Interactive Online Training and Prediction for Video Segmentation

Jun Ye
jun.ye@sv.cmu.edu
May 10, 2018

***Abstract*—The goal of this project is two-fold. First is to assess the effect of using optical flow information in the ClickBAIT-v2 algorithm. Secondly, beyond the first goal, we want to explore new approaches to what ClickBAIT-v2 algorithm does. The new approach is mainly based on One-Shot Video Object Segmentation(OSVOS) algorithm as a single unified system to do interactive online training and prediction. The end result show that optical flow doesn't enhance the segmentation performance much, only by about 1 percentage in mean Intersection-over-Union(mIoU) measurement. For the new OSVOS approach, we made some progress in adapting the algorithm to do online-training and prediction based on users' clicks, but the results are not as good as CickBAIT-v2. The main problem is trying to getting rid of presumed selection by the algorithm. Future work can be done to improve the current results.**

I. Introduction

In [1], it describes the problem that we want to solve which is in Time Ordered Online Training (ToOT) domain. Some applications requires the object detector to detect unseen objects for the algorithm in real-time. For example, an Unmanned Aerial Systems (UAS) is tracking a person. If the person changes clothes, the tracker would break down and lose the target. In this case, user assisted input can guild the model to train itself and then use it to keep tracking. Figure 1 shows the architecture of ClickBAIT-v2. It uses a modified segmenter based on [2] to take a user click and generate a bounding box as the training ground truth to the Single Shot Object Detector (SSD) [3]. It also adds an object tracker to automate the training process for objects in the sequent frames. If the object tracker can successfully find the object in the subsequent frames, the center of the tracker output is used as the user click and a training event is initiated. There are several areas that we can improve upon the current approach to the ToOT problem. ClickBAIT-v2 uses separate segmentation, detection, and tracking algorithms. This can pose processing burden and energy consumption on resource constraint embedded systems, like UAS. So maybe a unified system to do the whole job can be more energy efficient. Another problem came up in the ClickBAIT-v2 algorithm is that the output from the segmenter sometimes doesn't include the whole object of interest. For example, when the person of interest is waving his hands, the arms are sometimes not included in the produced bounding box because the segmentation doesn't include the arms due to its thin feature. To improve upon this, we plan to add optical flow information because it can improve the segmentation results as shown in [4]. It shows "an improvement of 4.8 percentage-points versus SegNet, the RGB-only segmentationmodel on which FlowSeg-A is based [4]" based on the CamVid dataset.

The paper is organized as follows. Section II describes the background of optical flow and segmentation algorithms. Section III describes the approach we took to achieve the two objectives mentioned in I. Section IV shows the findings from the results in Section III. Lastly, section V provides the conclusion and future work.

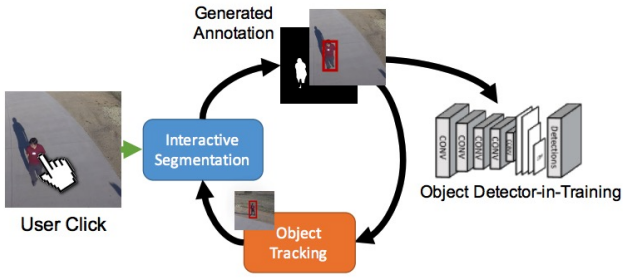


Fig. 1: ClickBAIT-v2 Architecture [1]

II. Related Work

Optical flow shows the motion of objects between frames. It has two categories, sparse optical flow and dense optical flow. Sparse optical flow shows the motion of pixels of some sparse features, like corners, but dense optical flow shows the motion of all pixels in the frame. The comparison is shown in [5]. There are three main datasets to evaluate the performance of different optical flow algorithms. They are Middlebury, MPI Sintel and KITTI datasets. All of them publish the algorithms' performance rank based on their evaluation datasets on websites [6] [7] [8] respectively. Middlebury has a very small dataset with only 8 pairs of training images with ground truth, but it includes different effects, like hidden texture and occlusions. Sintel dataset uses long sequences of synthetic graphics with features like large motion and motion blur. KITTI dataset is captured from driving in the city of Karlsruhe. From the performance ranking list, we choose the optical flow algorithms that produce good results but more importantly the run time needs to be short, less than a second, due to the ToOT nature. So we chose to compare FlowNet2 [9] and Farneback [10] algorithms. Farneback is based on polynomial expansion algorithm, and the FlowNet2 uses the combination of several neural nets.

