Distractor-Aware Video Object Segmentation

Andreas Robinson(✉), Abdelrahman Eldesokey, and Michael Felsberg

Linköping University, Linköping, Sweden
{andreas.robinson,abdelrahman.eldesokey,michael.felsberg}@liu.se

Abstract. Semi-supervised video object segmentation is a challenging task that aims to segment a target throughout a video sequence given an initial mask at the first frame. Discriminative approaches have demonstrated competitive performance on this task at a sensible complexity. These approaches typically formulate the problem as a one-versus-one classification between the target and the background. However, in reality, a video sequence usually encompasses a target, background, and possibly other distracting objects. Those objects increase the risk of introducing false positives, especially if they share visual similarities with the target. Therefore, it is more effective to separate distractors from the background, and handle them independently.

We propose a one-versus-many scheme to address this situation by separating distractors into their own class. This separation allows imposing special attention to challenging regions that are most likely to degrade the performance. We demonstrate the prominence of this formulation by modifying the learning-what-to-learn [3] method to be distractor-aware. Our proposed approach sets a new state-of-the-art on the DAVIS 2017 validation dataset, and improves over the baseline on the DAVIS 2017 test-dev benchmark by 4.6% points.

1 Introduction

Semi-supervised video object segmentation (VOS) aims to segment a target throughout a video sequence, given an initial segmentation mask in the first frame. This task can be very challenging due to camera motion, occlusion, and background clutter. Several deep learning based methods have been proposed recently to address these challenges [3,8,9,11,12,15]. Among those, discriminative methods [3,11] have shown competitive performance at a reasonable computational cost, making them suitable for real-time applications, e.g., enhancing visual object tracking in crowded scenes, removing or replacing the background in video sequences or live conference calls, for privacy masking in surveillance videos, or as an attention mechanism in downstream vision tasks such as action recognition.

The majority of the discriminative approaches formulate the problem as a one-versus-one classification between the target and the background. Based on this, they attempt to construct a robust representation of the target that is as distinct as possible from the background. However, the background usually includes other objects that

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-030-92659-5_14.

C. Bauckhage et al. (Eds.): DAGM GCPR 2021, LNCS 13024, pp. 222–234, 2021.
https://doi.org/10.1007/978-3-030-92659-5_14

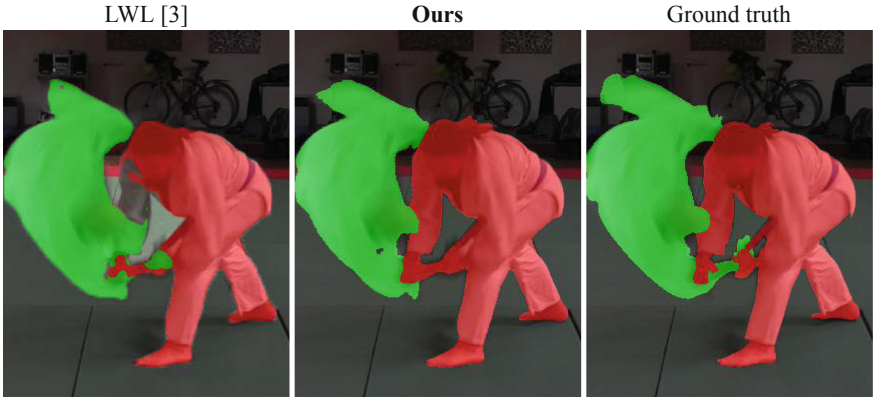

Fig. 1. The impact of incorporating distractor-awareness into the baseline (LWL [3]). Our distractor-aware approach produces more accurate predictions than the baseline in highly ambiguous regions, where objects share visual similarities.

could be visually similar to the target. In this case, it might be challenging to find a good representation that discriminates the target from those distracting objects. This can produce false positives as the classifier is likely to fail given the underlying coarse representation between the target and the background. Figure 1 shows an example where a top-performing discriminative approach fails to discriminate between the target and other objects of the same type.

In this paper, we address this aforementioned challenge by reformulating the problem as a one-versus-many classification. We propose to separate the distracting objects from the background, and handle them as a distinct class. As a result, making the network aware of these distractors during training, promotes the learning of a robust representation of the target that is more discriminative against both the background *and* distractors.

