Human-in-the-Loop Video Semantic Segmentation Auto-Annotation

Nan Qiao[1], Yuyin Sun[1], Chong Liu[2], Lu Xia[1], Jiajia Luo[1], Ke Zhang[1], and Cheng-Hao Kuo[1]
[1]Device CoRo, Amazon [2]UC Santa Barbara
[1]{qiaonan, yuyinsun, luxial, lujiajia, kezha, chkuo}@amazon.com [2]chongliu@cs.ucsb.edu

Abstract

Accurate per-pixel semantic class annotations of the entire video are crucial for designing and evaluating video semantic segmentation algorithms. However, the annotations are usually limited to a small subset of the video frames due to the high annotation cost and limited budget in practice. In this paper, we propose a novel human-in-the-loop framework called HVSA to generate semantic segmentation annotations for the entire video using only a small annotation budget. Our method alternates between active sample selection and test-time fine-tuning algorithms until annotation quality is satisfied. In particular, the active sample selection algorithm picks the most important samples to get manual annotations, where the sample can be a video frame, a rectangle, or even a super-pixel. Further, the test-time fine-tuning algorithm propagates the manual annotations of selected samples to the entire video. Real-world experiments show that our method generates highly accurate and consistent semantic segmentation annotations while simultaneously enjoys significantly small annotation cost.

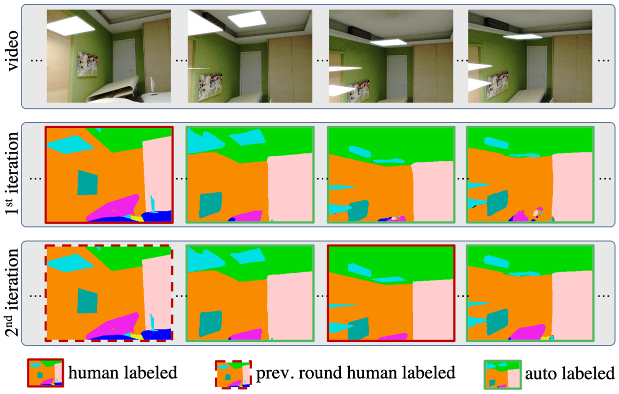


Figure 1. Performance of HVSA after 2 iterations. The method actively selects the most important samples to get human annotations in each iteration, then propagates the annotations to the entire video by jointly considering spatial-temporal consistency and semantic information of the video. Thus less human effort is required to obtain the high-quality pixel-level segmentation.

1. Introduction

Video-level segmentation annotations are important in multiple applications such as autonomous driving [19], flight [4], and augmented reality [23]. They also facilitate model training in other tasks like video deblurring/dehazing [35, 36], action recognition [22], and 3D reconstruction [24]. However, manually annotating per-pixel semantic segmentation labels for the entire video is usually expensive [14]. Therefore, a typical method is to only sample a subset of video frames to get human annotations [14, 6]. And then given sparsely annotated frames, the method applies Label Propagation (LP) to populate annotations on selected frames to all frames to get dense annotations [7, 3, 8]. Unfortunately, these annotate-once-then-propagate methods do not utilize annotation budget efficiently.

To annotate the entire video with semantic segmentation labels at a low cost, we propose the Human-in-the-loop Video Semantic segmentation Auto-annotation (HVSA) framework. Unlike most work that annotates sampled frames only *once*, our HVSA framework works *iteratively*, keeping collecting annotations and updating segmentation models at the same time unitl high quality of segmentation is satisfied. See Fig. 2. In each iteration of HVSA, samples are *actively* selected to be manually annotated and then a video-specific network is fine-tuned based on the accumulated manual annotations. The updated outputs of the network can be used in the next iteration to decide which sample to select for human annotations. Finally, the well fine-tuned network is used to generate segmentation annotations for the entire video. To the best of our knowledge, HVSA is the first human-in-the-loop framework that applies active sample selection for efficient video semantic segmentation auto-annotation. See Fig. 1.

To select video frames for annotation, most existing work only uses naive strategies, e.g., the first few frames, uniformly random sampling, or arbitrarily random sampling [3, 2, 8]. These strategies do not consider the video content or domain knowledge, leading to low utilization of the limited manual annotation budget. Instead, in our HVSA framework, we propose Active Sample Selection

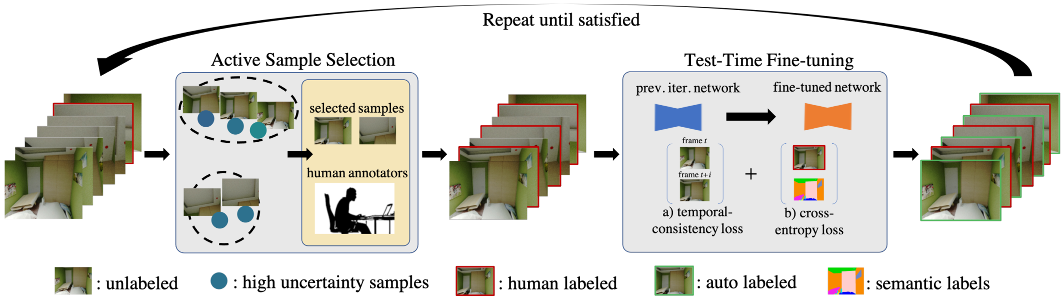


Figure 2. Overview of the HVSA framework. *Active sample selection* searches for uncertain and diverse samples from the input video. *Test-time fine-tuning* fine-tunes the image-based semantic segmentation network from the previous iteration by minimizing two complementary losses (a & b). The whole process repeats until high quality semantic segmentation is satisfied.

