Collaborative Video Object Segmentation by Foreground-Background Integration

Zongxin Yang[1,2], Yunchao Wei[2], and Yi Yang[2(✉)]

[1] Baidu Research, Beijing, China
`zongxin.yang@student.uts.edu.au`
[2] ReLER, Centre for Artificial Intelligence, University of Technology Sydney,
Ultimo, SYD, Australia
`{yunchao.wei,yi.yang}@uts.edu.au`

Abstract. This paper investigates the principles of embedding learning to tackle the challenging semi-supervised video object segmentation. Different from previous practices that only explore the embedding learning using pixels from foreground object (s), we consider background should be equally treated and thus propose Collaborative video object segmentation by Foreground-Background Integration (CFBI) approach. Our CFBI implicitly imposes the feature embedding from the target foreground object and its corresponding background to be contrastive, promoting the segmentation results accordingly. With the feature embedding from both foreground and background, our CFBI performs the matching process between the reference and the predicted sequence from both pixel and instance levels, making the CFBI be robust to various object scales. We conduct extensive experiments on three popular benchmarks, *i.e.*, DAVIS 2016, DAVIS 2017, and YouTube-VOS. Our CFBI achieves the performance ($\mathcal{J}\&\mathcal{F}$) of 89.4%, 81.9%, and 81.4%, respectively, outperforming all the other state-of-the-art methods. Code: https://github.com/z-x-yang/CFBI.

Keywords: Video Object Segmentation · Metric learning

1 Introduction

Video Object Segmentation (VOS) is a fundamental task in computer vision with many potential applications, including augmented reality [25] and self-driving cars [44]. In this paper, we focus on semi-supervised VOS, which targets on segmenting a particular object across the entire video sequence based on the object mask given at the first frame. The development of semi-supervised VOS can benefit many related tasks, such as video instance segmentation [13,41] and interactive video object segmentation [21,24,26].

This work was done when Zongxin Yang interned at Baidu Research.

Early VOS works [2,23,35] rely on fine-tuning with the first frame in evaluation, which heavily slows down the inference speed. Recent works (*e.g.*, [27,34,42]) aim to avoid fine-tuning and achieve better run-time. In these works, STMVOS [27] introduces memory networks to learn to read sequence information and outperforms all the fine-tuning based methods. However, STMVOS relies on simulating extensive frame sequences using large image datasets [7,12,15,22,32] for training. The simulated data significantly boosts the performance of STMVOS but makes the training procedure elaborate. Without simulated data, FEELVOS [34] adopts a semantic pixel-wise embedding together with a global (between the first and current frames) and a local (between the previous and current frames) matching mechanism to guide the prediction. The matching mechanism is simple and fast, but the performance is not comparable with STMVOS.

Even though the efforts mentioned above have made significant progress, current state-of-the-art works pay little attention to the feature embedding of background region in videos and only focus on exploring robust matching strategies for the foreground object (s). Intuitively, it is easy to extract the foreground region from a video when precisely removing all the background. Moreover, modern video scenes commonly focus on many similar objects, such as the cars in car racing, the people in a conference, and the animals on a farm. For these cases,

Fig. 1. CI means collaborative integration. There are two foreground sheep (pink and blue). In the top line, the contempt of background matching leads to a confusion of sheep's prediction. In the bottom line, we relieve the confusion problem by introducing background matching (dot-line arrow). (Color figure online)

the contempt of integrating foreground and background embeddings traps VOS in an unexpected background confusion problem. As shown in Fig. 1, if we focus on only the foreground matching like FEELVOS, a similar and same kind of object (sheep here) in the background is easy to confuse the prediction of the foreground object. Such an observation motivates us that the background should be equally treated compared with the foreground so that better feature embedding can be learned to relieve the background confusion and promote the accuracy of VOS.

We propose a novel framework for Collaborative video object segmentation by Foreground-Background Integration (CFBI) based on the above motivation. Different from the above methods, we not only extract the embedding and do match for the foreground target in the reference frame, but also for the background region to relieve the background confusion. Besides, our framework extracts two types of embedding (*i.e.*, pixel-level, and instance-level embedding) for each video frame to cover different scales of features. Like FEELVOS, we employ pixel-level embedding to match all the objects' details with the same global & local mechanism. However, the pixel-level matching is not sufficient and robust to match those objects with larger scales and may bring unexpected noises due to
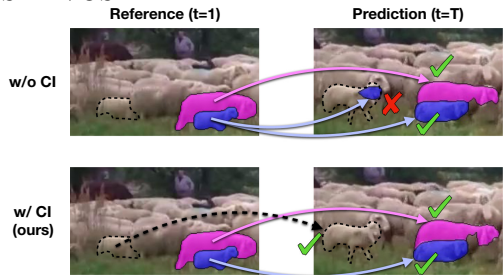
the pixel-wise diversity. Thus we introduce instance-level embedding to help the segmentation of large-scale objects by using attention mechanisms. Moreover, we propose a collaborative ensembler to aggregate the foreground & background and pixel-level & instance-level information and learn the collaborative relationship among them implicitly. For better convergence, we take a balanced random-crop scheme in training to avoid learned attributes being biased to the background attributes. All these proposed strategies can significantly improve the quality of the learned collaborative embeddings for conducting VOS while keeping the network simple yet effective simultaneously.

