Abstract

Training a Video Object Segmentation (VOS) model requires an abundance of manually labelled training videos. The de-facto way of annotating objects requires humans to draw detailed segmentation masks on the target objects at each frame, which is tedious and time-consuming. To reduce this annotation cost, we propose EVA-VOS, a human-in-the-loop Efficient Video Annotation framework for VOS. Unlike de-facto approaches, we introduce an agent that predicts iteratively both which frame to annotate and which annotation type to use. Then, the annotator annotates only the selected frame that is used to update a VOS module, leading to significant gains in annotation time. We experiment on the MOSE dataset and show that: (a) EVA-VOS leads to masks with accuracy close to the human agreement 3.5× faster than the standard way of annotating videos; (b) our frame selection achieves state-of-the-art performance; (c) EVA-VOS yields significant performance gains in terms of annotation time compared to all methods and baselines. Code data and models are available online.

1. Introduction

Video object segmentation (VOS) is the task of segmenting and tracking objects in videos [6, 48, 22, 42, 21, 43, 29, 31, 13, 12, 49, 50, 11, 4]. VOS is a central task for video understanding and enables various applications including video editing [23, 2], synthesis [44, 45], and decomposition [51]. Training a VOS model requires videos where the target objects have been manually annotated with object segmentation masks [6, 31, 29, 13, 12, 49, 50, 11, 4]. This process is expensive as it requires humans to manually draw a mask at each video frame, requiring 80 seconds per object per frame [28]. For instance, annotating only one object in a 10-second video would require more than 5 hours.

To address these, two groups of solutions are adopted. The first solution consists in sparsely annotating large VOS datasets [47, 38, 16, 15, 46]. The standard way of annotating a VOS dataset [36, 37, 47, 46, 15, 41, 38, 16] starts by uniformly sampling a subset of frames (1-5 fps) and then each frame is manually annotated with a mask by an annotator. In some datasets [15, 46], the sparsely annotated masks are interpolated to predict dense annotations.

The second solution consists in minimizing the annotation cost, typically by interactive segmentation using faster annotation types, such as clicks, scribbles, or bounding boxes. Albeit the progress on still images [1, 5, 8, 27, 30, 33, 35] with large datasets [5], there is limited work for videos [7, 9, 10, 14]. The most relevant to our work is [7] that propose a human-in-the-loop interactive VOS. The annotator provides a scribble and a VOS method predicts a mask for each frame. Then, the annotator iteratively selects the frame with the worst segmentation quality and provides scribbles. Despite its novelty, it has two limitations. First, it is unrealistic as the annotator cannot identify the worst frame; even if they could, it would require significant time [20], which defeats the cost minimization. Second, in challenging frames, low-cost annotation types cannot create good masks, and drawing the full mask is required.

To overcome these, we propose EVA-VOS, an Efficient Video Annotation pipeline for Video Object Segmentation (Fig. 1). We introduce an agent that predicts iteratively which frame should be annotated (frame selection) and which annotation type to use (annotation selection). Our agent is trained to maximize the annotation impact on the
segmentation quality while minimizing the annotation cost. For frame selection, we train a model to regress the quality of a segmentation mask. Then, we select the frame with the maximum distance from its closest pre-annotated frame. For annotation selection, we train a deep RL policy to select an annotation type (action) by maximizing the fraction of the segmentation quality improvement over the annotation time of the annotation type (reward). Our method iterates between (a) selecting the next frame for annotation and the annotation type, (b) asking annotators to improve a segmentation mask, and (c) predicting new object masks for all frames (Fig. 2). We experiment on the MOSE [16] dataset. We evaluate both each stage independently and our full pipeline. We show that (a) EVA-VOS leads to masks with accuracy close to human agreement 3.5× faster than the standard way of annotating a VOS dataset; (b) Our frame selection method achieves state-of-the-art performance; (c) EVA-VOS yields significant performance gains in terms of annotation time compared to other strong baselines.

2. Method

We propose EVA-VOS, a human-in-the-loop pipeline to annotate videos with segmentation masks using as little annotation as possible (Fig. 2). EVA-VOS consists of four stages: (a) Mask Propagation (Sec. 2.1), (b) Frame Selection (Sec. 2.2), (c) Annotation Selection (Sec. 2.3), (d) Annotation and Mask Prediction (Sec. 2.4).