(ASS) method, which takes both video content and semantic segmentation network into consideration. In detail, we evaluate the prediction uncertainty of the network and try to select samples with least prediction confidence. Also, we generate features of all samples and try to select the most representative samples. In this way, our ASS method not only samples by uncertainty but also by diversity, so it is able to improve the utilization of manual annotation budget and boost the label propagation accuracy.

Curious readers may find that we are doing active *sample* selection, rather than active *frame* selection as in previous work. This is because one of the critical considerations in semantic segmentation is the granularity of the annotation unit. It has been studied in the image semantic segmentation tasks, including frame-based [45, 41, 16], rectangle-based [28, 10, 13], and super pixel-based [40, 9] work. The recent work [9] suggests that super pixel-based annotation is the most efficient for image segmentation tasks. In our ASS method, the sample can be a frame, a rectangle of frame, or even a super pixel. Moreover, to resemble real-world manual annotation process, we first adopt the click-based annotation measurement [28, 13] to simulate annotation cost, then generate "manual annotations" based on clicks and use them in the evaluation. Our experiment results show that optimal granularity in video annotation task is not determined but depends on desired level of annotation quality.

Traditional LP methods [7, 3, 8] propagate manual annotations of selected frames to the entire video only using spatial-temporal information. Therefore, they do not take advantage of semantic information captured in existing semantic segmentation models or manual annotations, leading more manual annotations to fill where the spatial-temporal constraints do not cover. In Test-time Fine-Tuning (TFT) method of our HVSA framework, we design a new loss function considering both spatial-temporal consistency and semantic information in model fine-tuning. It further improves label propagation quality and saves annotation cost.

In summary, our contributions include:

1. A novel human-in-the-loop framework HVSA, alternating between active sample selection and test-time fine-tuning methods, is proposed for video semantic segmentation auto-annotation at a low annotation cost.
2. In active sample selection, the sample can be a frame, a rectangle of frame, or even a super-pixel. And samples are selected by both uncertainty of the network and the diversity among samples, taking advantage of information from both network and video.
3. In test-time fine-tuning, we propose a new loss function combining both the semantic knowledge and the spatial-temporal information.
4. We study the desired granularity for the video semantic segmentation auto-annotation problem. Our results give insights to the future work along the line in terms of selecting the annotation unit.
5. Real-world experiments, e.g., Fig. 1, demonstrate that our method generates highly accurate and consistent semantic segmentation annotations of the whole video at a low annotation cost.

2. Related Work

In this section, we briefly summarize related work due to page limit. See Appendix A for complete discussion.

Video semantic auto-annotation. Pseudo-labeling and semi-supervised learning are the two popular types of methods for automating video semantic segmentation annotations. The pseudo-labeling approaches [27] use a pre-trained teacher model to generate labels for the test video sequences. However, these approaches are typically frame-based and do not consider the rich temporal constraints in the videos. Among the semi-supervised learning approaches, Label Propagation (LP) is widely adapted [3, 2, 30, 18, 32]. These methods rely on accurate optical flow estimation, which is difficult to obtain. Instead, our test-time fine-tuning is optimizing a new loss that takes both semantic and temporal information into consideration and predicts temporally consistent semantic annotations across the full video without the limitations of traditional LP methods.

Active learning. Inspired by the success of active learning [39], previous methods [17, 40] studied how to select instances to refine a network for segmentation tasks. [43] studies the active *frame* selection problem for LP. Our work is different in two ways: First, their method selects frames only once, while our method could select video frames, rectangle of frames, or even super-pixels in multiple iterations. Second, their method closely ties with a particular LP technique and does not comply with modern deep networks. Our method is generic and can work with different segmentation networks.

Human-in-the-loop for visual annotations. There exists some work [1, 34] trying to reduce the annotation cost in human-in-the-loop model learning. And [20, 12] studied the interactive video object segmentation frameworks. However, solving video semantic segmentation problem in the human-in-the-loop framework has never been studied.

3. Methods

In this section, we describe our HVSA framework (Fig. 2) in detail, including pre-processing, active sample selection, test-time fine-tuning, and cost calculation.

3.1. Pre-Processing

Granularity of samples. A suitable sample granularity needs to be carefully chosen to minimize the human annotation effort. We investigate three types of annotation unit: *frame*, *rectangle*, and *super-pixel*, which are typically used in image semantic segmentation tasks. Fig. 3 shows example of three units. To get rectangle units, we uniformly crop each frame to non-overlapping rectangles. And we use DMMSS-FCN [21] to generate super-pixel units. The n-th sample from the t-th frame is denoted as s_t^n. For frame samples, n is always 0. All samples are prepared in the unlabeled sample pool at the beginning of our framework.

Build temporal correspondence. We rely on correspondences between frames to leverage video temporal information. Here we extract the dense correspondences by estimating optical flow [42], $O_{t\to t'}$, of a frame pair from frame t to t'. Computing optical flow for all frame pairs is expensive, thus we limit the distance between frames to be smaller than 3. We further apply a forward-backward consistency check to cope with occlusion/dis-occlusions to extract only reliable correspondences. As a result, each optical flow $O_{t\to t'}$ will have a binary mask $M_{t\to t'}$, where pixels with forward-backward flow difference larger than 1 pixel are set to 0.

3.2. Active Sample Selection

To reduce annotation cost, we propose the active sample selection (ASS) to *actively* select the most *important* samples for manual annotations in each iteration. The ASS method takes both the network and the video content into consideration, which involves uncertainty sampling and diversity sampling and their combination.