We perform extensive experiments on DAVIS [30,31], and YouTube-VOS [40] to validate the effectiveness of the proposed CFBI approach. Without any bells and whistles (such as the use of simulated data, fine-tuning or post-processing), CFBI outperforms all other state-of-the-art methods on the validation splits of DAVIS 2016 (ours, $\mathcal{J}\&\mathcal{F}$ **89**.**4**%), DAVIS 2017 (**81**.**9**%) and YouTube-VOS (**81**.**4**%) while keeping a competitive single-object inference speed of about 5 FPS. By additionally applying multi-scale & flip augmentation at the testing stage, the accuracy can be further boosted to **90**.**1**%, **83**.**3**% and **82**.**7**%, respectively. We hope our simple yet effective CFBI will serve as a solid baseline and help ease VOS's future research.

2   Related Work

Semi-supervised Video Object Segmentation. Many previous methods for semi-supervised VOS rely on fine-tuning at test time. Among them, OSVOS [2] and MoNet [39] fine-tune the network on the first-frame ground-truth at test time. OnAVOS [35] extends the first-frame fine-tuning by an online adaptation mechanism, *i.e.*, online fine-tuning. MaskTrack [29] uses optical flow to propagate the segmentation mask from one frame to the next. PReMVOS [23] combines four different neural networks (including an optical flow network [11]) using extensive fine-tuning and a merging algorithm. Despite achieving promising results, all these methods are seriously slowed down by fine-tuning during inference.

Some other recent works (*e.g.*, [6,42]) aim to avoid fine-tuning and achieve a better run-time. OSMN [42] employs two networks to extract the instance-level information and make segmentation predictions, respectively. PML [5] learns a pixel-wise embedding with the nearest neighbor classifier. Similar to PML, VideoMatch [18] uses a soft matching layer that maps the pixels of the current frame to the first frame in a learned embedding space. Following PML and Video-Match, FEELVOS [34] extends the pixel-level matching mechanism by additionally matching between the current frame and the previous frame. Compared to the methods with fine-tuning, FEELVOS achieves a much higher speed, but there is still a gap inaccuracy. Like FEELVOS, RGMP [38] and STMVOS [27] does not require any fine-tuning. STMVOS, which leverages a memory network to store and read the information from past frames, outperforms all the previous methods. However, STMVOS relies on an elaborate training procedure using extensive simulated data generated from multiple datasets. Moreover, the above methods do not focus on background matching.
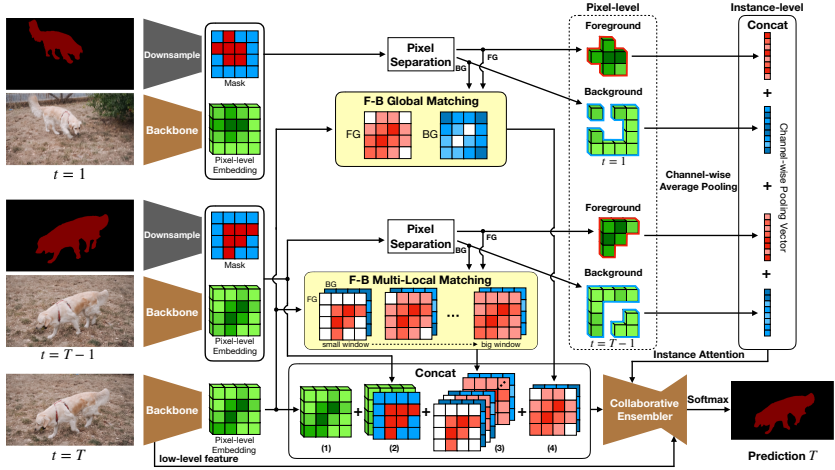
Fig. 2. An **overview** of CFBI. F-G denotes Foreground-Background. We use red and blue to indicate foreground and background separately. The deeper the red or blue color, the higher the confidence. Given the first frame ($t = 1$), previous frame ($t = T - 1$), and current frame ($t = T$), we firstly extract their pixel-wise embedding by using a backbone network. Second, we separate the first and previous frame embeddings into the foreground and background pixels based on their masks. After that, we use F-G pixel-level matching and instance-level attention to guide our collaborative ensembler network to generate a prediction. (Color figure online)

Our CFBI utilizes both the pixel-level and instance-level embeddings to guide prediction. Furthermore, we propose a collaborative integration method by additionally learning background embedding.

Attention Mechanisms. Recent works introduce the attention mechanism into convolutional networks (*e.g.*, [9,14]). Following them, SE-Nets [17] introduced a lightweight gating mechanism that focuses on enhancing the representational power of the convolutional network by modeling channel attention. Inspired by SE-Nets, CFBI uses an instance-level average pooling method to embed collaborative instance information from pixel-level embeddings. After that, we conduct a channel-wise attention mechanism to help guide prediction. Compared to OSMN, which employs an additional convolutional network to extract instance-level embedding, our instance-level attention method is more efficient and lightweight.

3 Method

Overview. Learning foreground feature embedding has been well explored by previous practices (*e.g.*, [34,42]). OSMN proposed to conduct an instance-level matching, but such a matching scheme fails to consider the feature diversity among the details of the target's appearance and results in coarse predictions. PML and FEELVOS alternatively adopt the pixel-level matching by matching

each pixel of the target, which effectively takes the feature diversity into account and achieves promising performance. Nevertheless, performing pixel-level matching may bring unexpected noises in the case of some pixels from the background are with a similar appearance to the ones from the foreground (Fig. 1).