More formally, at each iteration \( t \), the mask propagation receives the input video \( V = \{ f_1, f_2, \ldots, f_N \} \) of \( N \) frames and a set \( K \) containing all previously annotated frames to predict a new set of masks \( M^t = \{ M^t_1, M^t_2, \ldots, M^t_N \} \) for all frames. Then, the frame selection determines the frame \( f_\ast \) that should be annotated given \( V \) and \( M^t \). The annotation selection determines the most suitable annotation type \( a_{f_\ast} \) from a pool of candidate annotation types \( A = \{ a_1, a_2, \ldots, a_L \} \). For annotation types, we consider both the case where the annotator manually draws a complete mask (‘mask drawing’), and the case of weak annotations, where the human intervention is much faster, e.g., ‘corrective clicks’, ‘bounding boxes’, ‘scribbles’, etc. Finally, the annotator annotates \( f_\ast \) with \( a_{f_\ast} \), and the annotation is passed on to the mask prediction, where a new mask \( M^{t+1}_{f_\ast} \) is predicted and added to \( K \). Note that at \( t = 0 \), the annotator selects the target object and draws a mask on \( f_1 \).

2.1. Mask propagation

We predict masks \( M^t \) for all frames using all annotated masks from \( K \). We use a pre-trained VOS [12] with inputs video \( V \) and masks \( K \) and output mask \( M_t \) for each frame.

2.2. Frame selection

Given \( V \), \( M^t \), and \( K \), we aim at finding the frame to be annotated \( f_\ast \) at iteration \( t \) with the highest improvement on the video segmentation quality at iteration \( t + 1 \). Intuitively, we select frames that maximize the diversity among the selected ones while having low segmentation quality. We train a model to assess the segmentation quality of each frame, and then use the learned frame representations to select \( f_\ast \).

Architecture. To access the mask quality of each frame, we propose the Quality Network (QNet) which takes in a frame \( f_i \) and its mask \( M^t_i \) and performs mask quality classification into \( B \) classes, where 0 represents the worst quality and \( B - 1 \) the best. \( B \) determines the number of bins of the segmentation quality. QNet consists of two image encoders [19] in parallel branches, one for frame \( f_i \) and one for mask \( M^t_i \). The embeddings from each encoder are concatenated and fed into a linear classifier of \( B \) outputs.

Training. We train QNet in a supervised way with cross-entropy loss on a simulated training set. To generate it, we simulate a number of iterations with EVA-VOS (Fig. 2); at each iteration, we compute the segmentation quality of each frame and assign a quality label to each mask \( M^t_i \). To further augment it, we include random selections.

Selected frame. \( f_\ast \) is the one with the maximum distance in the feature space from its closest previously annotated frame. We (1) extract embeddings \( E_i \) from QNet; (2) compute the L2 between each embedding of frame \( j \) in \( K \) and frames of \( V \) (3) assign the minimum distance to each embedding \( i \), and select the frame with the maximum distance:

\[
 f_\ast = \arg \max_{i \in \{1, 2, \ldots, N\}} \left\{ \min_{j \in \{1, 2, \ldots, t\}} \{d(E_i, E_j)\} \right\} \tag{1}
\]

2.3. Annotation selection

Given annotation types \( A = \{a_1, a_2, \ldots, a_L\} \), this step chooses the most suitable type \( a_{f_\ast} \) for \( f_\ast \). We formulate this as a Markov Decision Process and train a model using reinforcement learning (RL). The model observes the image of \( f_\ast \) and its predicted mask \( M^t_{f_\ast} \) and predicts the most suitable annotation type \( a_{f_\ast} \). This \( a_{f_\ast} \) is used by the annotator to generate a new mask \( M^{t+1}_{f_\ast} \) for \( f_\ast \). The annotation is performed iteratively (e.g. 3 clicks are performed one by one). Therefore, we denote the annotation iteration as \( g \). The input \( M^t_{f_\ast} \) has an initial segmentation quality \( SQ_1 \) (\( g = 1 \)).

Environment. The state of the environment consists of \( f_\ast \) and its mask. Each step \( g \) yields \( SQ_g \) using the input action, which represents an annotation type from \( A \).

