# Online Model Distillation for Efficient Video Inference

Ravi Teja Mullapudi[1] Steven Chen[2] Keyi Zhang[2] Deva Ramanan[1] Kayvon Fatahalian[2]

[1] Carnegie Mellon University [2] Stanford University

## Abstract

*High-quality computer vision models typically address the problem of understanding the general distribution of real-world images. However, most cameras observe only a very small fraction of this distribution. This offers the possibility of achieving more efficient inference by specializing compact, low-cost models to the specific distribution of frames observed by a single camera. In this paper, we employ the technique of model distillation (supervising a low-cost student model using the output of a high-cost teacher) to specialize accurate, low-cost semantic segmentation models to a target video stream. Rather than learn a specialized student model on offline data from the video stream, we train the student in an online fashion on the live video, intermittently running the teacher to provide a target for learning. Online model distillation yields semantic segmentation models that closely approximate their Mask R-CNN teacher with 7 to 17× lower inference runtime cost (11 to 26× in FLOPs), even when the target video's distribution is non-stationary. Our method requires no offline pretraining on the target video stream, achieves higher accuracy and lower cost than solutions based on flow or video object segmentation, and can exhibit better temporal stability than the original teacher. We also provide a new video dataset for evaluating the efficiency of inference over long running video streams.*

## 1. Introduction

Many computer vision algorithms focus on the problem of understanding the most general distribution of real-world images (often modeled by "Internet"-scale datasets such as ImageNet [34] or COCO [24]). In contrast, most real-world video cameras capture scenes that feature a much narrower distribution of images, and this distribution can evolve continuously evolve over time. For example, stationary cameras observe scenes that evolve with time of day, changing weather conditions, and as different subjects move through the scene. TV cameras pan and zoom, most smartphone videos are hand-held, and egocentric cameras on vehicles or robots move through dynamic scenes.

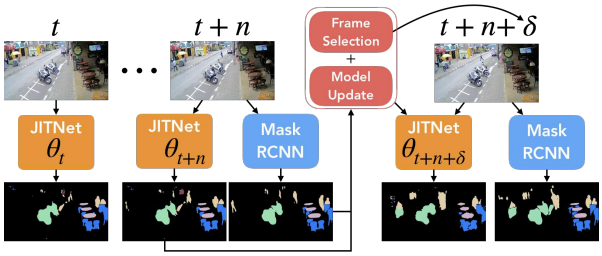


Figure 1: Online model distillation overview: A low-cost student model is tasked to generate a high-resolution, per-frame semantic segmentation. To retain high accuracy, as new frames arrive, an expensive teacher model's (MRCNN) output is periodically used as a learning target to adapt the student and selecting the next frame to request supervision. We call the student model "JITNet" since is designed to be specialized "just-in-time" for future frames.

In this paper, we embrace this reality and move away from attempting to pre-train a model on camera-specific datasets curated in advance, and instead train models *online on a live video stream as new video frames arrive*. Specifically, we apply this methodology to the task of realizing high-accuracy and low-cost semantic segmentation models that continuously adapt to the contents of a video stream.

We employ the technique of model distillation [2, 16], training a lightweight "student" model to output the predictions of a larger, reliable high-capacity "teacher", but do so in an online fashion, intermittently running the teacher on a live stream to provide a target for student learning. We find that simple models can be accurate, provided they are continuously adapted to the specific contents of a video stream as new frames arrive (i.e. models can learn to cheat—segmenting people sitting on a park lawn might be as easy as looking for shades of green!). To achieve high efficiency, we require a new model architecture that simultaneously supports low-cost inference and fast training, as well as judicious choice of when to periodically run the teacher to obtain supervision.