To overcome the problems raised by the above methods and promote the foreground objects from the background, we present Collaborative video object segmentation by Foreground-Background Integration (CFBI), as shown in Fig. 2. We use red and blue to indicate foreground and background separately. First, beyond learning feature embedding from foreground pixels, our CFBI also considers embedding learning from background pixels for collaboration. Such a learning scheme will encourage the feature embedding from the target object and its corresponding background to be contrastive, promoting the segmentation results accordingly. Second, we further conduct the embedding matching from both pixel-level and instance-level with the collaboration of pixels from the foreground and background. For the pixel-level matching, we improve the robustness of the local matching under various object moving rates. For the instance-level matching, we design an instance-level attention mechanism to augment the pixel-level matching efficiently. Moreover, to implicitly aggregate the learned foreground & background and pixel-level & instance-level information, we employ a collaborative ensembler to construct large receptive fields and make precise predictions.

3.1   Collaborative Pixel-Level Matching

For the pixel-level matching, we adopt a global and local matching mechanism similar to FEELVOS for introducing the guided information from the first and previous frames, respectively. Unlike previous methods [5,34], we additionally incorporate background information and apply multiple windows in the local matching, which is shown in the middle of Fig. 2.

For incorporating background information, we firstly redesign the pixel distance of [34] to further distinguish the foreground and background. Let B_t and F_t denote the pixel sets of background and all the foreground objects of frame t, respectively. We define a new distance between pixel p of the current frame T and pixel q of frame t in terms of their corresponding embedding, e_p and e_q, by

$$D_t(p,q) = \begin{cases} 1 - \frac{2}{1+exp(||e_p-e_q||^2+b_B)} & \text{if } q \in B_t \\ 1 - \frac{2}{1+exp(||e_p-e_q||^2+b_F)} & \text{if } q \in F_t \end{cases}, \tag{1}$$

where b_B and b_F are trainable background bias and foreground bias. We introduce these two biases to make our model be able further to learn the difference between foreground distance and background distance.

Foreground-Background Global Matching. Let $\mathcal{P}_t$ denote the set of all pixels (with a stride of 4) at time t and $\mathcal{P}_{t,o} \subseteq \mathcal{P}_t$ is the set of pixels at time t which belongs to the foreground object o. The global foreground matching

between one pixel p of the current frame T and the pixels of the first reference frame (*i.e.*, $t = 1$) is,

$$G_{T,o}(p) = \min_{q \in \mathcal{P}_{1,o}} D_1(p,q). \tag{2}$$

Similarly, let $\overline{\mathcal{P}}_{t,o} = \mathcal{P}_t \backslash \mathcal{P}_{t,o}$ denote the set of relative background pixels of object o at time t, and the global background matching is,

$$\overline{G}_{T,o}(p) = \min_{q \in \overline{\mathcal{P}}_{1,o}} D_1(p,q). \tag{3}$$

Foreground-Background Multi-Local Matching.

In FEELVOS, the local matching is limited in only one fixed extent of neighboring pixels, but the offset of objects across two adjacent frames in VOS is variable, as shown in Fig. 3. Thus, we propose to apply the local matching mechanism on different scales and let the network learn how to select an appropriate local scale, which makes our framework more robust to various moving rates of objects. Notably, we use the intermediate results of the local matching with the largest window to calculate on other windows. Thus, the increase of computational resources of our multi-local matching is negligible.

Formally, let $K = \{k_1, k_2, ..., k_n\}$ denote all the neighborhood sizes and $H(p,k)$ denote the neighborhood set of pixels that are at most k pixels away from p in both x and y directions, our foreground multi-local matching between the current frame T and its previous frame $T-1$ is

[Slow moving rate]

[Fast moving rate]

Fig. 3. The moving rate of objects across two adjacent frames is largely variable for different sequences. Examples are from YouTube-VOS [40].

$$ML_{T,o}(p, K) = \{L_{T,o}(p, k_1), L_{T,o}(p, k_2), ..., L_{T,o}(p, k_n)\}, \tag{4}$$

where

$$L_{T,o}(p,k) = \begin{cases} \min_{q \in \mathcal{P}_{T-1,o}^{p,k}} D_{T-1}(p,q) & \text{if } \mathcal{P}_{T-1,o}^{p,k} \neq \emptyset \\ 1 & \text{otherwise} \end{cases}. \tag{5}$$

Here, $\mathcal{P}_{T-1,o}^{p,k} := \mathcal{P}_{T-1,o} \cap H(p,k)$ denotes the pixels in the local window (or neighborhood). And our background multi-local matching is

$$\overline{ML}_{T,o}(p, K) = \{\overline{L}_{T,o}(p, k_1), \overline{L}_{T,o}(p, k_2), ..., \overline{L}_{T,o}(p, k_n)\}, \tag{6}$$

where

$$\overline{L}_{T,o}(p,k) = \begin{cases} \min_{q \in \overline{\mathcal{P}}_{T-1,o}^{p,k}} D_{T-1}(p,q) & \text{if } \overline{\mathcal{P}}_{T-1,o}^{p,k} \neq \emptyset \\ 1 & \text{otherwise} \end{cases}. \tag{7}$$

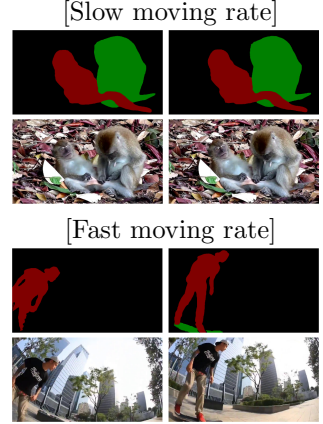Here similarly, $\overline{\mathcal{P}}_{T-1,o}^{p,k} := \overline{\mathcal{P}}_{T-1,o} \cap H(p,k)$.

In addition to the global and multi-local matching maps, we concatenate the pixel-level embedding feature and mask of the previous frame with the current frame feature. FEELVOS demonstrates the effectiveness of concatenating the previous mask. Following this, we empirically find that introducing the previous embedding can further improve the performance ($\mathcal{J}\&\mathcal{F}$) by about 0.5%.

