.055	0.123	0.109	0.045	0.149	0.089	0.085	0.121
Boundary F (Mean)	0.850	0.849	**0.875**	0.850	0.820	0.795	0.806	0.754	0.793	0.760
Boundary F (Recall)	0.899	0.897	**0.959**	0.921	0.908	0.894	0.926	0.871	0.934	0.855
Boundary F (Decay)	**0.046**	0.082	0.058	0.147	0.101	0.055	0.150	0.090	0.078	0.104

(a) DAVIS 2016 Validation Results. Top 10 are shown from a total of 20 teams.

Measure	PReMVOS	DyeNet	ClassAgno	OnlineGen	LucidTrack	HCMUS	FAVOS	TeamVia	Kthac	Huber99
Global Mean	**0.747**	0.738	0.697	0.695	0.678	0.663	0.606	0.601	0.589	0.545
Region J (Mean)	0.710	**0.719**	0.669	0.675	0.651	0.641	0.584	0.577	0.567	0.518
Region J (Recall)	**0.795**	0.794	0.741	0.770	0.725	0.750	0.656	0.649	0.631	0.564
Region J (Decay)	0.190	0.198	0.231	0.150	0.277	**0.117**	0.262	0.272	0.307	0.257
Boundary F (Mean)	**0.784**	0.758	0.725	0.715	0.706	0.686	0.629	0.624	0.611	0.572
Boundary F (Recall)	**0.867**	0.830	0.803	0.822	0.798	0.807	0.710	0.717	0.676	0.625
Boundary F (Decay)	0.208	0.203	0.259	0.185	0.302	**0.135**	0.297	0.281	0.331	0.295

(b) DAVIS 2018 Semi-supervised Challenge Results. Top 10 are shown from a total of 18 teams.

Table 4.1: Experimental results on DAVIS$_{16}$, DAVIS$_{17}$ and DAVIS$_{18}$. Our results are marked in blue.

$n_{init2} = 800$ and $\rho = 75\%$. We select the hyperparameters using the provided validation sets. Experimentally, the values for the hyperparameters are not sensitive and values within reasonable ranges produce similar results.

For DAVIS$_{18}$ Interactive Challenge, we use $\lambda = 3 \cdot 10^{-5}$ for training in first iterations, and $\lambda = 1.5 \cdot 10^{-5}$ for training in the following iterations. The confidence percentage value $\rho = 60\%$ for selecting confidence foreground from GrabCut results. For the insertion of unsure layers around confident foreground region, the distance threshold d = 60 for first scribble masks and d = 100 for simulated scribble masks due to higher level of uncertainty.

In the DAVIS$_{16}$ Challenge and DAVIS$_{17}$ Test-Challenge, both the region and boundary scores as measured as described in [32, 35]. For the DAVIS$_{18}$ Interactive Challenge, *AUC* is computed from the generated region score vs time plot. Additionally, region score at 60-second mark (J@60s) is also evaluated by interpolating the *AUC*.

4.0.3 Results

Single-object VOS. The DAVIS$_{16}$ Challenge ranks methods based on the region score (mIoU). As illustrated in Table 4.1a, we achieve rank 1 in the DAVIS 2016 Challenge with a mIoU of 0.869. Given only the annotation of the first frames in videos, it is challenging for models to generalize and maintain the initial accuracies. We achieve the most stable performance by having a region decay of 0.041 and boundary decay of 0.046.

Multiple-object VOS. The DAVIS$_{17/18}$ Semi-supervised Challenge is much more challenging than DAVIS$_{16}$ Challenge due to the introduction of multiple instance segmentation. Table 4.1b shows that we achieve the 7th place out of 18 teams on the DAVIS 2018 Test-Challenge with global score (average of region and boundary) of 0.606. However, despite of having slightly lower accuracy compared with the top-performing approaches, FAVOS has much lower model complexity and higher efficiency. Speed evaluation is not provided by most of the models, so we provide a study on the pipelines used by the top 5 approaches in the DAVIS 2018 Semi-supervised Challenge for complexity analysis. As shown in Table 4.2, we have concluded a summary of major techniques applied. In-domain data augmentation is applied in all studied approaches for more training data on the target objects. Most approaches use technique proposed by [18] for data augmentation. Both PReMVOS [28] and LucidTrack [18] generates 2.5k augmented images for pretraining, while FAVOS only pretrained on 100 augmented images. The top 4 models rely on an object proposal network to generate bounding boxes prior to the object segmentation, which requires per-video training for target-specific detection or object searching for general object detection. Online adaptation is applied for better tracking on the target objects by almost all studied models. Interestingly, DyeNet [24] utilizes similarity between frames and the propagation network, therefore it is not strictly dependent on domain knowledge and online adaptation. PReMVOS [28] and ClassAgno [50] both generate coarse predictions followed by a refinement network, which adds another deep model to their overall pipelines. Approaches that apply instance-based segmentation have time complexity that grows approximately linearly with the number of target objects in the video se-