Reward. The reward function reflects the trade-off between the quality of \( M^t_{f_\ast} \) and the cost of the annotation type. Each \( a \in A \) requires a different annotation cost \( \theta_a \). The reward at \( g \) is formulated by comparing \( SQ \) before and after annotation, divided by the total cost \( tc \) at \( g \) which is the sum of the costs \( \theta_a \) of all annotation types until \( g \): 

\[
r = \frac{SQ_{g+1} - SQ_g}{tc_g} \tag{2}
\]

Architecture. The model has two image encoders [26, 19] in parallel branches, one for the frame \( f_i \) and one for the mask \( M^t_i \). The extracted embeddings from each encoder are then concatenated and fed into two linear layers. The
Figure 2: EVA-VOS. At each iteration $t$, Mask propagation (Sec. 2.1) receives a video $V$ of $N$ frames and a set $K$ containing all previously annotated frames to predict a new set of masks $M^t = \{M^t_1, M^t_2, \ldots, M^t_N\}$ for all frames. Subsequently, the Frame selection (Sec. 2.2) stage selects the frame $f_*$ that should be annotated given the video $V$, the predicted masks $M^t$ and all previously annotated frames $K$. The Annotation selection (Sec. 2.3) takes as input the selected frame $f_*$ and its corresponding mask to predict the most suitable annotation type $a_{f_*}$. Finally, in the Human annotation and mask prediction (Sec. 2.4) stage, the annotator interacts with $f_*$ using the annotation type $a_{f_*}$ and A2M (in this work, we use SAM [26]) predicts the new mask $M^{t+1}_{f_*}$ of the frame $f_*$, which is then added to the set $K$.

first layer has $L$ outputs (possible annotation types), while the second layer has one output for the RL value.

Training. We use Proximal Policy Optimization [40] (PPO) to train our model. At training, we use the simulated masks described in Sec. 2.2. We perform multiple environment steps and the process terminates when we reach the maximum steps or the type of drawing a mask is selected.

Video Ranking. We use the predicted value of our RL agent to estimate the improvement of each annotation at each video. This enables ranking the videos and performing more annotation iterations in videos with higher RL value.

2.4. Human annotation and mask prediction

The annotator interacts with the $f_*$ to create the input $a_{f_*}$. When $a_{f_*}$ is ‘mask drawing’, the annotator draws a detailed $M^{t+1}_{f_*}$. Otherwise, i.e. clicks, this step predicts a new mask $M^{t+1}_{f_*}$ using a pre-trained A2M model. Since EVA-VOS is independent of this model, we opt for the recently introduced Segment Anything Model (SAM) [26].

3. Experimental setting

Datasets. We use the MOSE dataset [16] which contains 1507 videos with available ground-truth segmentation masks. We only consider videos with 15 to 104 frames leading to a MOSE-long with 1166 videos. We split it into 800 training, 150 validation, and 216 test videos.

Metrics. We use the curve of $J & F$ vs time [7] and the annotation time at $J & F = 0.85$ (human annotation agreement for instance segmentation [5, 18, 25, 53]). We consider 80 sec for drawing a mask [28] and 1.5 sec for each click plus 1 sec of overhead to locate the object [3, 5, 34].

Implementation details. QNet consists of two ResNet-18 [19]. We train it using SGD with $lr=10^{-5}$, batch size 64, 30 epochs, with $B=20$. The frame branch of the RL agent is the image encoder of SAM [26]; the mask branch is ResNet-18 [19]. The RL agent is trained using Adam [24] and $lr=10^{-5}$ for 50K iterations. The VOS module is pre-trained on the YouTubeVOS dataset [47]. We consider two annotation types: ‘mask drawing’ and ‘corrective clicks’ [5], denoted as Mask and Clicks, respectively. For Clicks, the annotator clicks 3 times to improve the segmentation and decides the number of positive and negative clicks.

Human annotator simulation. We perform experiments only by simulating human intervention. Given $M^t_{f_*}$ and ground-truth mask $m_g$ of $f_*$, we simulate positive and negative clicks to prompt SAM [26] similar to how a human would. We identify all false-neg and false-pos pixels between $m_g$ and $M^t_{f_*}$. We determine the connected components of each error region, and the center of the largest component is selected as click location, positive or negative.

4. Experimental results

This section presents our experimental results. We first evaluate the frame selection and annotation selection (Sec. 4.2) individually and display the results in Fig. 3(a) and (b), respectively. Finally, we analyze the results of our full pipeline (Sec. 4.3) and report results in Fig. 3(c).

4.1. Frame selection evaluation

Here, we evaluate the frame selection (Sec. 2.2). For a fair comparison among all methods, we use only object-mask as an annotation type (Fig. 3(a)).
Compared methods. We compare against a baseline that selects frames randomly and against IVOS-W [52] which is the state-of-the-art frame selection method for VOS. For a fair comparison, we train IVOS-W [52] in MOSE-long. We also use powerful image encoders [19, 17, 32] pretrained for image classification [39] to compute embeddings in Eq. (1) and compare against QNet. We implement an oracle approach that selects the frame with the worst $\mathcal{J} \& \mathcal{F}$ and an upper bound approach that selects the frame with the highest impact on the propagation stage after annotation.