In summary, the output of our collaborative pixel-level matching is a concatenation of (1) the pixel-level embedding of the current frame, (2) the pixel-level embedding and mask of the previous frame, (3) the multi-local matching map and (4) the global matching map, as shown in the bottom box of Fig. 2.

3.2   Collaborative Instance-Level Attention

As shown in the right of Fig. 2, we further design a Collaborative instance-level attention mechanism to guide the segmentation for large-scale objects.

After getting the pixel-level embeddings of the first and previous frames, we separate them into foreground and background pixels (*i.e.*, $\mathcal{P}_{1,o}$, $\overline{\mathcal{P}}_{1,o}$, $\mathcal{P}_{T-1,o}$, and $\overline{\mathcal{P}}_{T-1,o}$) according to their masks. Then, we apply channel-wise average pooling on each group of pixels to generate a total of four instance-level embedding vectors and concatenate these vectors into one collaborative instance-level guidance vector. Thus, the guidance vector contains the information from both the first and previous frames, and both the foreground and background regions.

In order to efficiently utilize the instance-level information, we employ an attention mechanism to adjust our Collaborative Ensembler (CE). We show a detailed illustration in Fig. 4. Inspired by SE-Nets [17], we leverage a fully-connected (FC) layer (we found this setting is better than using two FC layers as adopted by SE-Net) and a non-linear activation function to construct a gate for the input of each Res-Block in the CE. The gate will adjust the scale of the input feature channel-wisely.

Fig. 4. The trainable part of the instance-level attention. C_e denotes the channel dimension of pixel-wise embedding. H, W, C denote the height, width, channel dimension of CE features.

By introducing collaborative instance-level attention, we can leverage a full scale of foreground-background information to guide the prediction further. The information with a large (instance-level) receptive field is useful to relieve local ambiguities [33], which is inevitable with a small (pixel-wise) receptive field.

3.3   Collaborative Ensembler (CE)

In the lower right of Fig. 2, we design a collaborative ensembler for making large receptive fields to aggregate pixel-level and instance-level information and implicitly learn the collaborative relationship between foreground and background.
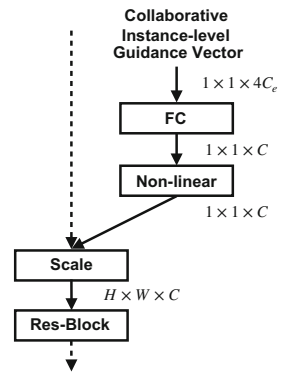
Inspired by ResNets [16] and Deeplabs [3,4], which both have shown significant representational power in image segmentation tasks, our CE uses a downsample-upsample structure, which contains three stages of Res-Blocks [16] and an Atrous Spatial Pyramid Pooling (ASPP) [4] module. The number of Res-Blocks in Stage 1, 2, and 3 are 2, 3, 3 in order. Besides, we employ dilated convolutional layers to improve the receptive fields efficiently. The dilated rates of the 3×3 convolutional layer of Res-Blocks in one stage are separately 1, 2, 4 (or 1, 2 for Stage 1). At the beginning of Stage 2 and Stage 3, the feature maps will be downsampled by the first Res-Block with a stride of 2. After these three stages, we employ an ASPP and a Decoder [4] module to increase the receptive fields further, upsample the scale of feature and fine-tune the prediction collaborated with the low-level backbone features.

4 Implementation Details

For better convergence, we modify the random-crop augmentation and the training method in previous methods [27,34].

Balanced Random-Crop. As shown in Fig. 5, there is an apparent imbalance between the foreground and the background pixel number on VOS datasets. Such an issue usually makes the models easier to be biased to background attributes.

(a) Normal   (b) Balanced

Fig. 5. When using normal random-crop, some red windows contain few or no foreground pixels. For reliving this problem, we propose balanced random-crop.

In order to relieve this problem, we take a balanced random-crop scheme, which crops a sequence of frames (*i.e.*, the first frame, the previous frame, and the current frame) by using a same cropped window and restricts the cropped region of the first frame to contain enough foreground information. The restriction method is simple yet effective. To be specific, the balanced random-crop will decide on whether the randomly cropped frame contains enough pixels from foreground objects or not. If not, the method will continually take the cropping operation until we obtain an expected one.

Sequential Training. In the training stage, FEELVOS predicts only one step in one iteration, and the guidance masks come from the ground-truth data. RGMP and STMVOS uses previous guidance information (mask or feature memory) in training, which is more consistent with the inference stage and performs better. In the evaluation stage, the previous guidance masks are always generated by the network in the previous inference steps.

