Supplement for SMARTADAPT: Multi-branch Object Detection Framework for Videos on Mobiles

Ran Xu[1], Fangzhou Mu[2], Jayoung Lee[1], Preeti Mukherjee[1],
Somali Chaterji[1], Saurabh Bagchi[1], Yin Li[2]
[1]Purdue University [2]University of Wisconsin-Madison
[1]{xu943, lee3716, mukher57, schaterji, sbagchi}@purdue.edu [2]{fmu2, yin.li}@wisc.edu

1. Implementation

(a) ED+MB

di	*dv*	*rt*	*ct*
1,2,4,8,20,100	D0,D3	100%,50%,25%	0.15,0.3

(b) YL+MB

di	*rd*	*rt*	*tv*
1,2,4,8, 20,50, 100	224,256,288,320, 352,384,416,448 480,512,544,576	100%*, 50%*, 25%	MedianFlow, KCF, CSRT, Optical Flow.

(c) SSD+MB

di	*rd*	*rt*	*ct*
1,2,4,8,20,100	192,256,320	100%,50%,25%	0.15,0.3

Table 1. Choices of the tuning knobs in the MBODF with EfficientDet (ED), YOLOv3 (YL), and SSD object detectors (* indicates that it can only support the MedianFlow object tracker). Notations are: di for the detector interval, dv for the variant of the detector, tv for the variant of the tracker, rd for the input resolution of the detector, rt for the input resolution of the tracker, ct for the confidence threshold of objects to be tracked.

We implement MBODFs for four object detectors—Faster R-CNN [8], EfficientDet [10], YOLOv3 [7], and SSD [5]. The implementation details of Faster R-CNN are in Table 2 of the main paper, and we include the implementation details of the other three object detectors here in Table 1. Note that we allow two object detector variants (dv)—EfficientDet D0 and D3, for ED+MB, and allow four object trackers—MedianFlow [4], KCF [3], CSRT [6], and dense Optical Flow [2], for YL+MB.

2. Supporting Experiments

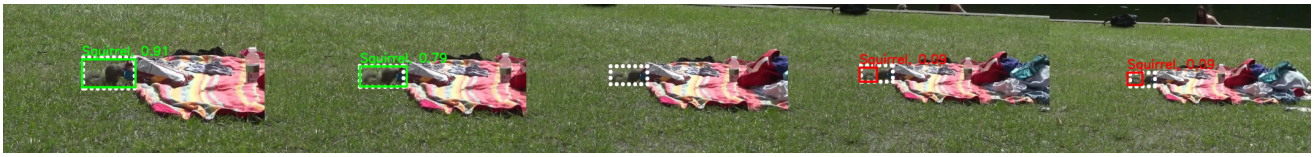
2.1. Content-Aware Scheduler (CAS)

We study FastAdapt as a baseline on how a content-aware scheduler (FastAdapt+CAS) improves such off-the-shelf adaptive object detection systems. Table 2 shows light content features, though coming with only 4 values, are better than the content-agnostic baseline by 0.5% mAP. However, directly applying off-the-shelf content features does not always yield accuracy improvements—when we train accuracy predictors on all branches, HoC, HOG, ResNet-50, MobileNet, and even trainable MobileNet's results are all neutral or negative, compared to the light features' results. Only the scheduler with the CPoP feature extractor achieves 0.5% mAP higher results than that with the light features, owing to these features extracted from the last layers of the object detector. Next, we narrow down the number of candidate branches to 200 (from 1,036 available ones), according to the scheme described in the main paper (Section 3.4, paragraph titled "Candidate Branches"). In brief, we consider a subset of branches, top-K, which covers a large fraction of the optimal branches. With this reduced number of branches, we observe that a better performing setting is FastAdapt+CAS with the jointly trainable MobileNet feature extractor and accuracy predictor, with 1.4% higher mAP than the baseline. Reducing the number of branches can be considered as imposing a regularizer that drives the model away from noisy branches, *i.e.* those which just happen to be optimal for one (or a few) videos.

