Supplementary Materials for : Multiclass Semantic Video Segmentation with Object-level Active Inference

Buyu Liu
ANU/NICTA
buyu.liu@anu.edu.au

Xuming He
NICTA/ANU
xuming.he@nicta.com.au

This supplementary material includes additional details on three aspects of our work:

(i) The supervoxel dense CRF model, including the kernels in the pairwise term and the mean-field equations.
(ii) The full pipeline of object trajectory proposal generation.
(iii) Benchmark datasets and experimental results on time complexity and model scalability.

Some examples of our video parsing result are also attached in the zip file.

1. Dense CRF details

1.1. Pairwise term in the supervoxel dense CRF

The mathematical form of our dense pairwise term is defined as follows :

$$\begin{aligned}\psi_v(l_i, l_j) =& \alpha_{s_1} \exp(-\frac{\|p_i - p_j\|^2}{2\theta_\gamma^2}) \\ &+ \alpha_{s_2} \exp(-\frac{\|p_i - p_j\|^2}{2\theta_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\theta_\beta^2})\end{aligned} \tag{1}$$

where I_i is the averaged color feature for supervoxel i in CIE-LAB space and p_i is the central position of supervoxel i in spatio-temporal domain. $\theta_\gamma, \theta_\alpha, \theta_\beta$ are widths of Gaussian kernels, and α_{s_1} and α_{s_2} are weighting coefficients.

1.2. Mean-field updating equations

We introduce the updating equation for supervoxel node in Eq (9). The rest of the updating equations for object and their relation nodes are derived as follows :

$$\begin{aligned}\hat{q}_o(d_m) \propto & [\![m \in \mathcal{S}]\!] \sum_{i \in o_m} \langle \psi_s(d_m, l_i) \rangle_{q_v(l_j)} \\ &+ \sum_{p \in \mathcal{P}} [\![m \in p]\!] \Big(\sum_{i \in o_m} \langle \psi_c(h_p, l_i, d_m) \rangle_{q_v(l_i) q_p(h_p)} \\ &+ \langle \psi_p(h_p, d_m, d_n) \rangle_{q_o(d_n) q_p(h_p)} \Big) + \phi_o(d_m)\end{aligned} \tag{2}$$

$$\begin{aligned}\hat{q}_p(h_p) \propto & \sum_{k \in p} \sum_{i \in o_k} \langle \psi_c(h_p, l_i, d_k) \rangle_{q_o(d_k) q_v(l_i)} \\ &+ \langle \psi_p(h_p, d_m, d_n) \rangle_{q_o(d_m) q_o(d_n)} + \phi_p(h_p)\end{aligned} \tag{3}$$

where $\langle \cdot \rangle_q$ denote the expectation with respect to q and $p = \{m, n\}$. Note that these two terms are summed over sparse connections, which can also be efficiently computed when the number of object and relation nodes is moderate.

2. Object proposal generation

We generate the object trajectory hypotheses in the following three steps, similar to [2].

1. We detect object instances and generate their masks in a sparse set of key frames based on an exemplar SVM [3] detector. We use only a small number of examples (10–20).

2. We propagate the static proposals to the entire video chunk in both forward and backward way. We compute an affine transformation of each object mask for neighboring frame pairs based on the dense pixel trajectories from [5]. A non-maximum suppression is then applied to remove redundant proposals from inaccurate trajectories based on match scores of object mask and intensity edges.

3. We extend the propagated object masks to longer object trajectories. We construct a directed graph on the object proposals from all the frames, in which an edge connects two proposals if they are from consecutive frames, share the same category and are significantly overlapped. The edges follow the time direction. Given the directed graph, we use depth-first search to generate all possible paths starting from those earliest static proposals.

3. More details on experiment

3.1. Dataset summary

We test our model on three video segmentation datasets.

Class	Car			Pedestrian			Bicyclists		
Threshold	-1	-0.9	-0.85	-0.95	-0.9	-0.85	-0.95	-0.9	-0.85
Precision	17.9	76.4	88.4	5.8	7.7	17.8	4.1	10.8	22.4
Recall	66.5	63.2	55.8	65.0	55.4	28.9	70	58.3	41.4

Table 1: Pixe-level precision/recall rate on CamVid test set in three foreground object classes. The recall rate of $Pedestrian$ and $Bicyclists$ increase significantly with lower thresholds, which may lead to more accuracy improvement in these two classes.