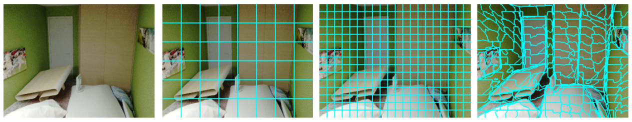


Figure 3. When annotating a video, user could annotate samples of frame (a), rectangle (b) (c), or even super-pixel [21] (d). Segments in (b) and (c) are of size 100 and 16 respectively. There are similar number of segments in (c) and (d).

Margin of confidence and uncertainty sampling. The motivation behind uncertainty sampling is that if a network predicts on a sample with little confidence, this sample needs to be selected for manual annotation. To capture confidence, we use the margin of confidence [25]. For each pixel, its margin of confidence is defined as the difference between the prediction scores of top-1 and top-2 label predictions from the model trained in each iteration. Intuitively, large margin means large prediction confidence. After being subtracted from 1, the pixel margin of confidence is converted to the pixel uncertainty. The uncertainty of sample s_t^n is then defined as the summation of pixel uncertainties within the sample region:

$$u(s_t^n) = \sum_{x\in s_t^n} P_{\theta_{k-1}}(y_{1,x}^*|I_t) - P_{\theta_{k-1}}(y_{2,x}^*|I_t), \quad (1)$$

where I_t is the input frame, y^* is the prediction from $softmax$, x is a pixel position within s_t^n, and θ_{k-1} is the previous model. By applying uncertainty sampling, the ASS method knows what are the samples that the current network is unsure about its prediction and then theses samples will be selected accordingly.

However, uncertainty sampling has a shortcoming in isolation. It might focus on one part of the decision boundary and select similar samples, causing a waste of human effort. To make the selection strategy comprehensive, we further require the method to samples that are different from each other, which refers to the *diversity sampling*.

Deep feature and diversity sampling. Clustering-based sampling naturally targets a diverse selection of samples. We first conduct clustering on unlabeled samples then select centroid samples for annotation. We re-use the downstream segmentation model as a feature extractor. Specifically, we transform each frame I_t to a feature map F_t using the previous model backbone network without segmentation head. Then the sample feature $\mathbf{f}_t^n$ is defined as the average along the spatial dimensions of F_t within s_t^n region:

$$\begin{aligned} F_t &= \psi_{\theta_{k-1}}(I_t), \\ \mathbf{f}_t^n &= \mathrm{MeanPool}_{x\in s_t^n}(F_{t,x}), \end{aligned} \quad (2)$$

where ψ denotes the segmentation network backbone. We employ the k-Means algorithm with Euclidean distance on $\mathbf{f}$ for clustering.

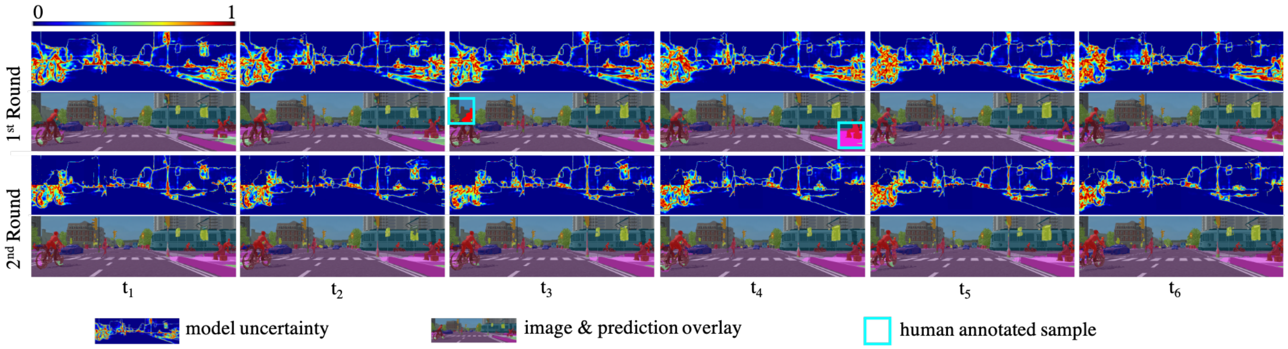
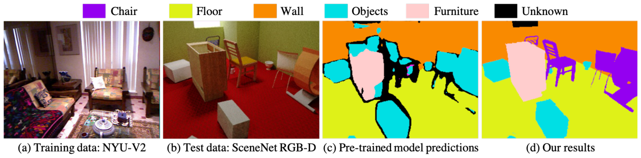


Figure 4. Visualization of model uncertainty and annotation selections on VEIS. After fine-tuning on the high uncertainty sample, the generated annotation in the second iteration improves significantly within sample regions across all neighbor frames.

Combining uncertainty and diversity sampling. In first iteration of our framework, as the network hasn't been fine-tuned, we only apply diversity sampling. Later in iterations, we first selects half of the most uncertain samples and cluster them into b clusters, where b is the annotation budget in one iteration. Then, b cluster centroids are selected and sent to human annotators. In this way, selected samples are of high uncertainty and are relatively different from each other. See Fig. 4 for an example.

3.3. Test-time Fine-tuning on Input Video

While a network may be pre-trained on relevant datasets, directly applying it to an arbitrary video would lead to inferior results, e.g., Figure 5. To progressively adapt it to a video, in each iteration, we fine-tune the model leveraging two different information sources, inspired by how human annotators handle the video annotation tasks. Given a target frame and the video, an annotator will naturally analyze its neighbor frames to decide the correct categories of the objects in the scene; The annotator will also refer to the existing annotations within the same video. Moreover, we propose a new loss designed from the two information sources, and show how we optimize it.