There are two main categories of video segmentation, unsupervised and semi-supervised. Unsupervised algorithms

TABLE I: Run Time and Optical Flow Accuracy Comparison

Algorithm	Run Time(s)	Accuracy(aEPE)
Farneback	0.128	1.206
FlowNet2	0.680	0.445

TABLE II: Segmentation Results Comparison

Algorithm	Accuracy(Cross Entropy)	Accuracy(mIoU)
No Flow	0.231	0.702
Farneback	0.213	0.702
FlowNet2	0.259	0.713

TABLE III: Bounding Box Results Comparison

Algorithm	Accuracy(mIoU)
No Flow	0.508
Farneback	0.504
FlowNet2	0.522

need to decide what the main object is in each frame. Semi-supervised algorithms are given the ground truth segmentation in the first frame and need to produce segmentations in the following frames. There are two main approaches to video segmentation emerged in 2016. One is One-Shot Video Object Segmentation(OSVOS) [11] and the other is MaskTrack [12]. OSVOS takes each frame independently and uses a Fully Convolutional Neural Net(FCN) adapted from VGG-16 network. MaskTrack uses the previous frame's prediction and add some distortion to become the mask input for the next frame. As a result, OSVOS runs about 100 times faster than MaskTrack. "OSVOS is able to segment each 480p frame (480 x 854) in 102 ms [11]." "At test time our base MaskTrack system runs at about 12 seconds per frame (averaged over DAVIS, amortizing the online fine-tuning time over all video frames) [12]." For assessing video segmentation, the two well-known datasets are PASCAL Visual Object Classes(VOC) dataset and Densely Annotated Video Segmentation(DAVIS) dataset. DAVIS is the newer and bigger dataset and the annotation is higher quality. So we will use DAVIS for segmentation evaluation.

III. Approach

In order to improve the segmentation result in ClickBAIT-v2, we decided to add optical flow information. As mentioned in Section II, we chose Farneback and FlowNet2 algorithms mainly due to run time constraint. Their run time and accuracy result is shown in Table I. The results are computed on Middlebury training dataset with 8 image pairs running on a GTX 1080. The accuracy is computed as the average End Point Error(aEPE). The formula for aEPE is shown below:

$$aEPE = \frac{1}{N} * \sum_{i} \sqrt{(u_i - u_i^{GT})^2 + (v_i - v_i^{GT})^2}$$

N is the total number of pixels. u and v are the flow vectors in the x and y direction respectively.

From here, we added the flow information in the segmentation algorithm. Some modification has to be made to the original FlowNet2 algorithm in order to split tensor-flow initialization and session run process. Then DAVIS dataset's optical flows are computed using Farneback and FlowNet2. The original images are added three more channels, the click channel, flow's magnitude and flow's angle. The click channel is shown in Figure 3 in Section VI. The single click is transformed into a Euclidean distance map depending how far away from the click is. The new images are then converted to tf-records for validation set and along with ground truth masks for training set. Then the segmentation algorithm which is FCN-8s uses the tf-records to train and then the trained model is used to do prediction. The images are resized to 384x384 pixels for training and prediction process. Since the tf-records and FCN-8s training of no-flow and flow using Farneback method are done previously, we only need to create the tf-record for FlowNet2 and train a new model. The segmentation results are shown in Table II.

The segmentation accuracy is calculated based on cross entropy and mean Intersection over Union(mIoU). One example segmentation output and ground truth is shown in Figure 4 in Section VI.