We demonstrate the effectiveness of our approach by modifying the learning-what-to-learn (LWL) approach [3] to become distractor-aware. First, we modify the learning pipeline to incorporate information about the distractors. In case of videos with multiple objects, we initialize other objects in the scene as distractors.

Second, to enhance the discriminative power of the network, we integrate high-resolution features to the target model. Finally, we introduce the use of adaptive refinement and upsampling [13] as a regularization to enforce local consistency at uncertain regions such as edges, and object-to-object boundaries.

Experiments show that our proposed framework sets a new state-of-the-art result on the DAVIS 2017 *val* dataset [10]. Moreover, we improve the results over the baseline on the DAVIS 2017 *test-dev* benchmark by a large margin. On the YouTube-VOS 2018 dataset [14], which is characterized by limited annotation accuracy at object-to-distractor boundaries, our method still shows significant improvement over the baseline.

The remainder of the paper is organized as follows: we start by providing an overview of existing discriminative VOS approaches in the literature, distractor-awareness in other vision tasks, and existing upsampling and refinement methods in

VOS. Next, we briefly describe the baseline approach, Learning-what-to-learn [3], followed by our proposed distractors modelling. Finally, we provide quantitative and qualitative results for our proposed approach in comparison with the baseline, and existing state-of-the-art methods as well as an ablation study.

2 Related Work

The video object segmentation (VOS) task can be tackled in a semi-supervised or an unsupervised manner, but we only consider the former in this paper. Semi-supervised approaches from the literature can usually be categorized as either generative or discriminative. Generative approaches such as [6], focus on constructing a robust model of the target of interest ignoring other objects in the scene. In contrast, discriminative approaches [11, 12] attempt to solve the task as a classification problem between the target and the background. A more recent method [3] follows an embedding approach to learn features that are as discriminative as possible. With the emergence of deep learning, the robustness of feature representations has significantly improved, boosting the performance of most variants of VOS approaches. However, the current top-performing semi-supervised VOS methods are mainly discriminative, taking into account both the target and the background when solving the task. Therefore, we focus on discriminative approaches in this paper.

Discriminative Video Object Segmentation. Discriminative VOS methods were introduced quite recently. Yang *et al.* [15] proposed to build a target model from separate target and background feature embeddings, extracted from past images and target masks. For test frames, extracted features are matched against the two pretrained models of the target and the background. STM [9] incorporates a feature memory, encoding past images and segmentation masks. Similarly, Lu *et al.* [7] introduce an episodic graph memory unit to mine newly available features, and update the model accordingly. In contrast to Yang *et al.*, both methods [7, 9] produce the final target mask using a dedicated decoder network, from concatenated memory features and new image features. Seong *et al.* [12] extended STM [9] further with a soft Gaussian-shaped attention mask to limit confusion with distant objects. Robinson *et al.* [11] introduced the use of discriminative correlation filters to construct a target model that produces a coarse segmentation mask. This coarse mask is then enhanced and refined through a decoder network. Learning-what-to-learn [3] improved it further by learning to produce target embeddings that are more reliable for training the target model. All of these aforementioned papers adopt a one-vs-one classification between the target and the background, where other objects in the sequence are considered as background. In contrast, our approach reduces the likelihood of predicting false positives when some background objects share visual similarities with the target.

Distractor-Aware Modelling. Distractor-aware modeling can be realized as a kind of hard-example mining when training a model. The general concept is to identify inputs that are more likely to confuse a given model and emphasis them during training, One of the earliest uses of this concept is found in the classical human detection algorithm, histograms of oriented gradients [4]. A more recent example [5] proposed marking flickering object detections in a video as hard-negatives, assigning them higher priority during training. A recent visual object tracking method [2] ranks target proposals

based on their signal strength, where the strongest is assumed to be the target and the rest are distractors. The method tracks the complete scene state in a dense vector field over all spatial locations, using it to classify regions as either target, distractor or background. A similar Siamese-based approach [16] also classifies a target and any distractors through ranking of detection strengths. In both approaches, distractors are provided as hard examples during online training their respective tracker target models. Zhu *et al.* [16] also introduced distractor awareness to Siamese visual tracking networks as they noticed that the standard trackers posses imbalanced distribution of training data leading to less discriminative features. They propose an effective sampling strategy to make the model more discriminative towards semantic distractors. In this paper, we follow the strategy adopted by these approaches, and we introduce distractor-awareness to the task of video object segmentation.