Following RGMP, we train the network using a sequence of consecutive frames in each SGD iteration. In each iteration, we randomly sample a batch
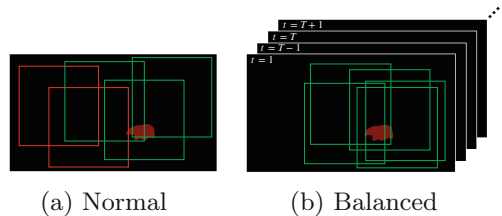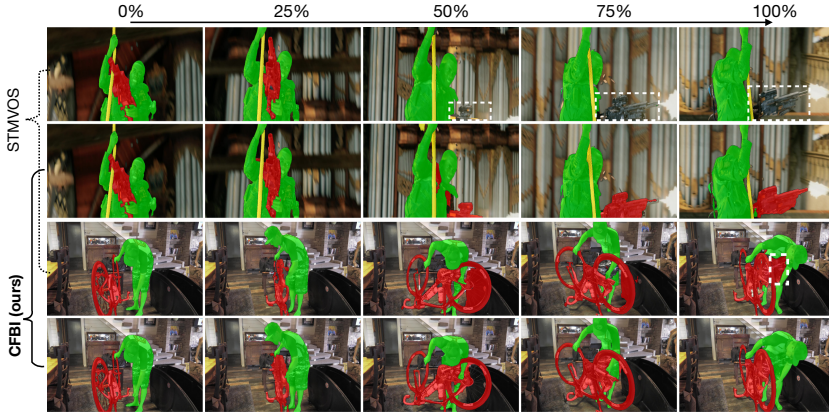
Fig. 6. Qualitative comparison with STMVOS on DAVIS 2017. In the first video, STMVOS fails in tracking the gun after occlusion and blur. In the second video, STMVOS is easier to partly confuse with bicycle and person.

of video sequences. For each video sequence, we randomly sample a frame as the reference frame and a continuous $N + 1$ frames as the previous frame and current frame sequence with N frames. When predicting the first frame, we use the ground-truth of the previous frame as the previous mask. When predicting the following frames, we use the latest prediction as the previous mask.

Training Details. Following FEELVOS, we use the DeepLabv3+ [4] architecture as the backbone for our network. However, our backbone is based on the dilated Resnet-101 [4] instead of Xception-65 [8] for saving computational resources. We apply batch normalization (BN) [19] in our backbone and pre-train it on ImageNet [10] and COCO [22]. The backbone is followed by one depth-wise separable convolution for extracting pixel-wise embedding with a stride of 4.

We initialize b_B and b_F to 0. For the multi-local matching, we further downsample the embedding feature to a half size using bi-linear interpolation for saving GPU memory. Besides, the window sizes in our setting are $K = \{2, 4, 6, 8, 10, 12\}$. For the collaborative ensembler, we apply group normalization (GN) [37] and gated channel transformation [43] to improving training stability and performance when using a small batch size. For sequential training, the current sequence's length is $N = 3$, which makes a better balance between computational resources and network performance.

We use the DAVIS 2017 [31] training set (60 videos) and the YouTube-VOS [40] training set (3471 videos) as the training data. We downsample all the videos to 480P resolution, which is same as the default setting in DAVIS. We adopt SGD with a momentum of 0.9 and apply a bootstrapped cross-entropy loss, which only considers the 15% hardest pixels. During the training stage, we freeze the parameters of BN in the backbone. For the experiments on YouTube-VOS, we use a learning rate of 0.01 for $100,000$ steps with a batch size of 4 videos (*i.e.*, 20 frames in total) per GPU using 2 Tesla V100 GPUs. The training time on YouTube-VOS is about 5 d. For DAVIS, we use a learning rate of 0.006 for $50,000$ steps with a batch size

of 3 videos (*i.e.*, 15 frames in total) per GPU using 2 GPUs. We apply flipping, scaling, and balanced random-crop as data augmentations. The cropped window size is 465×465. For the multi-scale testing, we apply the scales of $\{1.0, 1.15, 1.3, 1.5\}$ and $\{2.0, 2.15, 2.3\}$ for YouTube-VOS and DAVIS, respectively. CFBI achieves similar results in PyTorch [28] and PaddlePaddle [1].

5 Experiments

Following the previous state-of-the-art method [27], we evaluate our method on YouTube-VOS [40], DAVIS 2016 [30] and DAVIS 2017 [31]. For the evaluation on YouTube-VOS, we train our model on the YouTube-VOS training set [40] (3471 videos). For DAVIS, we train our model on the DAVIS-2017 training set [31] (60 videos). Both DAVIS 2016 and 2017 are evaluated using an identical model trained on DAVIS 2017 for a fair comparison with the previous works [27,34]. Furthermore, we provide DAVIS results using both DAVIS 2017 and YouTube-VOS for training following some latest works [27,34].

The evaluation metric is the $\mathcal{J}$ score, calculated as the average IoU between the prediction and the ground truth mask, and the $\mathcal{F}$ score, calculated as an average

Table 1. The quantitative evaluation on YouTube-VOS [40]. F, S, and * separately denote fine-tuning at test time, using simulated data in the training process and performing model ensemble in evaluation. CFBI^{MS} denotes using a multi-scale and flip strategy in evaluation.

Methods	F	S	Avg	Seen		Unseen	
				$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
Validation 2018 split							
AG [20]			66.1	67.8	-	60.8	-
PReM [23]	✓		66.9	71.4	75.9	56.5	63.7
BoLT [36]	✓		71.1	71.6	-	64.3	-
STM$^-$ [27]			68.2	-	-	-	-
STM [27]		✓	79.4	79.7	84.2	72.8	80.9
CFBI			**81.4**	**81.1**	**85.8**	**75.3**	**83.4**
CFBI^{MS}			**82.7**	**82.2**	**86.8**	**76.9**	**85.0**
Testing 2019 split							
MST* [45]		✓	81.7	80.0	83.3	**77.9**	85.5
EMN* [46]		✓	81.8	**80.7**	**84.7**	77.3	84.7
CFBI			81.5	79.6	84.0	77.3	85.3
CFBI^{MS}			**82.2**	80.4	**84.7**	**77.9**	**85.7**

boundary similarity measure between the boundary of the prediction and the ground truth, and their average value ($\mathcal{J}\&\mathcal{F}$). We evaluate our results on the official evaluation server or use the official tools.