	PReMVOS[28]	DyeNet[24]	ClassAgno[50]	OnlineGen[14]	LucidTrack[18]	FAVOS
In-domain Data Augmentation	✓	✓	✓	✓	✓	✓
Object Proposal Network	✓	✓	✓	✓	–	–
Online Adaptation	✓	Optional	✓	✓	–	✓
Refinement Network	✓	–	✓	–	–	–
Instance-based Segmentation	✓	✓	✓	✓	–	–
Optical Flow	✓	✓	✓	–	✓	✓
Ensemble	✓	–	–	–	✓	–
Temporal Consistent Merging	✓	✓	✓	–	–	–
Multiple Iterations	–	✓	–	✓	–	–

Table 4.2: Comparison of the main pipelines used for the top 5 models in DAVIS$_{18}$ Semi-supervised Challenge with FAVOS for complexity analysis. FAVOS is much simpler in terms of model complexity and also achieves competitive results. Detailed analysis is included in Section 4.0.3

quences. Inevitably, the object-merging algorithm for combining the predictions for all objects is needed for post-processing. Optical flow is used by all studied models except for OnlineGen [14], which does not depend on temporal consistency for object tracking. Ensembles can be used for improved predictions at the cost of model complexity and time. PReMVOS uses an ensemble of models trained on the top 11 sets of parameters and LucidTrack obtained results via an ensemble of 4 different models. After finishing the predictions for all frames in a video, top 3 models refine the set of predictions using merging algorithms for better temporal consistency. PReMVOS selects and links per-frame object proposals using optical flow, ReID embeddings, objectness scores and other present objects; DyeNet generates multiple streams of object predictions as a result of bi-directional propagation from multiple starting points in videos, and merges the streams of predictions by cosine similarities; ClassAgno applies a CNN-based spatio-temporal MRF [2] to refine the generated masks. Finally, DyeNet and OnlineGen improve their results by refining their models for multiple iterations through their entire pipelines.

To summarize, FAVOS produces competitive results with a much simpler pipeline compared with the other top approaches. Among the top 5 models, LucidTrack is the only approach with similar model complexity as FAVOS. They reported a training time of ∼3.5 hours per video. Our approach has a total training and testing time of ∼45 minutes per video, which is nearly a fifth of the timing of LucidTrack. We also show that higher model complexity does not always lead to higher accuracy. For evaluation on the DAVIS_{16} Challenge for single-object segmentation, PReMVOS and DyeNet reported mIoU of 86.8 and 86.2 respectively, while FAVOS achieves a mIoU of 86.9.

Interactive VOS. The Interactive Challenge provides only simple scribbles instead of fully annotated first frames in videos, therefore models have even more limited initial information. However, unlike the Semi-supervised Challenge, models can receive information on multiple frames from interactive feedback, which can potentially lead to better performance. We have observed that for the task of VOS, training on multiple informative scribbles can achieve comparible results as training on the fully annotated ground truth of the first frame. We compare our approach with the baseline since this challenge is very experimental. We achieve an *AUC* of 0.450 with J@60s of 0.230 and outperform the baseline of *AUC* = 0.299 with J@60s of 0.141 by a significant margin. Due to the time limit for iterations in the setup of DAVIS 2018 Interactive Challenge, we had to omit online adptation from FAVOS.

Chapter 5

Conclusion

In this work, we propose FAVOS for the task of Semi-supervised Video Object Segmentation. We have improved over OnAVOS, introducing a new pipeline that performs online adaptation with the utilization of optical flow and achieves better accuracy without increasing the model complexity. We show that our approach achieves competitive results on the DAVIS datasets with simplicity and efficiency. With interactive user feedback for practical applications, FAVOS is capable of adapting to various challenging changes in objects throughout the video and provide nearly perfect segmentation.

We also adapt FAVOS for the task of Interactive Video Object Segmentation. We demonstrate that fast and accurate learning from simple scribbles can be achieved by proper scheduled training on improved scribble masks.

References

[1] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: Robust video object cutout using localized classifiers. *Proc. ACM SIGGRAPH*, 28(3):70:1–70:11, 2009.

[2] 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 CVPR*, pages 5977–5986, 2018.

[3] G. Bertasius, J. Shi, and L. Torresani. High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. *In ICCV*, pages 504–512, 2015.

[4] G. Bertasius, J. Shi, and L. Torresani. Semantic segmentation with boundary neural fields. *In CVPR*, pages 3602–3610, 2016.

[5] J. Chang, D. Wei, and J. W. F. III. A video representation using temporal superpixels. *In CVPR*, pages 2051–2058, 2013.

[6] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. *In CVPR*, pages 53–60, 2006.

[7] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. *In ECCV*, 2016.

[8] J. Deng, W. Dong, R. Socher, L. J.Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *In CVPR*, pages 248–255, 2009.

[9] M. Everingham, S. M.A.Eslami, L. VanGool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *IJCV, 111(1)*, 111(1):98–136, 2015.

[10] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut : Non-successive mask transfer and interpolation for video cutout. *In SIGGRAPH*, 2015.