Segmentation Mask Upsampling and Refinement. Existing VOS CNNs employ different upsampling and refinement approaches to provide the final segmentation mask at the full resolution. RGMP and STM [8,9] employ two residual blocks with a bilinear interpolation in between for refinement and upsampling. Seong *et al.* [12] used a residual block followed by two refinement modules with skip connections. In contrast, Robinson *et al.* and Bhat *et al.* [3,11] replace the residual blocks with standard convolution, and employ bicubic rather than bilinear interpolation. All these approaches provide spatially independent prediction with no regularization to enforce local consistency, especially at uncertain regions such as edges and object boundaries. We employ the convex upsampler [13] to jointly upsample the final mask while enforcing spatial consistency.

3 Method

Ideally, in video object segmentation, it is desired to produce pixel-wise predictions y either as *target* $\mathcal{T}$ or *background* $\mathcal{B}$. In a probabilistic sense, we are interested in maximizing the posterior probability for the target given an input embedding X:

$$\begin{aligned} P(Y=\mathcal{T}|X) &= \frac{P(X,Y=\mathcal{T})}{P(X)} = \frac{P(X,Y=\mathcal{T})}{P(X,Y=\mathcal{T})+P(X,Y=\mathcal{B})} \\ &= \frac{1}{1+\dfrac{P(X,Y=\mathcal{B})}{P(X,Y=\mathcal{T})}} = \frac{1}{1+\dfrac{P(X|Y=\mathcal{B})P(Y=\mathcal{B})}{P(X|Y=\mathcal{T})P(Y=\mathcal{T})}}, \end{aligned} \tag{1}$$

The ratio in the denominator determines the posterior probability. If a pixel belongs to the target, the ratio becomes small and the posterior probability tends to 1. Contrarily, if a pixel belongs to the background and is quite distinct from the target, the ratio becomes large and the posterior probability goes to 0. However, if the target prior is large, and the background and target likelihoods are similar because X contains features of a distractor, the posterior can easily be larger than 0.5 and produce false positives. We propose splitting the non-target into two classes, background $\mathcal{B}$ and distractor $\mathcal{D}$:

$$\begin{aligned} P(X,Y\neq\mathcal{T}) &= P(X,Y=\mathcal{B})+P(X,Y=\mathcal{D}) \\ &= P(X|Y=\mathcal{B})P(Y=\mathcal{B})+P(X|Y=\mathcal{D})P(Y=\mathcal{D}). \end{aligned} \tag{2}$$

Consequently, the ratio in (1) will have two terms in the numerator as denoted by (2). This modification limits the occurrences of false positives as it models ambiguous pixels from the first frame and propagates them. As an example, if the likelihood of a certain pixel is similar between the target and the distractor at an intermediate frame, the propagated prior from previous frames will cause the ratio to be large, and the probability to drop. In the following sections, we will describe how to modify an existing baseline to be distractor-aware.

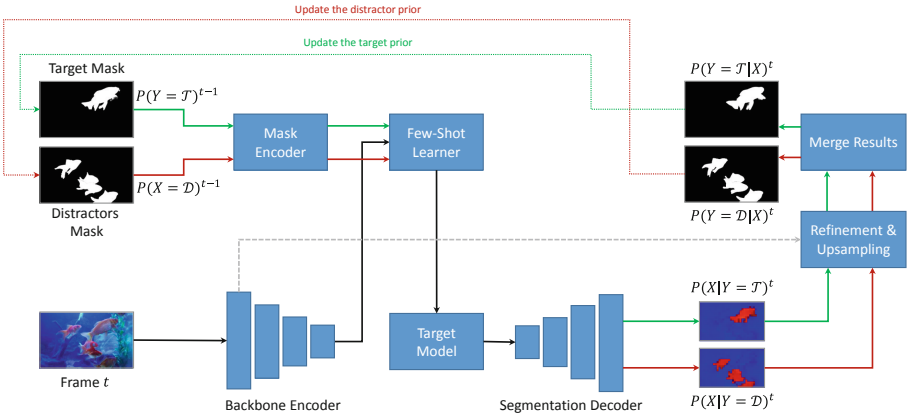

Fig. 2. An overview of our distractor-aware approach. We extend the baseline LWL to incorporate information about distractors throughout its mask encoder, target model, and segmentation decoder stages, and replace its upsampler with a joint refinement and upsampling approach, acting as local regularization.