5.1 Compare with the State-of-the-art Methods

YouTube-VOS. [40] is the latest large-scale dataset for multi-object video segmentation. Compared to the popular DAVIS benchmark that consists of 120 videos, YouTube-VOS is about 37 times larger. In detail, the dataset contains 3471 videos in the training set (65 categories), 507 videos in the validation set (additional 26 unseen categories), and 541 videos in the test set (additional 29 unseen categories). Due to the existence of unseen object categories, the
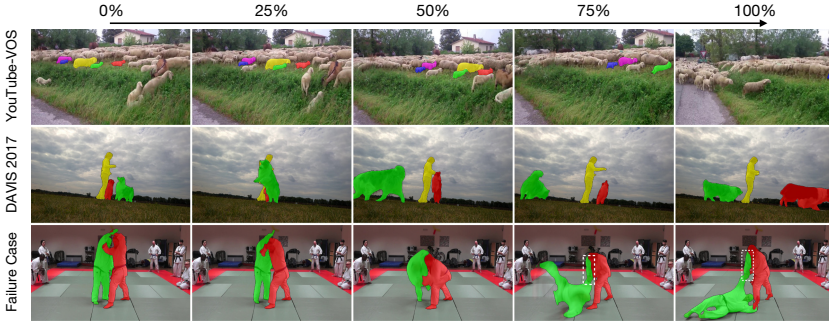
Fig. 7. Qualitative results on DAVIS 2017 and YouTube-VOS. In the first video, we succeed in tracking many similar-looking sheep. In the second video, our CFBI tracks the person and the dog with a red mask after occlusion well. In the last video, CFBI fails to segment one hand of the right person (the white box). A possible reason is that the two persons are too similar and close. (Color figure online)

YouTube-VOS validation set is much suitable for measuring the generalization ability of different methods.

As shown in Table 1, we compare our method to existing methods on both Validation 2018 and Testing 2019 splits. Without using any bells and whistles, like fine-tuning at test time [2,35] or pre-training on larger augmented simulated data [27,38], our method achieves an average score of **81.4**%, which significantly outperforms all other methods in every evaluation metric. Particularly, the 81.4% result is 2.0% higher than the previous state-of-the-art method, STMVOS, which uses extensive simulated data from [7,12, 15,22,32] for training. Without simulated data, the performance of STMVOS will drop from 79.4% to 68.2%. More-

Table 2. The quantitative evaluation on DAVIS 2016 [30] validation set. (**Y**) denotes using YouTube-VOS for training.

Methods	F	S	Avg	$\mathcal{J}$	$\mathcal{F}$	t/s
OSMN [42]	-		-	74.0		0.14
PML [5]			77.4	75.5	79.3	0.28
VideoMatch [18]			80.9	81.0	80.8	0.32
RGMP$^-$ [38]			68.8	68.6	68.9	0.14
RGMP [38]		✓	81.8	81.5	82.0	0.14
A-GAME [20] (**Y**)			82.1	82.2	82.0	**0.07**
FEELVOS [34] (**Y**)			81.7	81.1	82.2	0.45
OnAVOS [35]	✓		85.0	85.7	84.2	13
PReMVOS [23]	✓		86.8	84.9	88.6	32.8
STMVOS [27]		✓	86.5	84.8	88.1	0.16
STMVOS [27] (**Y**)		✓	**89.3**	**88.7**	89.9	0.16
CFBI			86.1	85.3	86.9	0.18
CFBI (**Y**)			**89.4**	88.3	**90.5**	0.18
CFBI^{MS} (**Y**)			**90.7**	**89.6**	**91.7**	9

over, we further boost our performance to **82.7**% by applying a multi-scale and flip strategy during the evaluation.

We also compare our method with two of the best results on the Testing 2019 split, *i.e.*, *Rank 1* (EMN [46]) and *Rank 2* (MST [45]) results in the 2nd Large-scale Video Object Segmentation Challenge. Without applying model ensemble,

our single-model result (**82.2**%) outperforms the *Rank 1* result (81.8%) in the unseen and average metrics, which further demonstrates our generalization ability and effectiveness.

DAVIS 2016. [30] contains 20 videos annotated with high-quality masks each for a single target object. We compare our CFBI method with state-of-the-art methods in Table 2. On the DAVIS-2016 validation set, our method trained with an additional YouTube-VOS training set achieves an average score of **89.4**%, which is slightly better than STMVOS (89.3%), a method using simulated data as mentioned before. The accuracy gap between CFBI and STMVOS on DAVIS is smaller than the gap on YouTube-VOS. A possible reason is that DAVIS is too small and easy to over-fit. Compare to a much fair baseline (*i.e.*, FEELVOS) whose setting is same to ours, the proposed CFBI not only achieves much better accuracy (**89.4**% *vs.* 81.7%) but also maintains a comparable fast inference speed (0.18*s vs.* 0.45*s*). After applying multi-scale and flip for evaluation, we can improve the performance from **89.4**% to **90.1**%. However, this strategy will cost much more inference time (9*s*).

DAVIS 2017. [31] is a multi-object extension of DAVIS 2016. The validation set of DAVIS 2017 consists of 59 objects in 30 videos. Next, we evaluate the generalization ability of our model on the popular DAVIS-2017 benchmark.

As shown in Table 3, our CFBI makes significantly improvement over FEELVOS (**81.9**% *vs.* 71.5%). Besides, our CFBI without using simulated data is slightly better than the previous state-of-the-art method, STMVOS (**81.9**% *vs.* 81.8%). We show some examples compared with STMVOS in Fig. 6. Same as previous experiments, the augmentation in evaluation can further boost the results to a higher score of **83.3**%. We also evaluate our method on the testing split of DAVIS 2017, which is much more challenging than the validation split. As shown in Table 3, we significantly outperforms STMVOS (72.2%) by **2.6**%. By applying augmentation, we can further boost the result to **77.5**%. The strong results prove that our method has the best generalization ability among the latest methods.