Then bounding boxes are computed based on the segmentation. Because the ClickBAIT-v2 uses bounding boxes as ground truth to train the object detector, we also want to compare how much the bounding-box IoU result changes. The bounding boxes are computed by finding the closest segmentation area to the click point, which may not be the biggest segmentation area. If there is no segmentation output or the segmentation is smaller than the 20x20 pixels' area, a small bounding box of 20x20 pixels around the click point becomes the output. The three comparison segmentation models are all trained for 100 epochs on the DAVIS train dataset. The bounding box outputs corresponding to Figure 4 segmentation is shown in Figure 5. The bounding box mIoU results are shown in Table III.

From this point, we started to explore new segmentation algorithms and focused on OSVOS for reasons mentioned in Section II. OSVOS achieved a mean Jaccard index(J) of 79.8 [13] which is the same as mIoU. The formula for Jaccard is shown below:

$$J = \frac{|M \cap G|}{|M \cup G|}$$

M is the predicted segmentation and G is the ground truth mask.

Since OSVOS can find the same object across different frames, it inherently can keep track of the object. This algorithm can theoretically replace the ClickBAIT-v2 system which has an object detector, tracker, and segmenter. Currently the OSVOS algorithm takes a ground truth mask in the first frame, but we can't provide a ground truth mask in our application. The user can only provide clicks since it's hard to do more annotations in a video without pausing. Therefore we

TABLE IV: Training Process Comparison

Training Process	Accuracy(mIoU)
2000 iterations on GT	0.767
20 iterations on GT	0.741
20 iterations on Scribble	0.687
20 iterations on Scribble upper half	0.636
20 iterations on Scribble lower half	0.651
20 iterations on Single dot	0.600

TABLE V: Training Across Frames Results

Training Across Frames	Accuracy(mIoU)
20 iterations on Single dot	0.600
20 iterations more on the 10th frame	0.641
20 iterations more on the 20th frame	0.657
60 iterations on Single dot	0.651

need to access how much effect on the segmentation accuracy by reducing ground truth quality and fine-tuning time. The result of testing on Bmx-trees DAVIS sequence is shown in Table IV.

The scribble is the yellow line as shown in Figure2. Scribble upper half and lower half refers to the upper half and lower half of the line with a width of 10 pixels. The circle is the red dot in the middle of the line with a radius of 10 pixels. These scribble and dot annotations are provided as separate training inputs. The image shown here with all the annotations is for illustration purpose only.

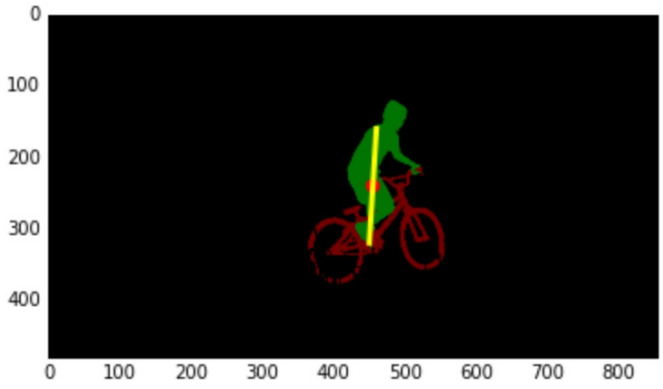


Fig. 2: Scribbles and Dot Training Input

From Table IV, we see that the dot as ground truth doesn't produce as good result as a fully annotated mask. We want to see how much accuracy improvement we can get when we do training across frames. The result of training on Bmx-trees DAVIS sequence is shown in Table V.

After seeing that OSVOS can take a dot as ground truth and we can get accuracy improvement from training across frames, we start to modify the OSVOS algorithm to build an interactive system to do online training and prediction. Basically, when a user clicks on an object in a video frame, that click initiates a training cycle. Then the later prediction will produce a better result due to the additional training. The result of testing on Bmx-trees DAVIS sequence is shown in Table VI. The IoU values is from the last frame which demonstrates the cumulated effect of 12 clicks throughout the video frames.

TABLE VI: Training Clicks Improvement

Training Process	Accuracy(IoU)
12 clicks	0.669
Without clicks	0.609

IV. Analysis

As we can see from Table I, the FlowNet2 is about 3 times better than Farneback in terms of accuracy. But the run time is about 5 times slower than Farneback. However, it's still within a second which is tolerable.