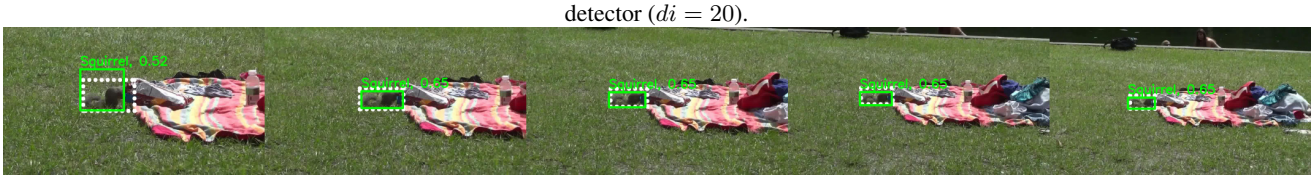
	light	HoC	HOG	Resnet50	CPoP	MobileNet	MobileNet*	MobileNet$^+$
Content-agnostic	43.8%							
All br.	44.3%	44.4%	44.3%	44.4%	44.8%	44.3%	44.0%	45.7%
200 br.	43.8%	44.1%	44.3%	44.1%	44.1%	43.8%	45.2%	**46.1%**

Table 2. Ablation study for the components in the CAS, over a FastAdapt baseline (content-agnostic) on the ILSVRC 2015 VID validation dataset, at a real-time 33.3 msec latency constraint (30 frames/sec video quality). * denotes the CAS with the jointly trainable feature extractor and the content-aware accuracy predictor. $^+$ denotes the CAS with joint modeling of content and latency requirement.

Finally, we further add the joint modeling of both content and latency requirements and reach an mAP of 46.1%, which is 2.3% mAP higher than the content-agnostic baseline. To summarize, it is hard to achieve an accuracy gain by simply plugging in an off-the-shelf feature extractor. Further, a smaller subset of branches using our technique can make the entire model easier to train and converge. This subsetting must be done carefully, ensuring (with high likelihood) that branches that appear on the Pareto optimal



(a) Faster R-CNN + Multi-branch, or FR+MB (content-agnostic) chooses a branch of $di = 20$, $rd = 288$, $nprop = 100$, $rt = 25\%$, and $ct = 0.05$, given a 20 msec latency constraint. The protocol misses the squirrel on the third frame and localizes the squirrel wrongly on the fourth and fifth frames due to the small resolution fed into the object detector ($rd = 288$) and too frequent calibration with the object detector ($di = 20$).



(b) Faster R-CNN + Multi-branch + our Content Aware Scheduler, or FR+MB+CAS chooses a branch of $di = 50$, $rd = 384$, $nprop = 100$, $rt = 25\%$, and $ct = 0.05$, given a 20 msec latency constraint. The protocol detects the squirrel correctly on all five frames, owing to the large resolution fed into the object detector ($rd = 384$) and using the object tracker on more frames ($di = 50$).

(c) Faster R-CNN + Multi-branch, or FR+MB (content-agnostic) chooses a branch of $di = 8$, $rd = 288$, $nprop = 100$, $rt = 25\%$, and $ct = 0.10$, given a 33.3 msec latency constraint. The protocol localizes the snake on the third and the fourth frames wrongly due to the small resolution fed into the object detector ($rd = 288$).

(d) Faster R-CNN + Multi-branch + our Content Aware Scheduler, or FR+MB+CAS chooses a branch of $di = 20$, $rd = 384$, $nprop = 100$, $rt = 25\%$, and $ct = 0.05$, given a 33.3 msec latency constraint. The protocol detects the snake correctly on all five frames, owing to the large resolution fed into the object detector ($rd = 384$) and using the object tracker on more frames ($di = 20$).

Figure 1. Qualitative results on comparing the content-agnostic FR+MB and content-aware FR+MB+CAS on two videos, one with a small squirrel, and the second, with a still snake. The first frame out of every 10 frames is visualized. The white boxes (dashed lines) show the ground truth annotations, the red boxes denote false positive boxes, and the green boxes denote true positive boxes. Predicted class labels and confidence scores are displayed on top of the detection boxes.

curve are not left out. A MobileNet feature extractor with the joint modeling of both content and latency requirements yields the best content-aware results. This aspect of joint training of the feature extractor gives another dimension of performance improvement of content-aware models on top of the content-agnostic performance (FastAdapt) (see Figure 6 of the main paper).