(a) Training data: NYU-V2 (b) Test data: SceneNet RGB-D (c) Pre-trained model predictions (d) Our results

Figure 5. The model pre-trained on NYU-V2 performs poorly on new out-of-domain input video as in (c). (d) Our framework adapts the model to the input video and produces better results.

Temporal consistency loss. Our *temporal consistency loss*, $\mathcal{L}_{tc}$, encourages consistent predictions across corresponding pixels on different frames. Unlike other methods [46, 8] which directly propagate labels between frames, we propagate predicted class probabilities from the model. More specifically, we penalize the difference between predicted class probabilities $\mathbf{q}_t$ and $\mathbf{q}'_t$ of frame t and t' at pixel position x as:

$$\mathcal{L}_{tc,t\rightarrow t'}(x) = M_{t\rightarrow t'}(x)\left\|\mathbf{q}_t(x) - \hat{\mathbf{q}}_{t'\rightarrow t}(x)\right\|_2^2, \tag{3}$$

where $\hat{\mathbf{q}}_{t'\rightarrow t}(x)$ is the warped prediction score from frame t' to t using the pre-computed flow $F_{t\rightarrow t'}$ and $M_{t\rightarrow t'}$ is the mask associated with $F_{t\rightarrow t'}$.

Here we illustrate why and how we have the mask $M_{t\rightarrow t'}$. We apply a forward-backward consistency check to cope with occlusion/dis-occlusions to extract only reliable correspondences. As a result, each optical flow $O_{t\rightarrow t'}$ will have a binary mask $M_{t\rightarrow t'}$, where pixels with forward-backward flow difference larger than 1 pixel are marked as 0. $M_{t\rightarrow t'}$ at position x can be formulated as

$$M^{(x)}_{t\rightarrow t'} = \mathbf{1}\left[\left\|O^{(x)}_{t\rightarrow t'} - \hat{O}^{(x)}_{t'\rightarrow t}\right\|_2^2 < 1\right], \tag{4}$$

where $\hat{O}_{t'\rightarrow t}$ is the warped version of $O_{t'\rightarrow t}$ using flow $O_{t\rightarrow t'}$. So that the position of $O_{t\rightarrow t'}$ and $\hat{O}_{t'\rightarrow t}$ is aligned and they can be compared directly.

Unlike the existing approaches [46] which only consider the temporal relation between annotated frames and their neighbors, we apply the temporal consistency loss to even unlabeled image pairs. As a result, labeled image information transforms to more than 3 frames away, which is the distance limitation of optical flow.

Semantic loss. Temporal constraints tell the model which pixel to share labels with but not where to hold. This semantic information will have to come from the annotated samples on the input video. We compute the regular cross-entropy loss, $\mathcal{L}_{ce}$, for any frame or frame region with manual annotations:

$$\mathcal{L}_{ce,t} = \mathcal{L}_{CE}(\mathbf{q}_t, L_t), \tag{5}$$

L_t denotes the semantic label at frame t, where unlabeled region is set to a special "ignored index".

Optimization. In the test-time fine-tuning, each training sample consists of two frames that pass through the single-frame model in parallel, giving two sets of class probability predictions. The two predictions are then used to compute the temporal loss $\mathcal{L}_{tc}$. If any frame region of the pair has manual annotations, the cross-entropy loss $\mathcal{L}_{ce}$ will be calculated as well. In summary, we fine-tune the single-frame segmentation network weights using standard backpropagation during test-time fine-tuning by minimizing:

$$\mathcal{L} = \lambda \mathcal{L}_{tc} + \mathcal{L}_{ce}. \tag{6}$$

We initialize the network weights using the pre-trained model in the first selection iteration. In later iterations, the network fine-tunes from the previous checkpoint, and then predicts segmentation labels on all the frames.

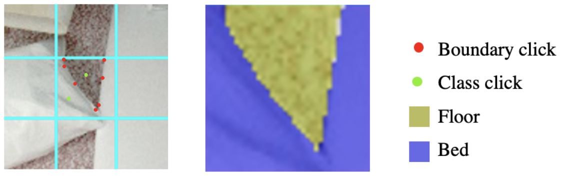
3.4. Annotation Cost Calculation

In practice, the annotation cost is measured by expense or human labeling time. Some conventional semantic segmentation AL work [40] uses percentage of labeled pixels to represent manual effort. We follow some recent work [28, 13, 9] to measure cost by annotation clicks, which is more realistic. Semantic segmentation label mask is pixel-level, while in actual labeling tasks, human annotators usually use a polygon-based tool [13]. Annotators first click on several vertices on the boundary of the one object to form a closed polygon ("Boundary click"), then select the object type by clicking once ("Class click"). In this way, all pixels within this polygon get the label of this class.

Here we introduce how we use algorithm to mimic human annotator to locate the "Boundary click" positions from the existing segmentation labels, and calculate the total clicks as the annotation cost. For each connect component of a single class object, we find its contour pixels, and simplify the contour pixels into some polygon vertices using Ramer–Douglas–Peucker (RDP) algorithm. Each polygon vertex costs one "Boundary click". In addition, each polygon costs one "Class click" to specify its class label. Fig. 6 shows an example.

For rectangle-based and super-pixel-based annotations, there are no clicks required on the sample boundary. If a sample only consists of a single class object, the required number of clicks is one "Class click". For super-pixel, unlike [9], we do not assign the dominant label to the entire super-pixel since the error label will be propagated to neighbor frames, hurting the final annotation quality.

Mimic "manual annotation". [9] uses a similar method to estimate annotation clicks, while using the GT labels provided by the dataset as training labels. However, this is not appropriate, as the RDP algorithm simplifies the object polygon boundaries, which leads to a rougher annotation of GT. In their case, the click-based cost is underestimated compared to the training label quality. On the contrary, we mimic manual annotation (MA) by converting the simplified polygons back to label masks. We use MA rather than GT in fine-tuning models, which better fits the video segmentation annotation tasks in practice.