As we can see from Table II and Table III, comparing to no flow information, the flow doesn't improve the result much. There is only about 1 percent segmentation and bounding box improvement for FlowNet2 and no improvement for Farneback. The bounding box mIoU result is lower than the segmentation IoU is mainly because of the bounding box selection algorithm. The segmentation IoU computation considers the whole segmentation output but the bounding box selects the segmentation area that's closest to the click as explained in Section III.

Table IV shows the effect of reducing ground truth quality and training iterations. Decreasing from 2000 fine-tuning iterations to 20 iterations doesn't impact the accuracy much, only 2 percent. Changing from ground truth to a scribble line impacts the accuracy by about 5 percent. By reducing the amount of scribble to be half the length decreases the accuracy by about 4 percent. Lastly, reducing the input to be a single dot results a decrease of additional 4 percent.

Table V shows how much accuracy we can get back by training for multiple iterations on a single dot as ground truth. From training 20 iterations to 40 iterations, the accuracy increased by about 4 percent. Then additional 20 iterations on a different frame increases the accuracy by about 2 percent. It's similar to training on the same frame for a total of 60 iterations. Table VI shows the result of the interactive system. By 12 clicks which initiates 12 iterations of training on different frames produce an increase of about 7 percent compared to no clicks.

Even though Table VI shows the segmentation improvement, but it's actually not directly applicable to what ClickBAIT-v2 tries to achieve. First of all, ClickBAIT-v2 is not going for the accurate segmentation information, but only the bounding box. The main reason that the OSVOS algorithm's segmentation result improves is because it can recognize the main object. For common objects and especially objects in the DAVIS training dataset, the algorithm can get the segmentation pretty well. However, this has a downside, when the object of interest is not the common object. It's hard to get rid of the selection of the common objects presented in the frame.

Another problem came up is that the algorithm tends to add extra areas of segmentation, false positive areas. This can lead to a bigger bounding box area.

V. Conclusion

In conclusion, this paper shows that adding optical flow information doesn't help the ClickBAIT-v2 algorithm much, because there is only about 1 percent improvement in the segmentation result by adding optical flow information. In the exploration of new approaches, the interactive OSVOS presents some promises of replacing ClickBAIT-v2. The segmentation results improve as more clicks are provided. However, it also poses problems when the object of interest is not a common object. It's hard to get rid of the common object selection. Further improvements need to be made in order to use the interactive OSVOS approach. Some possible future work is summarized below.

Future Work

To improve upon the interactive OSVOS approach, a way of getting rid of some selection area needs to be added. This may require the network to take in an addition channel of input as the negative click channel. To improve the segmentation result, the ClickBAIT-v2 segmentation approach can be used. The segmenter only output segmentation area where the click lands, and thus getting rid of false positives.

References

[1] E. Teng, R. Huang, and B. Iannucci, "ClickBAIT-v2: Training an Object Detector in Real-Time," *arxiv*, 27 March 2018.

[2] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang., "Deep Interactive Object Selection," *CVPR, pages 373âĂŞ381*, 2016.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," *ECCV*, 2016.

[4] J. Gorgen, "Using Optical Flow to Improve Semantic Video Segmentation," *escholarship*, 2017.

[5] "OpenCV Optical Flow Algorithms," https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_lucas_kanade.html, accessed: 2018-05-01.

[6] "Middlebury Optical Flow Ranking," http://vision.middlebury.edu/flow/eval/results/results-e1.php, accessed: 2018-05-01.

[7] "MPI Sintel Optical Flow Ranking," http://sintel.is.tue.mpg.de/quant?metric_id=0&selected_pass=0, accessed: 2018-05-01.

[8] "KITTI Optical Flow Ranking," http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow, accessed: 2018-05-01.

[9] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," *CVPR*, 2017.

[10] G. Farneback, "Two-Frame Motion Estimation Based on Polynomial Expansion," *Springer*, 24 June 2013.