2.2. Visualization

In Figure 1(a) and (b), we show a visualization to demonstrate the benefits of CAS over the MBODF with Faster R-CNN (FR+MB). In this example, the latency constraint is strict, only 20 msec per frame, and the scheduler has to either choose a larger di (less frequent invocation of the detector and correspondingly, more frames on the object tracker), a smaller resolution of the object detector rd, or more efficient choices on other knobs. The content-agnostic protocol FR+MB chooses a smaller resolution of the object detector to meet the latency constraint. This branch ($di = 20$, $rd = 288$, $nprop = 100$, $rt = 25\%$, and $ct = 0.05$) results in missing detections and wrong localizations for this video due to the small size of the object (small squirrel). In contrast, FR+MB+CAS, being content-aware, smartly chooses a branch with a larger di and a larger rd. Such a branch allows precise calibrations on the frames with the object detector since the input resolution is higher ($rd = 384$) and

maintains correct detections with the object tracker over a longer GoF. We show another example in Figure 1(c) and (d) with a video of a still snake. The snake, with its camouflage, is localized wrongly by the FR+MB baselines in the third and the fourth frames due to the small resolution of the object detector ($rd = 288$). In contrast, FR+MB+CAS, being content-aware, detects the snake correctly on all frames with a larger di and a larger rd. To summarize, given the MBODF with thousands of execution branches, CAS is able to choose a more accurate branch that is best adapted to the content and subject to the latency constraint.

2.3. Post-processing Techniques

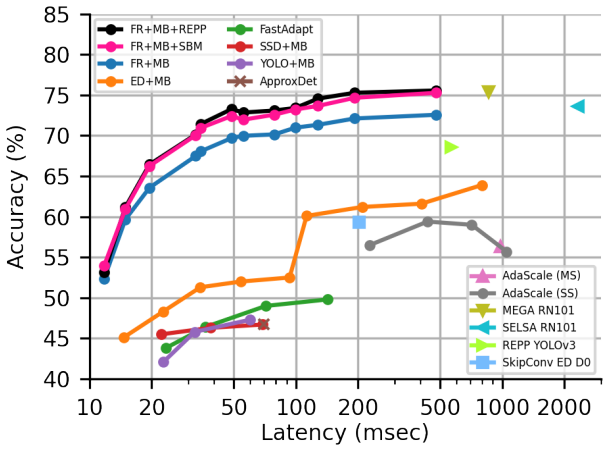


Figure 2. Accuracy-latency frontiers with post-processing techniques in the offline mode.

Several post-processing techniques have been developed for offline video object detection. The key idea behind these techniques is to re-score and link frame-level detection results. In the offline mode, REPP [9] and Seq-BBox Matching (SBM) [1] improve over FR+MB by 2.60% and 2.38% on average (averaged over all latency constraints). The results of the accuracy-latency Pareto frontier branches are shown in Figure 2 (FR+MB+REPP and FR+MB+SBM). We observe that the post-processing techniques yield a larger accuracy improvement for the branches with higher latency. This is because those branches perform much more object detection than object tracking (di is smaller) and the detection results from the object detector can benefit more than those from the object tracker due to the nature of the re-scoring techniques in REPP and SBM.

To study how the post-processing can benefit FR+MB+CAS in the offline mode, we compare the accuracy in Table 3 at stringent latency constraints. The results show that both REPP and SBM can benefit FR+MB+CAS, which is our best performing protocol. Further, REPP is slightly better than SBM and makes FR+MB+CAS+REPP the best performing protocol with post-processing techniques in the offline mode.

Protocols	20.0 ms	33.3 ms	50 ms	100 ms
FR+MB+CAS+REPP	**66.8%**	**70.8%**	**73.4%**	**74.1%**
FR+MB+CAS+SBM	66.7%	70.7%	72.5%	73.8%
FR+MB+REPP	66.4%	70.1%	73.3%	73.4%
FR+MB+SBM	66.2%	70.0%	72.4%	73.2%
FR+MB+CAS	64.1%	68.3%	69.8%	71.1%
FR+MB	63.6%	67.5%	69.7%	71.0%

Table 3. Accuracy comparison of FR+MB and FR+MB+CAS without post-processing techniques, with SBM, and with REPP post-processing techniques, given stringent latency constraints on the ILSVRC VID validation dataset.

