Clockwork Convnets for Video Semantic Segmentation

Anonymous ECCV submission

Paper ID 449

This supplementary material accompanies submission 449. In our submission we presented Clockwork FCN, an algorithm for scheduling computation in a fully convolutional network for semantic segmentation. We propose a technique for adaptively updating the clock schedule based on the input video and also explore a pipeline schedule to incorporate asynchronous updates from various layers in the convolutional network. In this document and the accompanying videos, we provide further qualitative examples from the Youtube-Objects dataset.

1 Pipeling Scheduling Video

We provide an additional qualitative example of the pipeline schedule in the attached video "pipeline.mp4." The video simultaneously shows in order from left to right: original video, per frame oracle, pipelined (ours), and a skip frame alternative that computes the full FCN on every other frame. (Note this is not a fair baseline since it does not have the same reduced latency as the pipeline. However it is a useful comparison to see how the skip layers in the pipeline schedule correctly update fine spatial details for the current frame.) The sequences of frames presented are illustrative examples of those for which our method performs better than the skip frame baseline, as measured by mean IU using the per frame oracle as ground truth. The sequences are drawn from a random sample of 40 shots from videos in the horse and car categories. The four example sequences shown demonstrate some properties of our approach. Fine local details are correct in real time - note the the accurate positions of the horse's legs in the first sequence, and the correct occlusion of the car in the second sequence. Boundaries of objects are correctly updated at each frame - note the car growing larger at every frame as it moves towards the camera in the third sequence. Entrance and exit of objects is detected immediately - note the segmentation tracking the car as it drives off screen in the fourth sequence.

2 Adaptive Clockwork Results

We present additional adaptive clockwork results in Figure 1, where we show three further examples. For these results, we use the threshold $\theta = 0.25$ as it represents a favorable tradeoff between accuracy and computation. In general, the threshold can be tuned to a particular application and dataset. The *top row* demonstrates the performance of the adaptive clock on a video with lots of motion. The full network is updated relatively frequently (71.6% of the time) with an almost negligible drop in accuracy (62.9% vs 63.0%). The *middle row* shows an example where the only object in the scene, a dog, moves quickly but generally remains in the same part of the frame: we fully update only

32.3% of the frames causing a drop of 1.4% in accuracy. Finally in the *bottom row*, we show an extreme example where the full network is updated on only 2.4% of frames. As expected, this results in a significant drop in accuracy (57.8% vs. the oracle 90.5%), however, remarkably, updating only the lower layers of the network still results in updated segmentations that follow the train throughout the video while benefiting from a dramatic reduction in computation.

Lastly, we show our clockwork FCN algorithm on a full video in "adaptive.mp4." Again we use the threshold $\theta = 0.25$. This video has relatively little motion in the first half, and more motion in the second half as the camera pans. Below the video clip (original video overlaid with our segmentation) we visualize the firings of the adaptive clock as the video progresses - the blue vertical bar corresponds to the current frame while the black vertical bars correspond to the clock firing. During the static first half of the video, the clock fires rarely, while during the period of high motion, the adaptive clock fires more often to better track the bird as it walks. On this video, we run the full FCN on only 8.5% of the frames and achieve a mean IU of 69.4 (oracle mean IU is 75.3).

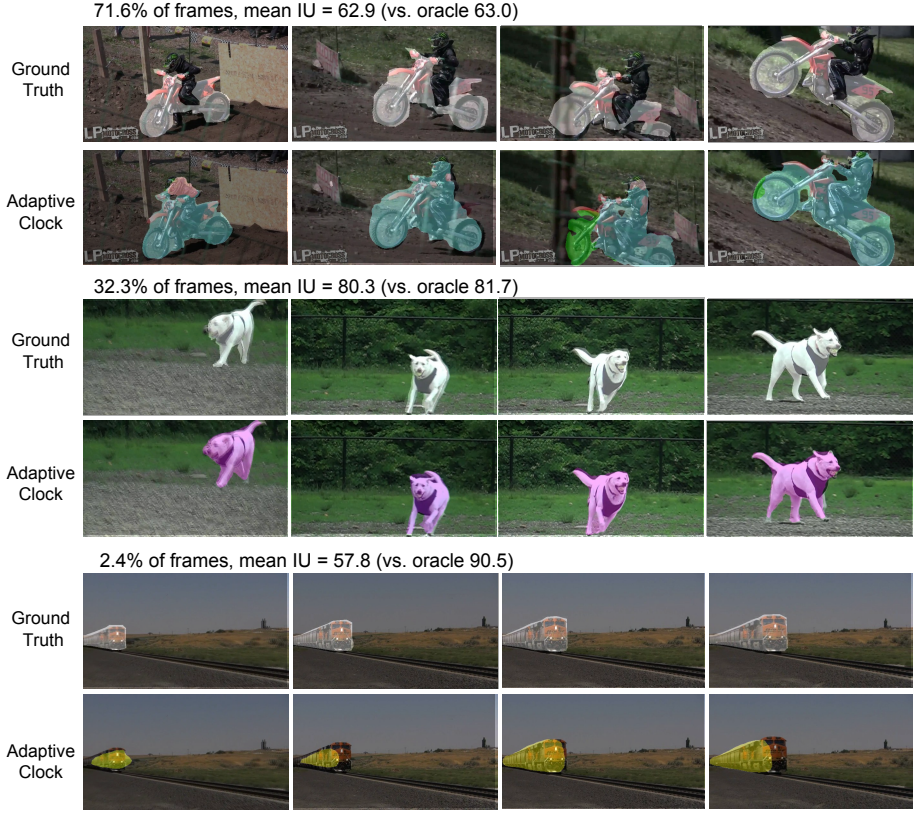

Fig. 1: Illustrative examples of our adaptive clockwork method on three different videos from Youtube-Objects. We choose the threshold (on the proportional output label change across frames) $\theta = 0.25$ for the adaptive clock. For each video, the top row shows the ground truth annotations, while the bottom row shows the output of the adaptive clockwork network. Above each sequence of frames, we include the percentage of frames for which the full network was computed, as well as the mean IU score for the adaptive method compared to the oracle of the full FCN run every frame.