[11] S. Caelles, K. Maninis, J. Pont-Tuset, L. Leal-TaixÃl', D. Cremers, and L. V. Gool, "One-Shot Video Object Segmentation," *CVPR*, 2017.

[12] F. Perazzi1, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning Video Object Segmentation from Static Images," *CVPR*, 2017.

[13] "DAVIS 2016 Benchmark Results," http://davischallenge.org/davis2016/soa_compare.html, accessed: 2018-05-01.

VI. Appendix

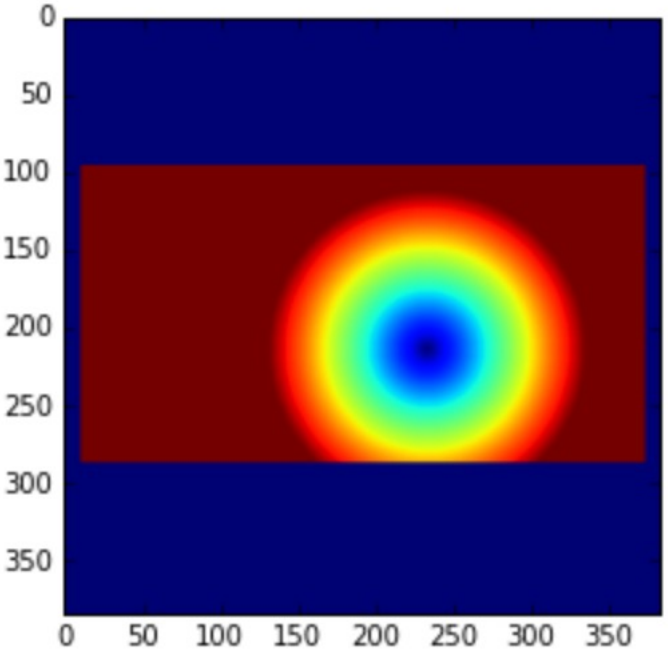
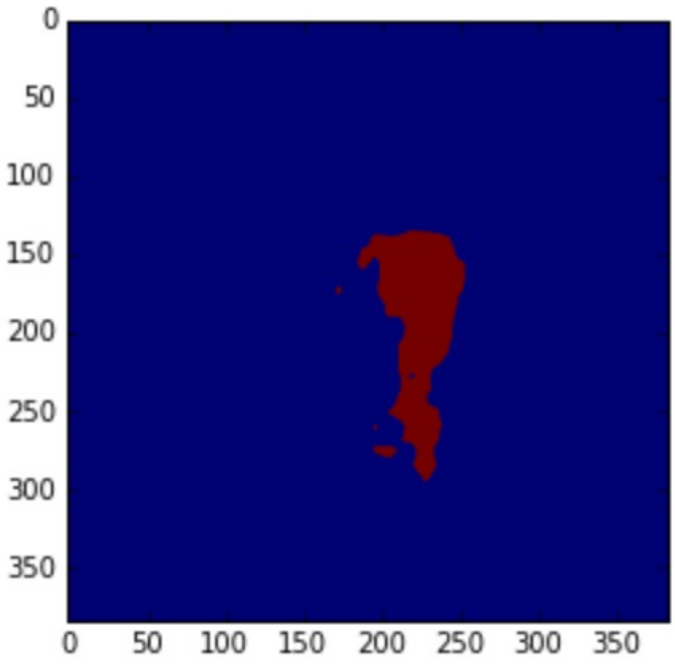