(a) Annotation Clicks (b) Annotation Label

Figure 6. Example of annotating the center rectangle sample in (a). The red "Boundary click" are generated by the RDP algorithm from the original object contour. No clicks are needed in the boundaries of the sample to enclose the polygon. The green "Class click" specifies Bed and Floor class in this example. (b) is the segmentation label annotated by the shown 9 clicks.

4. Experiments

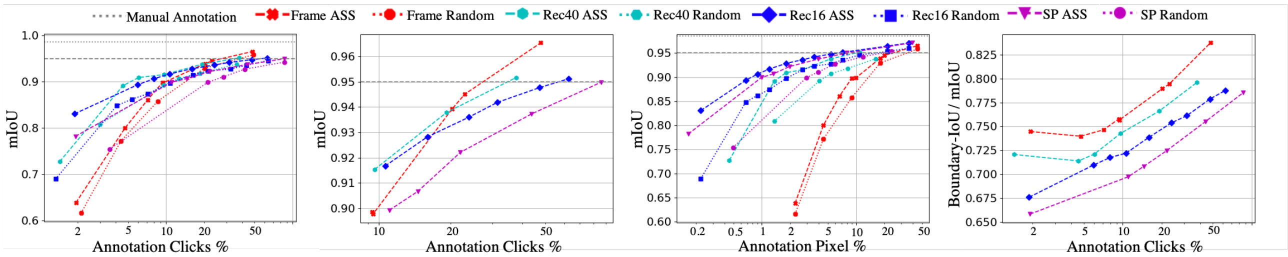
In this section, we conduct experiments on two datasets with dense segmentation GT on every frame to support the evaluation of the framework. We first compare the proposed ASS method with different frame selection baselines using various sample granularities. Then, we study the effectiveness of the proposed test-time fine-tuning by comparing it with other label propagation methods. Finally, we envision the generated annotations to show more details of the outputs from the proposed framework.

4.1. Experimental Settings

Training settings. We perform three iterations of ASS for each testing sequence. The annotation budget for each iteration is divided equally from the total budget.

We use the HRNet-W48 [44] as the backbone network (other networks can be easily incorporated). We set the consistency loss weight $\lambda = 1$. The initial learning rate in each iteration is 0.004. In each iteration, we fine-tune the network for 15 epochs with a learning rate of 0.004 and SGD optimizer [37] with momentum 0.9. We follow the "poly" learning rate policy to reduce the learning rate gradually. The batch size is 14 for SceneNet RGB-D [29] dataset, and 2 for VEIS [38] dataset.

Evaluation and metrics. We use four metrics to evaluate our method thoroughly, which are pixel accuracy, mean Intersection over Union (mIoU), boundary Intersection over Union (Boundary-IoU), and temporal consistency. The first two are commonly used in segmentation tasks to measure the accuracy of predictions. See Appendix B for details.



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

Figure 7. Active sample selection results on SceneNet RGB-D: (a),(b),(d) show the generated annotation mIoU and the normalized boundary-IoU in annotation clicks %, and (b) is a zoom-in version of (a). (c) shows the generated annotation mIoU in annotated pixel %.

4.2. Comparative Assessment

4.2.1 SceneNet RGB-D

We use the SceneNet RGB-D [29], which is a photorealistic indoor trajectory dataset with semantic segmentation annotations for every video frame to evaluate the overall system performance. Unlike regular indoor scene datasets [15, 31], the room layouts/object placements of the ScenNet RGB-D dataset are generated randomly. We train a 14-class HRNet-W48 model using the NYU-V2 [31] training set as the pre-trained model, which has only 15.04% mean-Intersection-over-Union (mIoU) on the SceneNet testing videos. We will demonstrate that our test-time fine-tuning method adapts the segmentation model to randomly generated scenes and achieves more satisfying results (examples in Fig. 5). We randomly picked five sequences from the SceneNet test set in our experiments, each containing 300 frames. We test four granularity settings: frame, 40×40-pixel rectangle, 16×16-pixel rectangle and super-pixel, denoted as Frame, Rec40, Rec16, and SP respectively. Given the SceneNet frame resolution of 240×320, Rec40 and Rec16 split a frame into 56 and 300 segments respectively. We let SP split a frame into about 300 segments.

We evaluate the generated annotations by measuring their mIoU with the GT. Fig. 7 compares the generated label quality from different selection methods and annotation sample granularities. The "Annotation Clicks %" (shown in log scale) is the number of annotation clicks normalized by the number of clicks to annotate the whole video. We can see from (a) that the proposed ASS method outperforms random selection baselines in all sample granularity. Rec16 gives the best annotation mIoU with fewer clicks among all the granularities because it provides better sample diversity than larger samples. This diversity favors model fine-tuning when annotations are limited. As annotation clicks increase, the gap between all the settings becomes smaller, so we zoom in on the curves in this part in (b).

Annotating Frame surpasses others when the percentage of clicks is over 20%. The reason is that the sample diversity saturates with more manually annotated samples. In this stage, annotating more pixels keep improving final outputs quality by label propagation. Annotating frames obtains the most labeled pixels per click compared to smaller-sized samples, due to the effort to handle truncated object contours or the dividing objects merged by imperfect superpixels. As a result, a larger granularity annotation sample achieves higher label quality faster. To this end, we suggest users choose a proper sample granularity to annotate depending on their desired label quality.

Table 1. This table shows the most efficient sample granularity for different mIoU benchmarks in SceneNet. The last row represents manually annotating all the frames.