3.1 Baseline Approach

We base our approach on the recently published method learning-what-to-learn [3] (LWL). In the LWL baseline, a target segmentation mask is encoded by a mask encoder into an multi-channel embedding at one sixteenth of the original image resolution. At the same time, a standard backbone encoder network (ResNet-50) is used to extract features from the whole image (see Fig. 2). At the first frame, both the mask embeddings and the image features are utilized as training data for a target model $T_\theta(x) = x * \theta$ that is trained with a few-shot learner. In subsequent video frames, this target model generates new multi-channel embeddings from the corresponding image features. A decoder network finally recovers segmentation masks at full resolution from these embeddings. Eventually, the newly predicted masks and deep features are added to the training data, so that the target model can be tuned, adapting it to the changing target appearance. Note that multiple objects are tracked independently and individual target models do not interact with each other. In other words, the target model is not aware of any objects in the scene since everything other than the target is treated as background.

3.2 Introducing Distractor-Awareness

As describe in the related work section, there are several ways to detect distractors. Here, we consider other objects in the scene as potential distractors. More specifically, under the assumption that segmentation masks are binary, given a target masks $\mathbf{1}_{t_i}$, we generate a distractor mask as:

$$\mathbf{1}_{d_i} = \underset{j \neq i}{\cup} \mathbf{1}_{t_j} \; \forall j \in \mathbb{I}, \tag{3}$$

where $\mathbb{I} = \{1...N\}$ is the set of target IDs in the current video sequence. Both the target and the distractor masks are set to the ground truth in the first frame, and updated with the previous predictions in later frames. To accommodate the distractor, we add a second input channel to the mask encoder and a second output channel to the segmentation decoder (see Fig. 2).

As the segmentation of frames progresses, we merge the decoded and upsampled target masks to form new distractors. However, in this case, the propagated masks are no longer binary and (3) needs to be replaced. For this, we develop a per-pixel winner-take-all (WTA) function.

Let $p_{t_i}(x) \in \mathbb{R}^{H \times W}$ be the target segment probability map, i.e. the network decoder output after a sigmoid activation, of the target with index i. Now let

$$p_{\max}(x) = \sup_j p_{t_j}(x) \; \forall j \in \mathbb{I}, \tag{4}$$

$$p_{\min}(x) = \inf_j p_{t_j}(x) \; \forall j \in \mathbb{I}, \tag{5}$$

merging the highest and lowest probabilities (per pixel), into $p_{\max}$ and $p_{\min}$.

Now let, $L(x) \in (\mathbb{I} \cup \{0\})^{H \times W}$ be the map of merged segmentation labels (after softmax-aggregation, as introduced in [8]), with 0 the background label.

Also let

$$\mathbf{1}_f(x) = \begin{cases} 1 & \text{if } \; L(x) > 0, \\ 0 & \text{if } \; L(x) = 0. \end{cases} \tag{6}$$

indicate regions with any foreground pixel, and

$$\mathbf{1}_{d_i}(x) = \begin{cases} \mathbf{1}_f(x) & \text{if } L(x) \neq i, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

indicate distractors of target i. A new probability map of distractor i is generated as

$$p_{d_i}(x) = \mathbf{1}_{d_i}(x) p_{\max}(x) + (1 - \mathbf{1}_f(x)) p_{\min}(x). \tag{8}$$

In less formal terms, we let the the most certain predictions of target and background "win" in every pixel. p_{d_i} takes information from every p_{t_j} except that of target i.

In the ablation study, we compare the performance of this function to that of feeding back the decoded distractor to the few-shot learner without modification.