Fig. 3: Click Channel



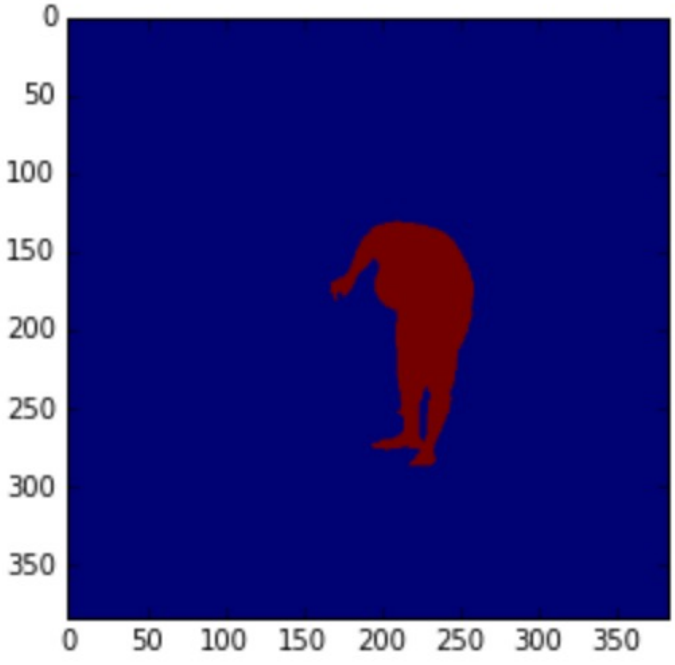
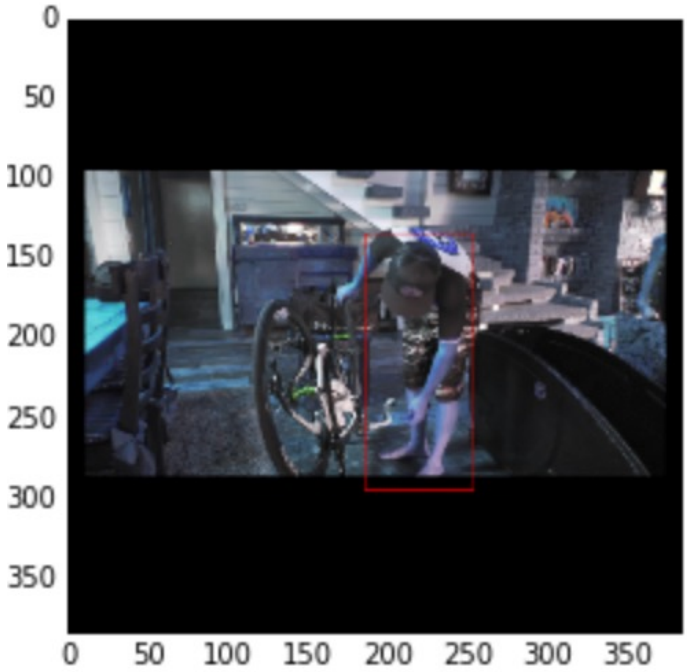


Fig. 4: Segmentation Output and Ground Truth



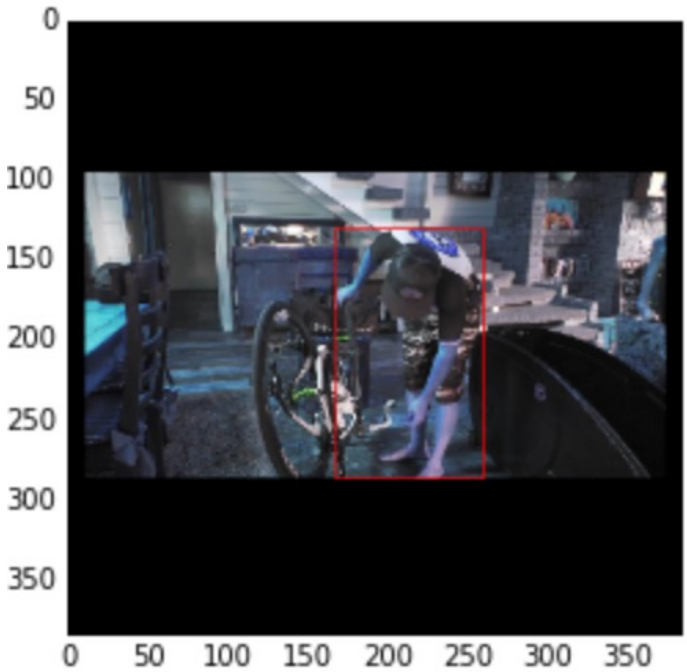


Fig. 5: Bounding Box Output