Comparison to the state of the art. 
Random is shown as the green line in Fig. 3(a). We run all random baselines 15 times and report the average result. EVA-VOS (Ours) is shown as the black line in Fig. 3(a)). Given the same annotation time, our framework consistently outperforms Random. For instance, we achieve $\mathcal{J} \& \mathcal{F} = 0.85$ at 80.7 hours, 26.7 hours faster than Random. State-of-the-art frame selection (IVOS-W [52]) performs significantly worse than our method. We observe that our method reaches $\mathcal{J} \& \mathcal{F} = 0.85$ 2.3× faster than IVOS-W. L2-Encoders yield approximately the same performance as random. This shows that our task-specific QNet learns much better representations and outperforms all pre-trained encoders that have even 28× more parameters. Oracle is shown as the red dashed line in Fig. 3(a)). Interestingly, we observe that for low budgets (up to 40 hours), our method yields almost identical $\mathcal{J} \& \mathcal{F}$. Upper bound consistently outperforms the oracle indicating that the frame with the worst $\mathcal{J} \& \mathcal{F}$ is not the most impactful one. Interestingly, the upper bound is only 2.1× faster than our method at $\mathcal{J} \& \mathcal{F} = 0.85$.

4.2. Annotation selection evaluation

Here, we evaluate only our annotation selection stage. For a fair comparison, we set the frame selection for all approaches to oracle, i.e., the frame with the worst $\mathcal{J} \& \mathcal{F}$ is selected to be annotated at each iteration (results in Fig. 3(b)). Compared methods. We compare our annotation selection to approaches that consider one annotation type (Clicks, Mask), a random approach selecting $a_f$, randomly, and an oracle that selects the $a_f$ that yields the maximum quality improvement normalized by the annotation cost.

Comparison to annotation selection methods EVA-VOS (Ours) is shown as the black line in Fig. 3(b). It reaches $\mathcal{J} \& \mathcal{F} = 0.85$ in only 29.8 hours. Random is shown as the green line in Fig. 3(b). Even though it reaches $\mathcal{J} \& \mathcal{F} = 0.9$ at a similar time as our method, it performs significantly worse at lower budgets (e.g., we yield $\mathcal{J} \& \mathcal{F} = 0.85$ 1.9× faster). Mask-only performs consistently worse than random at all budgets, indicating that the traditional way of manually drawing object mask [36, 38, 47] is not a good approach. Clicks-only performs on par with our method at low annotation budgets. However, it plateaus quickly at lower $\mathcal{J} \& \mathcal{F}$ values and it is not able to reach $\mathcal{J} \& \mathcal{F} = 0.9$, whereas our method can yield higher $\mathcal{J} \& \mathcal{F}$ at larger budgets. Oracle (red dashed line in Fig. 3(b)) performs on par with our method at low budgets. Oracle performs better in very high annotation budgets that reach high $\mathcal{J} \& \mathcal{F}$ above 0.85.

4.3. Frame and Annotation selection evaluation

We evaluate here our full pipeline, showing the effect of both selection modules (results of full pipeline in Fig. 3(c)). Compared methods. Similar to Sec. 4.2, we compare our method to Clicks-only and Masks-only which select a random frame and consider only one annotation type.

Comparison of annotation methods. EVA-VOS (Ours) is shown as the black line in Fig. 3(c) and yield a $\mathcal{J} \& \mathcal{F}$ of 0.85 in 29.8 hours. Oracle uses both oracle frame selection and annotation selection and shows the trade-off that EVA-VOS could achieve with an ideal oracle training scenario. Mask-Only resembles the traditional way of annotating videos with object segmentation masks [47, 46, 15, 38, 16]. Our method performs significantly better and achieves a 3.5× speed up compared to Mask-only at $\mathcal{J} \& \mathcal{F} = 0.85$. Click-Only performs similarly to EVA-VOS at 50 hours but has a worse trade-off for either lower or higher budgets.

Conclusions. We presented the efficient EVA-VOS to annotate objects in videos with segmentation masks. EVA-VOS shows significant gains in annotation time (3.5× speed up) compared to the manual annotation of objects in videos. EVA-VOS reduces the total human annotation time while leading to high-quality segmentation masks.
References

[34] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Training object class detectors with click supervision. In CVPR, 2017.