Table 3. The quantitative evaluation on DAVIS-2017 [31].

Methods	F	S	Avg	$\mathcal{J}$	$\mathcal{F}$
Validation split					
OSMN [42]			54.8	52.5	57.1
VideoMatch [18]			62.4	56.5	68.2
OnAVOS [35]	✓		63.6	61.0	66.1
RGMP [38]		✓	66.7	64.8	68.6
A-GAME [20] (**Y**)			70.0	67.2	72.7
FEELVOS [34] (**Y**)			71.5	69.1	74.0
PReMVOS [23]	✓		77.8	73.9	81.7
STMVOS [27]		✓	71.6	69.2	74.0
STMVOS [27] (**Y**)		✓	**81.8**	**79.2**	**84.3**
CFBI			74.9	72.1	77.7
CFBI (**Y**)			**81.9**	**79.1**	**84.6**
CFBI^{MS} (**Y**)			**83.3**	**80.5**	**86.0**
Testing split					
OSMN [42]			41.3	37.7	44.9
OnAVOS [35]	✓		56.5	53.4	59.6
RGMP [38]		✓	52.9	51.3	54.4
FEELVOS [34] (**Y**)			57.8	55.2	60.5
PReMVOS [23]	✓		71.6	67.5	75.7
STMVOS [27] (**Y**)		✓	72.2	69.3	75.2
CFBI (**Y**)			**74.8**	**71.1**	**78.5**
CFBI^{MS} (**Y**)			**77.5**	**73.8**	**81.1**

Qualitative Results. We show more results of CFBI on the validation set of DAVIS 2017 (**81.9**%) and YouTube-VOS (**81.4**%) in Fig. 7. It can be seen that CFBI is capable of producing accurate segmentation under challenging situations, such as large motion, occlusion, blur, and similar objects. In the *sheep* video, CFBI succeeds in tracking five selected sheep inside a crowded flock. In the *judo* video, CFBI fails to segment one hand of the right person. A possible reason is that the two persons are too similar in appearance and too close in position. Besides, their hands are with blur appearance due to the fast motion.

5.2   Ablation Study

We analyze the ablation effect of each component proposed in CFBI on the DAVIS-2017 validation set. Following FEELVOS, we only use the DAVIS-2017 training set as training data for these experiments.

Table 4. Ablation of background embedding. P and I separately denote the pixel-level matching and instance-level attention. * denotes removing the foreground and background bias.

P	I	Avg	$\mathcal{J}$	$\mathcal{F}$
✓	✓	74.9	72.1	77.7
✓*	✓	72.8	69.5	76.1
✓		73.0	69.9	76.0
	✓	72.3	69.1	75.4
		70.9	68.2	73.6

Background Embedding. As shown in Table 4, we first analyze the influence of removing the background embedding while keeping the foreground only as [34,42]. Without any background mechanisms, the result of our method heavily drops from 74.9% to 70.9%. This result shows that it is significant to embed both foreground and background features collaboratively. Besides, the missing of background information in the pixel-level matching or the instance-level attention will decrease the result to 73.0% or 72.3% separately. Thus, compared to instance-level attention, the pixel-level matching performance is more sensitive to the effect of background embedding. A possible reason for this phenomenon is that the possibility of existing some background pixels similar to the foreground is higher than some background instances. Finally, we remove the foreground and background bias, b_F and b_B, from the distance metric and the result drops to 72.8%, which further shows that the distance between foreground pixels and the distance between background pixels should be separately considered.

Other Components. The ablation study of other proposed components is shown in Table 5. Line 0 (74.9%) is the result of proposed CFBI, and Line 6 (68.3%) is our baseline method reproduced by us. Under the same setting, our CFBI significantly outperforms the baseline.

In line 1, we use only one local neighborhood window to conduct

Table 5. Ablation of other components.

	Ablation	Avg	$\mathcal{J}$	$\mathcal{F}$
0	Ours (CFBI)	74.9	72.1	77.7
1	w/o multi-local windows	73.8	70.8	76.8
2	w/o sequential training	73.3	70.8	75.7
3	w/o collaborative ensembler	73.3	70.5	76.1
4	w/o balanced random-crop	72.8	69.8	75.8
5	w/o instance-level attention	72.7	69.8	75.5
6	Baseline (FEELVOS)	68.3	65.6	70.9

the local matching following the setting of FEELVOS, which degrades the result from 74.9% to 73.8%. It demonstrates that our multi-local matching module is more robust and effective than the single-local matching module of FEELVOS. Notably, the computational complexity of multi-local matching dominantly depends on the biggest local window size because we use the intermediate results of the local matching of the biggest window to calculate on smaller windows.

In line 2, we replace our sequential training by using ground-truth masks instead of network predictions as the previous mask. By doing this, the performance of CFBI drops from 74.9% to 73.3%, which shows the effectiveness of our sequential training under the same setting.

In line 3, we replace our collaborative ensembler with 4 depth-wise separable convolutional layers. This architecture is the same as the dynamic segmentation head of [34]. Compared to our collaborative ensembler, the dynamic segmentation head has much smaller receptive fields and performs 1.6% worse.

In line 4, we use normal random-crop instead of our balanced random-crop during the training process. In this situation, the performance drops by 2.1% to 72.8% as well. As expected, our balanced random-crop is successful in relieving the model form biasing to background attributes.