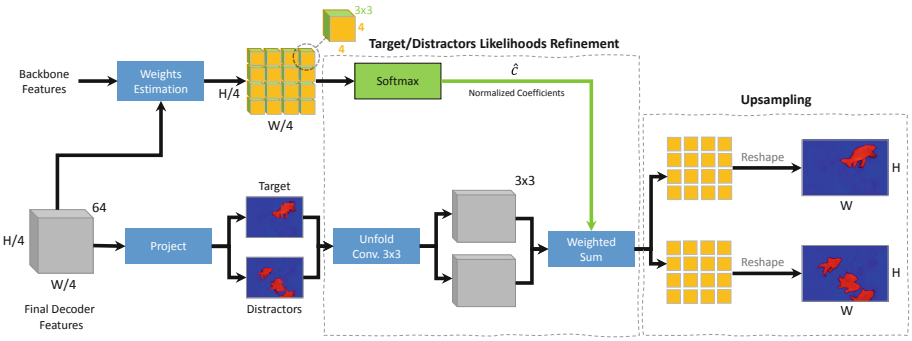

Fig. 3. An overview of the refinement and upsampling module. The final decoder features are first projected into target and distractor logit maps. A weights estimation network then predicts a 5D tensor of weights in 3×3 windows around each pixel in the logit maps (shown in green), as well as interpolation data (shown in yellow). The weights are mapped to normalized probability vectors with the softmax function and then used refine the target and the distractor logits with weighted summation. The refined logits are finally upsampled to full resolution with the interpolation data. (Color figure online)

3.3 Joint Refinement and Upsampling for Local Consistency

The majority of existing VOS methods adopt a binary classification scheme between the foreground and the background. It is typically challenging to produce accurate predictions at uncertain regions such as target boundaries due to multi-modality along edges. This problem is aggravated when we introduce a new class for distractors, as the decision is now made between three classes instead of two. We tackle this problem by introducing two modifications to the baseline encoder and decoder, respectively.

First, we provide high-resolution feature maps from the backbone when training the target model in the few-shot learner. Unlike the baseline, we employ backbone features with a 1/8th of the full resolution instead of 1/16th. These higher-resolution features provide finer details to the target model, especially along edges, to help resolving ambiguities in uncertain regions. Second, we replace the baseline upsampler on the decoder output, with a joint refinement and upsampling unit based on the convex combination module of [13]. This is originally employed to upsample flow fields, while we adopt it to produce consistent selections between the target and distractors in logit space.

The proposed joint refinement/upsampling approach is illustrated in Fig. 3. First, the output from the decoder is projected using two convolution layers to two channels resembling the likelihoods of the target and the distractor. Then, the likelihoods are unfolded into 3×3 patches around each pixel for computational efficiency. In parallel, we employ a weights estimation network that jointly perform likelihoods refinement, and mask upsampling. The former allows modifying the likelihoods of the target and the distractor based on their neighbors to enforce some local consistency, while the latter produces pixel values needed for upsampling.

For the refinement, the weights estimation network predicts a local coefficients vector $\mathbf{c}_x$ for a 3×3 window centered around each pixel $x \in X$:

$$\mathbf{c}_x = [c_{-4}, \ldots, c_{-1}c_{-1}, c_0, c_1, \ldots, c_4], \tag{9}$$

where c_0 is the coefficient on top of x. However, these coefficients are not normalized, and to preserve likelihood ratios we first map $\mathbf{c}_x$ to a normalized vector $\hat{\mathbf{c}}_x$ using the softmax function. To modify the likelihoods, we subsequently apply the coefficients to the likelihoods using a weighted sum resembling convolution:

$$P(X|Y)' * \hat{\mathbf{c}}_x[x] = \sum_{m \in [-4:4]} P(X|Y)[x-m]\, \hat{\mathbf{c}}_x[m]. \tag{10}$$

For the upsampling, we need to predict pixel values for 4×4 patches around each pixel for a scaling of 4. Those values are also predicted using the weights-estimation network and are needed to produce the full resolution prediction. This combined refinement-and-upsampling feature volume takes the shape of a local 3D tensor $\mathbf{A}_x \in \mathbb{R}^{4\times4\times9}$ for each pixel $x \in X$.

4 Experiments

In this section, we provide implementation details including the training procedure and loss function. We compare against state-of-the-art approaches on DAVIS 2017 [10], and YouTube-VOS 2018 [14]. In addition, we provide an extensive ablation analysis.

4.1 Implementation Details

Training Procedure. Similar to the baseline, our training setup imitates the segmentation processing during inference. Each training sample is a mini-video of four frames with one main target object to segment. The few-shot learner is provided the first frame to train a target model. The target model and the decoder then predicts segmentation masks from the subsequent three frames. The predictions in turn, are both used to update the target model and compute the network training loss.

To extend this procedure for distractors, we replicate the segmentation process for all other objects that are present in the ground-truth label map. These additional segments are then merged and provided as distractors to the main target. However, we set the network into evaluation mode when processing these targets to conserve memory.