Annotation mIoU	Granularity	Anno. Clicks	Anno. Pixel
80%	Rec16	1.5%	0.2%
85%	Rec16	2.5%	0.3%
90%	Rec40	5.0%	1.4%
95%	Frame	27%	23%
99%	Frame	100%	100%

Click cost for label quality benchmarks. In Tab. 1 we list the least annotation clicks required to generate 80%, 85%, 90%, and 95% mIoU labels, and the corresponding sample granularity. The last row represents the manual labeling of the full video. Annotating Rec16 samples first achieves 80% and 85% mIoU, the annotation click cost is 1.5% and 2.5%. Rec40 first achieves 90% mIoU with 5% of annotation clicks. Annotating Frame first achieves 95% mIoU with 27% annotation clicks, which is over five times the clicks to achieve 90% mIoU. This observation shows the mIoU gain is sub-linear to the annotation clicks. However, it still saves 73% annotation effort compared to annotating the full video, demonstrating the proposed method generates very high-quality annotations while saving human effort significantly. It is worth mentioning that the pre-trained model performs poorly on testing sequences (Fig. 5), which shows the proposed framework can adapt to the target sequence by learning from selected samples and leveraging the temporal information.

Fig. 7 (c) shows the comparison under the traditional

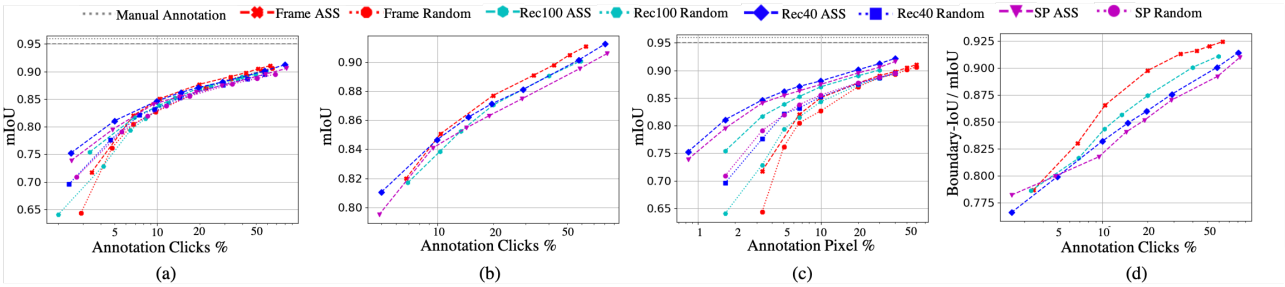


Figure 8. Active sample selection results on VEIS: (a),(b),(d) show the generated annotation mIoU and the normalized boundary-IoU in annotation clicks %, and (b) is a zoom-in version of (a). (c) shows the generated annotation mIoU in annotated pixel %.

Table 2. Comparison of the overall performance on SceneNet [29] with manual annotations selected by ASS method. Given the same information from annotated frames, our method outperforms the other two and shows advantages at lower annotation cost.

	2% clicks			4.6% clicks			7.1% clicks			9.3% clicks		
	mIoU	P-Acc.	TC	mIoU	P-Acc.	TC	mIoU	P-Acc.	TC	mIoU	P-Acc.	TC
Fine-tune only	45.72	76.57	61.98	64.73	88.90	76.86	70.76	91.76	81.20	81.31	94.72	87.09
LP [8]	48.57	76.46	66.09	59.5	85.06	84.68	68.2	87.66	86.97	76.34	91.41	86.66
Ours	**63.67**	**88.71**	**84.43**	**79.54**	**95.15**	**89.33**	**86.07**	**96.90**	**93.45**	**89.96**	**97.28**	**94.70**

pixel-based annotation cost measurement. The observation is very different from (a), as annotating Frame is always the worst. We believe that the traditional pixel-based cost measurement could be misleading in Segmentation AL tasks.

Comparison of boundary-IoU. Object boundary quality is crucial in segmentation annotations. In Fig. 7 (d), we show the boundary-Intersection-over-Union (boundary-IoU) [11] normalized by mIoU, which reflects the boundary annotation accuracy. Models trained on frame samples outperform the others with no exceptions. The reason is frame level annotation provides the richest semantic/boundary information. On the contrary, the super-pixel-based selection is usually composed of pixels of the same object, which lacks the information of the object boundaries. So its boundary prediction accuracy is the worst. For rectangle samples, larger granularity samples give better predictions on the boundary. If the user has high requirements on the label boundary quality, annotating whole frames is the best choice.

4.2.2 VEIS

For more extensive experiments, we conducted auto-annotation experiments on an outdoor-scene synthetic dataset VEIS [38]. It includes semantic segmentation ground-truth for every video frame with the object classes of standard real urban scene datasets, such as CamVid [6] and Cityscapes [14]. We randomly pick six video clips from the full VEIS sequence, each of which contains 200 frames. The pre-trained model is trained with Cityscapes training set from an ImageNet pre-trained checkpoint with mIoU of 32.56% on all the testing videos. We tested four granularity settings: Frame, Rec100, Rec40, and SP. As the resolution of VEIS frames is 600×800, Rec100 and Rec40 split a frame into 48 and 200 segments, respectively. We let SP split a frame into about 200 segments.