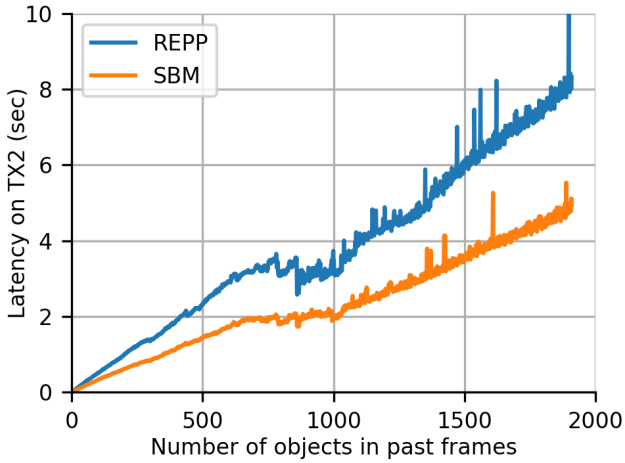


Figure 3. Latency of REPP and SBM in the online mode by the number of objects to post-process in the past frames, measured on the NVIDIA TX2 embedded board.

We empirically find the latency cost of REPP and SBM is significant larger in the online mode—277X and 385X larger than those in the offline mode, averaged on the entire dataset and measured on the TX2 board. Figure 3 uncovers the root cause of such large latency cost by showing the latency per frame vs. the number of objects in past frames. The basic version of post-processing considers all prior frames when doing post processing of the last frame. Hence, when considering later frames of a video, the total number of objects that need to be considered is cumulative and grows. We observe that as the number of objects accumulates in the later frames of a video, the latency to post-process the results increases linearly and reaches 8 seconds per frame when post-processing 1,900 objects in the past frames. Moreover, running post-processing for *every* frame in the online mode instead of doing once for the entire video (as is done in the offline mode) is another cause of such high latency cost. On the other hand, we also find that the accuracy improvement of the post-processing techniques diminishes in the online mode as the technique has to operate in streaming mode, *i.e.* it cannot go back and refine the detection results of the past frames and it cannot use

the detection results from the future frames. For example, the accuracy is 0.12%-0.61% mAP *lower* than that without post-processing technique in the SBM case. Thus, we conclude that post-processing iteratively for every frame, even for every N frames, is not acceptable for our model in the online mode due to the significant latency cost and diminishing accuracy improvements.

2.4. Offline Profiling Cost

The cost of profiling MBODF to realize an Oracle scheduler and to derive a snippet-granularity dataset to study content-aware accuracy is significant. Considering the 3,942-branch MBODF on top of the FRCNN object detector, in the basic case, we need to run every branch on the training and test datasets to collect its latency and accuracy. We deploy a set of engineering techniques to speed up the profiling. We use the following techniques to reduce the cost—(1) accuracy and latency profiling can be done separately, the former on the server, and the latter, on the embedded device but on a smaller set of videos since the latency does not vary significantly across videos for a given branch, (2) we profile the detector-only branches first and reuse the object detection results for other execution branches with $di > 1$, (3) the profiling of multiple execution branches can trivially be done in parallel and distributed in multiple servers. Combining these techniques, we are able to finish the profiling within 5 days on our two servers (specs in Section 4 of main paper).

References

[1] Hatem Belhassen, Heng Zhang, Virginie Fresse, and El-Bay Bourennane. Improving video object detection by seq-bbox matching. In *VISIGRAPP (5: VISAPP)*, pages 226–233, 2019. 3

[2] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proceedings of Scandinavian Conference on Image Analysis*, pages 363–370, 2003. 1

[3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2014. 1

[4] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pages 2756–2759. IEEE, 2010. 1

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9907, pages 21–37, 2016. 1

[6] Alan Lukežič, Tom'aš Voj'iř, Luka Čehovin Zajc, Jiř'i Matas, and Matej Kristan. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, 126:671–688, 2018. 1

[7] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, pages 1–6, 2018. 1

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 1

[9] Alberto Sabater, Luis Montesano, and Ana C Murillo. Robust and efficient post-processing for video object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10536–10542. IEEE, 2020. 3

[10] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10781–10790, 2020. 1