We employ the same data augmentation and first two training phases as the baseline [3]. The proposed refinement and upsampling module is trained in a third phase for 6,000 iterations on DAVIS 2017, with learning rate 10^{-4}, all other weights frozen.

Loss Function. LWL was trained with the Lovász-softmax loss [1], a differentiable relaxation of the Jaccard similarity measure, and we continue to do so here. However, in the SOTA experiments, we did initially not see any improvements with our method on YouTube-VOS. We hypothesize that this is due to large size-differences between

objects in the same sequence in YouTube-VOS. To counter this, we split the loss into two terms, or a balanced loss:

$$L = \text{Lovasz}(T) + w(\hat{T}, \hat{D})\text{Lovasz}(D) \tag{11}$$

where T and D are the output batch from the decoder network, separated into target and distractor channels. $w(\hat{T}, \hat{D})$ reduces the influence of large objects as a function of the number of ground-truth target pixels $|\hat{T}|$ and ground-truth distractor pixels $|\hat{D}|$ across the training batch:

$$w(\hat{T}, \hat{D}) = \min(|\hat{T}|/|\hat{D}|, 1.0) \tag{12}$$

In other words, when the distractors jointly occupy a larger region than the target, their influence on the loss is reduced.

Relaxed Distractor Loss. Some sequences have only one target. This implies that no distractors exist, since we derive them from every other target. To not unnecessarily over-constrain the training, we partially disable the computation of the loss in training samples in these cases. Specifically, we require the distractor output to be zero in the area under the target, but allow it to take any value elsewhere.

We also train a variant with a "hard" loss, requiring that no distractor is output when no distractor is given, and compare them in the experiments.

4.2 State-of-the-Art Comparisons

Table 1. Results on DAVIS 2017 and YouTube-VOS 2018, comparing our method to the LWL baseline and the state-of-the-art. The LWL scores are from our own run with the official code.

Method	DAVIS'17 val			DAVIS'17 test-dev			YouTube-VOS'18 valid				
Name	$\mathcal{G}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{J}_s$	$\mathcal{J}_u$	$\mathcal{F}_s$	$\mathcal{F}_u$
CFBI [15]	81.9	79.1	84.6	74.8	71.1	78.5	81.4	81.1	75.3	85.8	83.4
CFBI-MS [15]	**83.3**	**80.5**	**86.0**	**77.5**	73.8	**81.1**	**82.7**	**82.2**	**76.9**	**86.8**	**85.0**
KMN [12]	82.8	80.0	85.6	77.2	**74.1**	80.3	81.4	81.4	75.3	85.6	83.3
STM [9]	81.8	79.2	84.3	72.2	69.3	75.2	79.4	79.7	72.8	84.2	80.9
GMVOS [7]	82.8	80.2	85.2	–	–	–	80.2	80.7	74.0	85.1	80.9
LWL [3]	80.1	77.4	82.8	70.8	68.0	73.7	80.7	79.5	75.6	84.0	83.5
Ours	**83.7**	**81.1**	**86.2**	74.1	71.2	77.1	79.8	79.8	74.0	84.1	81.3
Ours (Balanced loss)	82.6	80.8	85.3	**75.4**	**72.5**	**78.2**	**81.5**	**80.4**	**76.0**	**85.1**	**84.5**

We compare against the most recent state-of-the-art approaches for video object segmentation: CFBI [15], STM [9], KMN [12], GMVOS [7], and the baseline LWL [3]. CFBI-MS is a multi-scale variant operating on three different scales. Our method in contrast, is single scale and thus more comparable to the plain CFBI. Running the official LWL implementation we noticed that our results do not coincide exactly with those

reported in [3]. Therefore, we use these newly obtained scores to accurately compare the baseline to our method.

Table 1 summarizes the quantiative results on DAVIS 2017 and the YouTube-VOS 2018 validation split. On DAVIS val, our proposed approach without the balanced loss sets a new state-of-the-art on DAVIS val, while improving over the baseline on test-dev by 3.6% points. With the balanced loss however, test-dev surprisingly improves 4.6% points, while reducing the gain on val to 2.5% points. On the YouTube-VOS 2018 validation split, our approach improves the baseline by 0.8% points when trained *with* the balanced loss, while scoring similarly to other state-of-the-art approaches.

Some qualitative results are found in the supplement.

4.3 Ablation Study

Table 2. Ablation study on DAVIS 2017. See Sect. 4.3 for details. Interesting scores are printed in bold type.