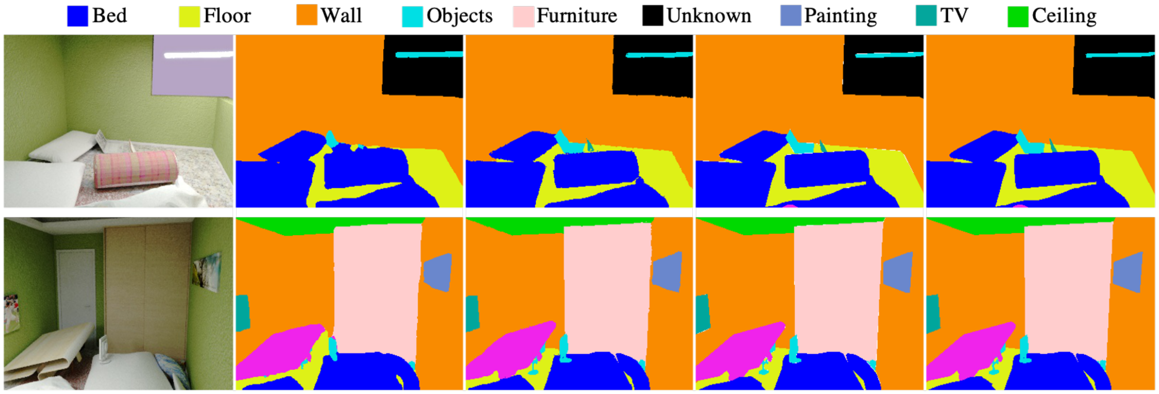
Fig. 8 (a) compares the generated label mIoU given annotation clicks, and (b) zooms in the high mIoU plots. The observations are very similar to the SceneNet results. First, the ASS method always outperforms random selection baselines. Second, larger granularity annotation samples achieve higher label quality faster. When annotation clicks are small, annotating Rec40 samples leads to the best-generated annotations. After the annotation cost in clicks is greater than 10%, annotating Frame outperforms all others.

Due to page limit, see Appendix C for click cost for label quality benchmarks and comparison of boundary-IoU results on VEIS dataset.

4.3. Analysis

Model uncertainty and selected samples. Fig. 4 illustrates how the proposed framework selects sample and learns from it. This VEIS example is of annotating Rec100 with about 3.3% annotation clicks. The first two rows are the model uncertainty and generated annotation after the first iteration. The ASS method selects a sample that are of high uncertainty and inferior prediction. The similar regions in other frames are not selected, as the proposed ASS considers both sample uncertainty and diversity. The last two rows show the model results after fine-tuning with the selected sample. The regions' annotation quality in all the neighbor frames is improved significantly, demonstrating the effectiveness of the test-time fine-tuning component.

Effectiveness of label propagation module. We compare the proposed test-time fine-tuning method with it's ablated version by removing temporal consistency loss (Fine-tune



(a) Raw (b) 90% mIoU (c) 95% mIoU (d) MA (e) GT

Figure 9. Visualization of our generated annotations in SceneNet RGB-D. (a) is the video frame, (b) costs about 5.0% clicks with annotating Rec40, and (c) costs about 27% clicks with annotating Frame. (d) is the mimic manual annotation, and (e) is the ground-truth.

only) and LP [8]. LP is a well known label propagation algorithm, which can be directly applied to new target domain videos to propagate sparse annotations. Here we use our ASS method to select manual annotated samples. Tab. 2 shows generated label's mIoU, pixel accuracy, and Temporal consistency (TC). TC measures the mIoU between two consecutive predictions similar to [26]. Given the same selected samples, our method outperforms the Fine-tune only and LP methods by a large margin in mIoU and TC at various annotation clicks percentages. The results prove the effectiveness of consistency loss and test-time fine-tuning method. The benefit is even more significant when the sample rates are lower, as our method incorporates both motion and semantic cues to the test sequences.

Impact of number of ASS iteration. We conduct experiments to understand the impact of the number of iterations to the segmentation quality on SceneNet RGB-D. We feed 0.3% clicks of annotations per iteration, and fine-tune the model up to nine iterations. The mIoU gains per iteration are 6.91%, 1.32%, 0.89%, 0.35%, 0.16%, 0.43%, 0.07%, 0.31%, and 0.03%. Starting from the fourth iteration, the mIoU gain becomes negligible. As a result, we use three iterations for ASS.

Error pattern in high quality generated annotations. We conduct experiments to investigate where the remaining errors are when the generated annotation is already of high quality. On SceneNet RGB-D, the 100% manual annotation mIoU is 98.56%. When our method achieves 97.46% mIoU, the boundary IoU is only 83.44%, indicating errors appear in the object boundaries. Categories with high boundary-to-area ratio have the largest impact from the imperfect boundary predictions. This can be reflected from their below-average per-class IoU. In SceneNet, they are ”Object”, ”Chair”, and ”Table”. In VEIS, they are ”Pole”, ”Traffic Light”, and ”Rider”. With the error pattern in mind, users could use the generated annotations more confidently.

Generated annotation visualization. In Fig. 9 we show our generated annotations in SceneNet. The 90% mIoU annotations in (b) only cost about 5.0% clicks; The 95% mIoU annotations in (c) cost about 27% clicks. See Appendix C for visualizations of generated annotations in VEIS.

Model computation time. The model computation time for one ASS iteration is mainly from sample selection and test-time training steps. The test time fine-tuning computation time depends on image resolution and video sequence length. For SceneNet RGB-D, a sequence of 300 frames with resolution 320 × 240 takes 20 minutes for one iteration on average. For VEIS, a sequence of 200 frames with resolution 800 × 600 takes 33.3 minutes for one iteration on average. Our experiments runs on 4×Nvidia 1080s. The 9 seconds sample selection CPU runtime can be neglected. The dozens of minutes computation time prevents the annotators from labeling the next batch of samples immediately. However, this can be easily mitigated by multitasking arrangements in practice.