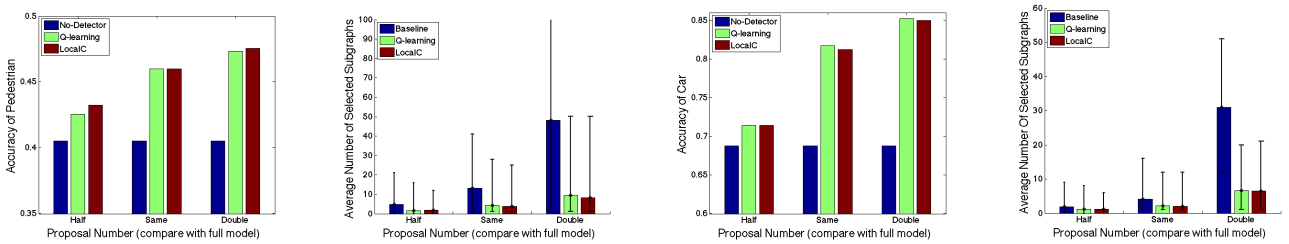

Figure 1: Scalibility with more proposals from a single class. Left: Pedestrian; Right: Car. Our proposed methods benefits from richer proposal pools while selecting much fewer proposals.

- **CamVid** [1] consists of 5 video sequences captured during the daytime and dusk. These sequences are sparsely labelled at 1Hz with 11 semantic classes. We follow the data split in [1] for training and test.
- **MPIScene** [6] consists of one sequence with 156 annotated frames and 5 semantic classes. We follow the set-up in [4].
- **DynamicScene** [7] consists of 176 sequences with 11 successive image frames each, and the last frame of each sequence is labelled with 8 classes. We test our model on the same test set as [7].

3.2. Scalability results on other classes

We use the same setting as in *Bicyclist* class, in which we lower the thresholds of object detectors to generate more proposals. We summarize the precision/recall rate in Table 1 on CamVid test set.

Figure 1 shows the results for class $Pedstrian$ and class Car. We can observe the same trend as in class $Bycyclists$. Note that as the precision/recall does not change much under the pre-defined thresholds in Car, we test our methods with more proposals under threshold of $(-1, -0.9, -0.85)$ instead. The improvement shown in the right panel of Figure 1 is similar to the other two classes.

3.3. Runtime complexity

We compare the average inference time in second (s) for video segments in Camvid with different lengths, including 61, 121 and 241 frames, in Table 2. Here we apply Q-learning method to obtain the results but the LocalC method has similar runtime efficiency. We can see that Q-learning is more efficient than the full model.

	Full Model		Our Method	
# of Frames	# of Subgraphs	Time(s)	# of Subgraphs	Time(s)
61	21.6	2.6	6.7	1.5
121	40.9	7.8	10.4	3.0
241	81.6	15.3	18.2	5.4

Table 2: Average number of subgraphs and running time to perform inference on different lengths of videos in CamVid.

3.4. Results on other dataset

On MPIScene dataset, we use 10 exemplars for $Vechile$ detector, which is applied every 10 frames. On DynamicScene dataset, we annotate 10 exemplars to train exemplarSVM detector for Car and apply detector to the first and last frame for each chunk.

We show the active inference result on the DynamicScene dataset in Figure 2. We can see that our method can achieve better performance with only one-third of the proposals.

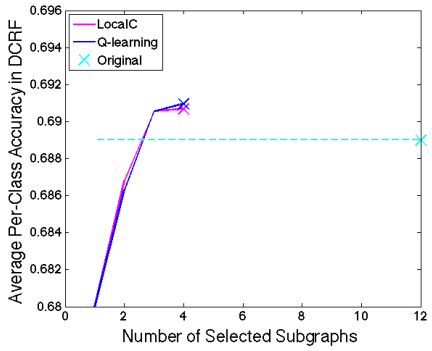

Figure 2: Active inference results on DynamicScene dataset. The cross signs show when the algorithm stops.

References

[1] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008. 2

[2] B. Liu, X. He, and S. Gould. Multi-class semantic video segmentation with exemplar-based object reasoning. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, 2015. 1

[3] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 1

[4] O. Miksik, D. Munoz, J. A. Bagnell, and M. Hebert. Efficient temporal consistency for streaming video scene analysis. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013. 2

[5] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 1

[6] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV*, 2010. 2

[7] C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV*, 2008. 2