Parameters				DAVIS'17 val			DAVIS'17 test-dev		
L2	D	U	B	$\mathcal{G}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{J}$	$\mathcal{F}$
				80.1	77.4	82.8	70.8	68.0	73.7
	✓			80.7	77.9	83.4	71.0	67.7	74.2
	✓	✓		80.7	78.2	83.3	71.7	68.7	74.8
✓				81.9	78.9	84.9	72.7	69.5	75.9
✓		✓		81.6	78.9	84.3	71.6	68.5	74.7
✓	✓			81.3	78.8	83.9	73.2	70.0	76.4
✓	✓	✓		**83.7**	81.1	86.2	**74.1**	71.2	77.1
Hard loss				**80.2**	77.5	82.8	**74.9**	72.4	77.4
No Distractors				81.1	78.7	83.4	71.2	68.0	74.4
No WTA				83.0	80.5	85.5	71.8	68.4	75.2
✓	✓	✓	✓	**82.6**	80.8	85.3	**75.4**	72.5	78.2

In this section, we analyze the impact of different components of our method on the overall performance. We use the DAVIS 2017 dataset [10] for this purpose, where we include both *val* and *test-dev* splits for diversity. The results are shown in Table 2. Since LWL [3] is our baseline method, we reevaluate their official pretrained model. As mentioned above, there is a discrepancy between our obtained scores and the published ones. This could be due to several factors, e.g. hardware/software differences and driver variations. However, we use the scores that we obtained for a valid comparison.

We first enable or disable various combinations of these components: the high-resolution features from ResNet block/layer 2 (L2), the distractor-awareness (D), the refinement/upsampling module (U). Enabling D alone causes a slight improvement. Adding U to this improves the test-dev split further. We believe that the lack of high-resolution features prevents further gains. Similarly, introducing L2 alone, causes a

marginal improvement on both splits and enabling U at the same time, shows no improvement on the valdiation set, and hurts the performance on the test-set. This can be explained by our argument in Sect. 3.3: When the decision is made between two classes (target and background), there is no need for the upsampler to resolve ambiguities. As to why this combinatios hurts test-dev performance, is unclear. However, enabling all three (L2+D+U) at the same time, yields significantly better results over the baseline.

With L2, D and U included, we then separately replace the relaxed distractor loss with the "hard" loss (Hard loss). The results then drops back to the baseline for DAVIS validation split, but oddly *improves* on the test-dev split.

We also test a pair of inference-time modifications. First, we zero out the distractor masks so that the few shot learner do not use them. Second, we replace the WTA distractor-merging function with a simple pass-through; Distractors are directly routed from the decoder output back to the few-shot learner, bypassing the merging step. As one would expect, the results drop in both cases, proving that WTA is important. Surprisingly though, disabling the WTA function hits DAVIS test-dev split much harder than the validation split.

Finally, we add the balanced loss (B) to the L2+D+U variant. This clearly damages the DAVIS validation scores but improves the test-dev results. However, at the same time it also improves the YouTube-VOS results in Table 1.

These results suggest that the method works, but also reveal differences between the DAVIS dataset splits that we cannot explain at this time.

4.4 A Note on WTA vs. Softmax Aggregation

The softmax aggregation introduced in [8], and used to merge estimations of multiple targets, was introduced as a superior alternative to a winner-take-all approach. In our case however, we found that to merge distractors, WTA performs better. We hypothesize that lacking a dedicated background channel is beneficial to the refinement and upsampling module, as the decoder may output low activations on both target and distractors when the classification is uncertain.

4.5 Emergent Distractors

An essential question is how our framework would behave when there is only one labeled object in the scene. Interestingly, the model learns to identify ambiguous regions. Figure 4 shows an example where our approach learns to identify the camel in the background without any explicit supervision (if columns 2+3 from the left). We attribute this to the relaxed distractor loss described in Sect. 4.1. To test this, we modify the loss to be "hard" (H). As suspected, this suppresses the behaviour greatly, while increasing the decoders' certainty in both the target and background (columns 4+5).