5. Conclusion

We propose a human-in-the-loop framework HVSA to generate video semantic segmentation annotations. It actively selects annotation samples at each iteration that bring the most information for annotating. After selected samples get manual annotations, our method leverages both semantic knowledge and temporal constraints to fine-tune a video-specific semantic segmentation model. Finally, the model is used to generate annotations for the entire video. We conducte experiments on two datasets to show HVSA can generate close-to-perfect annotations at a low cost, even without good pre-trained networks. Each iteration of HVSA takes dozens of minutes, which can be further optimized using multi-task parallelization.

References

[1] Azad Abad, Moin Nabi, and Alessandro Moschitti. Autonomous crowdsourcing through human-machine collaborative learning. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2017.

[2] Vijay Badrinarayanan, Ignas Budvytis, and Roberto Cipolla. Semi-supervised video segmentation using tree structured graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(11):2751–2764, 2013.

[3] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[4] Bianca-Cerasela-Zelia Blaga and Sergiu Nedevschi. Semantic segmentation learning for autonomous uavs using simulators and real data. In *Conference on Intelligent Computer Communication and Processing (ICCP)*, 2019.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters (PRL)*, 30(2):88–97, 2009.

[7] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Label propagation in complex video sequences using semi-supervised learning. In *British Machine Vision Conference (BMVC)*, 2010.

[8] Ignas Budvytis, Patrick Sauer, Thomas Roddick, Kesar Breen, and Roberto Cipolla. Large scale labelled video data augmentation for semantic segmentation in driving scenarios. In *International Conference on Computer Vision Workshops (ICCVW)*, 2017.

[9] Lile Cai, Xun Xu, Jun Hao Liew, and Chuan Sheng Foo. Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[10] Arantxa Casanova, Pedro O Pinheiro, Negar Rostamzadeh, and Christopher J Pal. Reinforced active learning for image segmentation. In *International Conference on Learning Representations (ICLR)*, 2020.

[11] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C. Berg, and Alexander Kirillov. Boundary IoU: Improving object-centric image segmentation evaluation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[13] Pascal Colling, Lutz Roese-Koerner, Hanno Gottschalk, and Matthias Rottmann. Metabox+: a new region based active learning method for semantic segmentation using priority maps. In *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2021.

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] Chengliang Dai, Shuo Wang, Yuanhan Mo, Kaichen Zhou, Elsa Angelini, Yike Guo, and Wenjia Bai. Suggestive annotation of brain tumour images with gradient-guided sampling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2020.

[17] Alireza Fathi, Maria Florina Balcan, Xiaofeng Ren, and James M Rehg. Combining self training and active learning for video segmentation. In *British Machine Vision Conference (BMVC)*, 2011.

[18] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video cnns through representation warping. In *International Conference on Computer Vision (ICCV)*, 2017.

[19] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[20] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Interactive video object segmentation using global and local transfer modules. In *European Conference on Computer Vision (ECCV)*, 2020.

[21] Jin-Yu Huang and Jian-Jiun Ding. Generic image segmentation in fully convolutional networks by superpixel merging map. In *Asian Conference on Computer Vision (ACCV)*, 2020.

[22] Jingwei Ji, Shyamal Buch, Alvaro Soto, and Juan Carlos Niebles. End-to-end joint semantic segmentation

of actors and actions in video. In *European Conference on Computer Vision (ECCV)*, 2018.

[23] Tae-young Ko and Seung-ho Lee. Novel method of semantic segmentation applicable to augmented reality. *Sensors*, 20(6):1737, 2020.

[24] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision (ECCV)*, 2014.

[25] Chong Liu and Yu-Xiang Wang. Doubly robust crowdsourcing. *Journal of Artificial Intelligence Research (JAIR)*, 73:209–229, 2022.

[26] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. In *European Conference on Computer Vision (ECCV)*, 2020.

[27] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *International Conference on Computer Vision (ICCV)*, 2017.

[28] Radek Mackowiak, Philip Lenz, Omair Ghori, Ferran Diego, Oliver Lange, and Carsten Rother. Cereals - cost-effective region-based active learning for semantic segmentation. In *British Machine Vision Conference (BMVC)*, 2018.

[29] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *International Conference on Computer Vision (ICCV)*, 2017.

[30] Siva Karthik Mustikovela, Michael Ying Yang, and Carsten Rother. Can ground truth label propagation from video help semantic segmentation? In *European Conference on Computer Vision (ECCV)*, 2016.

[31] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, 2012.

[32] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[34] Mahdyar Ravanbakhsh, Tassilo Klein, Kayhan Batmanghelich, and Moin Nabi. Uncertainty-driven semantic segmentation through human-machine collaborative learning. In *Medical Imaging with Deep Learning (MIDL)*, 2019.

[35] Wenqi Ren, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *International Conference on Computer Vision (ICCV)*, 2017.

[36] Wenqi Ren, Jingang Zhang, Xiangyu Xu, Lin Ma, Xiaochun Cao, Gaofeng Meng, and Wei Liu. Deep video dehazing with semantic segmentation. *IEEE Transactions on Image Processing (TIP)*, 28(4):1895–1908, 2019.

[37] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747*, 2016.

[38] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[39] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2009.

[40] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. Viewal: Active learning with viewpoint entropy for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[41] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *International Conference on Computer Vision (ICCV)*, 2019.

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

[43] Sudheendra Vijayanarasimhan and Kristen Grauman. Active frame selection for label propagation in videos. In *European Conference on Computer Vision (ECCV)*, 2012.

[44] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 43(10):3349–3364, 2021.

[45] L. Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Ziyi Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International Conference on Medical*

Image Computing and Computer-Assisted Intervention (MICCAI), 2017.

[46] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.