In line 5, we disable the use of instance-level attention as guidance information to the collaborative ensembler, which means we only use pixel-level information to guide the prediction. In this case, the result deteriorates even further to 72.7, which proves that instance-level information can further help the segmentation with pixel-level information.

In summary, we explain the effectiveness of each proposed component of CFBI. For VOS, it is necessary to embed both foreground and background features. Besides, the model will be more robust by combining pixel-level information and instance-level information, and by using more local windows in the matching between two continuous frames. Apart from this, the proposed balanced random-crop and sequential training are useful but straightforward in improving training performance.

6   Conclusion

This paper proposes a novel framework for video object segmentation by introducing collaborative foreground-background integration and achieves new state-of-the-art results on three popular benchmarks. Specifically, we impose the feature embedding from the foreground target and its corresponding background to be contrastive. Moreover, we integrate both pixel-level and instance-level embeddings to make our framework robust to various object scales while keeping the network simple and fast. We hope CFBI will serve as a solid baseline and help ease the future research of VOS and related areas, such as video object tracking and interactive video editing.

References

1. Parallel distributed deep learning: Machine learning framework from industrial practice. https://www.paddlepaddle.org.cn/
2. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR, pp. 221–230 (2017)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. TPAMI **40**(4), 834–848 (2017)
4. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, Part VII. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
5. Chen, Y., Pont-Tuset, J., Montes, A., Van Gool, L.: Blazingly fast video object segmentation with pixel-wise metric learning. In: CVPR, pp. 1189–1198 (2018)
6. Cheng, J., Tsai, Y.H., Hung, W.C., Wang, S., Yang, M.H.: Fast and accurate online video object segmentation via tracking parts. In: CVPR, pp. 7415–7424 (2018)
7. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.M.: Global contrast based salient region detection. TPAMI **37**(3), 569–582 (2014)
8. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: CVPR, pp. 1251–1258 (2017)
9. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: ICML (2017)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR, pp. 248–255. IEEE (2009)
11. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: ICCV, pp. 2758–2766 (2015)
12. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. IJCV **88**(2), 303–338 (2010)
13. Feng, Q., Yang, Z., Li, P., Wei, Y., Yang, Y.: Dual embedding learning for video instance segmentation. In: ICCV Workshops (2019)
14. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: ICML (2017)
15. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV, pp. 991–998. IEEE (2011)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
17. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
18. Hu, Y.-T., Huang, J.-B., Schwing, A.G.: VideoMatch: matching based video object segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, Part VIII. LNCS, vol. 11212, pp. 56–73. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01237-3_4
19. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
20. Johnander, J., Danelljan, M., Brissman, E., Khan, F.S., Felsberg, M.: A generative appearance model for end-to-end video object segmentation. In: CVPR, pp. 8953–8962 (2019)
21. Liang, C., Yang, Z., Miao, J., Wei, Y., Yang, Y.: Memory aggregated CFBI+ for interactive video object segmentation. In: CVPR Workshops (2020)

22. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

23. Luiten, J., Voigtlaender, P., Leibe, B.: PReMVOS: proposal-generation, refinement and merging for video object segmentation. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV 2018, Part IV. LNCS, vol. 11364, pp. 565–580. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20870-7_35

24. Miao, J., Wei, Y., Yang, Y.: Memory aggregation networks for efficient interactive video object segmentation. In: CVPR (2020)

25. Ngan, K.N., Li, H.: Video Segmentation and Its Applications. Springer Science & Business Media, New York (2011). https://doi.org/10.1007/978-1-4419-9482-0

26. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Fast user-guided video object segmentation by interaction-and-propagation networks. In: CVPR, pp. 5247–5256 (2019)

27. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019)

28. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)

29. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR, pp. 2663–2672 (2017)

30. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR, pp. 724–732 (2016)

31. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)

32. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended CSSD. TPAMI **38**(4), 717–729 (2015)

33. Torralba, A.: Contextual priming for object detection. IJCV **53**(2), 169–191 (2003)

34. Voigtlaender, P., Chai, Y., Schroff, F., Adam, H., Leibe, B., Chen, L.C.: FEELVOS: fast end-to-end embedding learning for video object segmentation. In: CVPR, pp. 9481–9490 (2019)

35. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. In: BMVC (2017)

36. Voigtlaender, P., Luiten, J., Leibe, B.: BoLTVOS: Box-level tracking for video object segmentation. arXiv preprint arXiv:1904.04552 (2019)

37. Wu, Y., He, K.: Group normalization. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018, Part XIII. LNCS, vol. 11217, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_1

38. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR, pp. 7376–7385 (2018)

39. Xiao, H., Feng, J., Lin, G., Liu, Y., Zhang, M.: MoNet: deep motion exploitation for video object segmentation. In: CVPR, pp. 1140–1148 (2018)

40. Xu, N., et al.: YouTube-VOS: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018)

41. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV, pp. 5188–5197 (2019)

42. Yang, L., Wang, Y., Xiong, X., Yang, J., Katsaggelos, A.K.: Efficient video object segmentation via network modulation. In: CVPR, pp. 6499–6507 (2018)

43. Yang, Z., Zhu, L., Wu, Y., Yang, Y.: Gated channel transformation for visual recognition. In: CVPR (2020)

44. Zhang, Z., Fidler, S., Urtasun, R.: Instance-level segmentation for autonomous driving with deep densely connected MRFs. In: CVPR, pp. 669–677 (2016)
45. Zhou, Q., et al.: Motion-guided spatial time attention for video object segmentation. In: ICCV Workshops (2019)
46. Zhou, Z., et al.: Enhanced memory network for video segmentation. In: ICCV Workshops (2019)