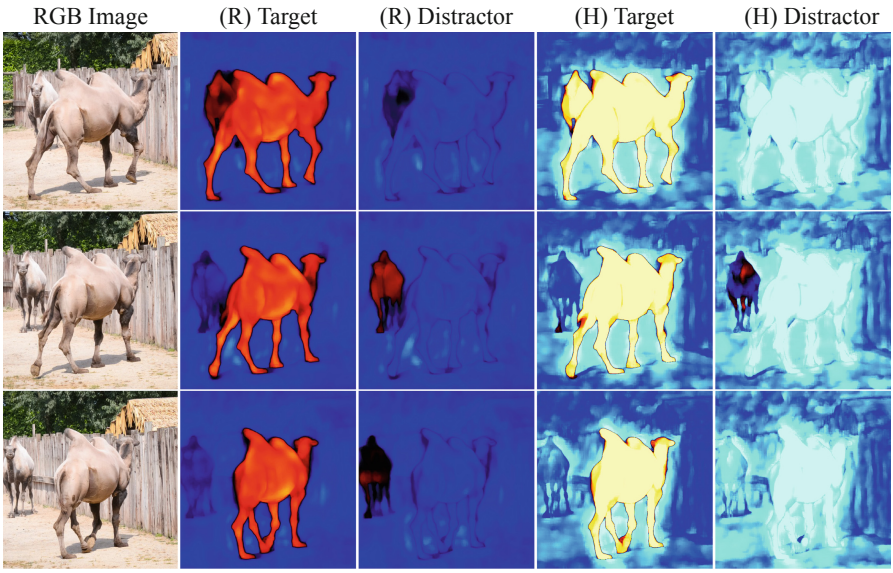

Fig. 4. Frames 40, 50 60 (from top to bottom) from the 'camel' sequence, with target/distractor score maps. Our model can learn to identify distractors in case they were not explicitly provided, provided it is trained with the relaxed distractor loss. Red/yellow indicates positive log-likelihoods, blue/cyan negative. The colors represent the same value range in all columns. (R) and (H) indicate results from models trained with the relaxed and hard distractor losses, respectively. (Color figure online)

5 Conclusion

We have proposed a distractor-aware, discriminative video object segmentation approach. In contrast to existing methods, our proposed method encodes distractors into a separate class, to exploit information about other objects in the scene that are likely to be confused with the target. Moreover, we propose the use of joint refinement and upsampling to regularize the likelihoods for highly uncertain regions with their neighborhoods. We demonstrated the effectiveness of our approach by modifying an existing state-of-the-art approach to be distractor-aware. Our modification sets a new state-of-the-art on the DAVIS 2017 *val* dataset, while improving over the baseline with a remarkable margin on the DAVIS 2017 *test-dev* dataset. These results clearly indicate the efficacy of explicitly modelling distractors when solving video object segmentation.

Acknowledgements. This project was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the Excellence Center at Linköping-Lund in Information Technology (ELLIIT), the Swedish Research Council grant no. 2018-04673, and the Swedish Foundation for Strategic Research (SSF) project Symbicloud.

References

1. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
2. Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Know your surroundings: exploiting scene information for object tracking. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 205–221. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_13
3. Bhat, G., et al.: Learning what to learn for video object segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 777–794. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_46
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005) (2005)
5. Jin, S., et al.: Unsupervised hard example mining from videos for improved object detection. In: The European Conference on Computer Vision (ECCV), September 2018
6. Johnander, J., Danelljan, M., Brissman, E., Khan, F.S., Felsberg, M.: A generative appearance model for end-to-end video object segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
7. Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Van Gool, L.: Video object segmentation with episodic graph memory networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 661–679. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8_39
8. Oh, S.W., Lee, J.Y., Sunkavalli, K., Kim, S.J.: Fast video object segmentation by reference-guided mask propagation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
9. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
10. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 Davis challenge on video object segmentation. arXiv:1704.00675 (2017)
11. Robinson, A., Lawin, F.J., Danelljan, M., Khan, F.S., Felsberg, M.: Learning fast and robust target models for video object segmentation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020
12. Seong, H., Hyun, J., Kim, E.: Kernelized memory network for video object segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12367, pp. 629–645. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58542-6_38
13. Teed, Z., Deng, J.: RAFT: recurrent all-pairs field transforms for optical flow. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 402–419. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_24
14. Xu, N., et al.: YouTube-VOS: a large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018)
15. Yang, Z., Wei, Y., Yang, Y.: Collaborative video object segmentation by foreground-background integration. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12350, pp. 332–348. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_20
16. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware Siamese networks for visual object tracking. In: The European Conference on Computer Vision (ECCV), September 2018