[11] R. Girshick. Fast R-CNN. *In ICCV*, pages 1440–1448, 2015.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *In CVPR*, pages 580–587, 2014.

[13] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Effcient hierarchical graph-based video segmentation. *In CVPR*, 2010.

[14] P. Guo, L. Zhang, H. Zhang, X. Liu, H. Ren, and Y. Zhang. Adaptive video object segmentation with online data generation. *In CVPR Workshops*, 2018.

[15] K. He, S. R. X. Zhang, and J. Sun. Deep residual learning for image recognition. *In CVPR*, pages 770–778, 2016.

[16] 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*, pages 1647–1655, 2017.

[17] 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.

[18] A. Khoreva, R. Benenson, T. B. E. Ilg, and B. Schiele. Lucid data dreaming for object tracking. *In arXiv preprint arXiv: 1703.09554*, 2017.

[19] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. S. Hornung. Learning video object segmentation from static images. *In CVPR*, pages 3491–3500, 2017.

[20] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. *In CVPR*, pages 7417–7425, 2017.

[21] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. *In ICLR*, 2016.

[22] P. Krhenbhl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. *In NIPS*, pages 109–117, 2011.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *In NIPS*, pages 1097–1105, 2012.

[24] X. Li and C. C. Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. *arXiv:1803.04242*, 2018.

[25] X. Li, Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, C. C. Loy, and X. Tang. Video object segmentation with re-identification. *In CVPR Workshops*, 2017.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. SSD: Single shot multibox detector. *In ECCV*, pages 21–37, 2016.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *In CVPR*, pages 3431–3440, 2015.

[28] J. Luiten, P. Voigtlaender, and B. Leibe. PReMVOS: Proposal-generation, Refinement and Merging for Video Object Segmentation. *arXiv:1807.09190*, 2018.

[29] K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. V. Gool. Convolutional oriented boundaries. *In ECCV*, pages 580–596, 2016.

[30] M. Najafi, V. Kulharia, T. Ajanthan, and P. Torr. Similarity learning for dense label transfer. *In CVPR Workshops*, 2018.

[31] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. *In CVPR*, pages 4293–4302, 2016.

[32] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. S. Hornung. A benchmark dataset and evaluation methodology for video object segmentation. *In CVPR*, pages 724–732, 2016.

[33] F. Perazzi, O. Wang, M. Gross, and A. S. Hornung. Fully connected object proposals for video segmentation. *In ICCV*, pages 3227–3234, 2015.

[34] J. Pont-Tuset, S. Caelles, F. Perazzi, A. Montes, K.-K. Maninis, Y. Chen, and L. V. Gool. The 2018 DAVIS challenge on video object segmentation. *arXiv preprint arXiv: 1803.00557*, 2018.

[35] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbelaez, A. S. Hornung, and L. V. Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[36] B. Price, B. Morse, and S. Cohen. Livecut: Learningbased interactive video segmentation by evaluation of multiple propagated cues. *In ICCV*, pages 779–786, 2009.

[37] S. A. Ramakanth and R. V. Babu. Seamseg: Video object segmentation using patch seams. *In CVPR*, pages 376–383, 2014.

[38] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *In CVPR*, pages 6517–6525, 2017.

[39] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. *In CVPR*, pages 1164–1172, 2015.

[40] C. Rother, V. Kolmogorov, and A. Blake. ”Grabcut”: Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Trans. Graphics*, 23(3):309–314, 2004.

[41] S.Caelles, K.-K.Maninis, J.Pont-Tuset, L.Leal-Taix, D.Cremers, and L. V. Gool. One-shot video object segmentation. *In CVPR*, pages 5320–5329, 2017.

[42] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical flow with semantic segmentation and localized layers. *In CVPR*, pages 3889–3898, 2016.

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In ICLR*, pages 1–14, 2015.

[44] T.-N, Le, K.-T. Nguyen, M.-H. Nguyen-Phan, T.-V. Ton, T.-A. N. (2), X.-S. Trinh, Q.-H. Dinh, V.-T. Nguyen, A.-D. Duong, A. Sugimoto, T. V. Nguyen, and M.-T. Tran. Instance re-identification flow for video object segmentation. *In CVPR Workshops*, 2017.

[45] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. *In ICCV*, pages 4491–4500, 2017.

[46] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. *In CVPR*, pages 3899–3908, 2016.

[47] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. *In CVPR Workshops*, 2017.

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

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

[50] S. Xu, L. Bao, and P. Zhou. Class-agnostic video object segmentation without semantic re-identification. *In CVPR Workshops*, 2018.

[51] M. P. J. Y. Boykov. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *In ICCV*, pages 105–112, 2001.

[52] C. S. Z. Wu and A. v. d. Hengel. Bridging category-level and instance-level semantic image segmentation. *arXiv:1605.06885*, 2016.

[53] S. Zagoruyko and N. Komodakis. Wide residual networks. *In BMVA*, pages 87.1–87.12, 2016.