We show that online model distillation yields semantic segmentation models that closely approximate their Mask R-CNN [13] teacher with 7 to 17× lower inference runtime cost (11-26× when comparing FLOPs), even when the target video's distribution is non-stationary over time. Our method requires no offline pretraining on data from the tar-

get video stream, has a small number of hyper parameters, and delivers higher accuracy segmentation output, than low-cost video semantic segmentation solutions based on flow. The output of our low-cost student models can be preferable (in terms of temporal stability) to that of the expensive teacher. We also provide a new video dataset designed for evaluating the efficiency of inference over long running video streams.

## 2. Related Work

**Distillation for specialization:** Training a small, efficient model to mimic the output of a more expensive teacher has been proposed as a form of model compression (also called knowledge distillation) [2, 16]. While early explorations of distillation focused on approximating the output of a large model over the entire original data distribution, our work, like other recent work from the systems community [21], leverages distillation to create highly compact, domain-specialized models that need only mimic the teacher for a desired subset of the data. Prior specialization approaches rely on tedious configuration of models [25, 9] or careful selection of model training samples so as not to miss rare events [26]. Rather than treating model distillation as an offline training preprocess for a stationary target distribution (and incurring the high up-front training cost and the challenges of curating a representative training set for each unique video stream), we perform distillation online to adapt the student model dynamically to the changing contents of a video stream.

**Online training:** Training a model online as new video frames arrive violates the independent and identically distributed (i.i.d) assumptions of traditional stochastic gradient descent optimization. Although online learning from non-i.i.d data streams has been explored [5, 37], in general there has been relatively little work on online optimization of "deep" non-convex predictors on correlated streaming data. The major exception is the body of work on deep reinforcement learning [30], where the focus is on learning policies from experience. Online distillation can be formulated as a reinforcement or a meta-learning [8] problem. However, training methods [36, 29] employed in typical reinforcement settings are computationally expensive, require a large amount of samples, and are largely for offline use. Our goal is to train a compact model which mimics the teacher in a small temporal window. In this context, we demonstrate that standard gradient decent is effective for online training our compact architecture.

**Tracking:** Traditional object tracking methods [20, 12, 15] and more recent methods built upon deep feature hierarchies [27, 45, 17, 31] can be viewed as a form of rapid online learning of appearance models from video. Tracking parameterizes objects with bounding boxes rather than segmentation masks and its cost scales in complexity with the number of objects being tracked. Our approach for online distillation focuses on pixel-level semantic segmentation and poses a different set of performance challenges. It can be viewed as learning an appearance model for the entire scene as opposed to individual objects.

**Fast-retraining of compact models:** A fundamental theme in our work is that low-cost models that do not generalize widely are useful, provided they can be quickly retrained to new distributions. Thus, our ideas bear similarity to recent work accelerating image classification in video via online adaptation to category skew [39] and on-the-fly model training for image super-resolution [40].

**Video object segmentation:** Solutions to video object segmentation (VOS) leverage online adaptation of high-capacity deep models to a provided reference segmentation in order to propagate instance masks to future frames [32, 47, 44, 4]. The goal of these algorithms is to learn a high-quality, video-specific segmentation model for use on subsequent frames of a short video clip, not to synthesize a low-cost approximation to a pre-trained general segmentation model like Mask R-CNN [13] (MRCNN). VOS solutions require seconds to minutes of training per short video clip (longer than directly evaluating a general segmentation model itself), precluding their use in a real-time setting. We believe our compact segmentation architecture and online distillation method could be used to significantly accelerate top-performing VOS solutions (see Section 5).

**Temporal coherence in video:** Leveraging frame-to-frame coherence in video streams, such as background subtraction or difference detection, is a common way to reduce computation when processing video streams. More advanced methods seek to activate different network layers at different temporal frequencies according to expected rates of change [22, 38] or use frame-to-frame flow to warp inference results (or intermediate features) from prior frames to subsequent frames in a video [10, 48]. We show that for the task of semantic segmentation, exploiting frame-to-frame coherence in the form of model specialization (using a compact model trained on recent frames to perform inference on near future frames) is both more accurate and more efficient than flow-based methods on a wide range of videos.

## 3. Just-In-Time Model Distillation

Figure 1 provides a high-level overview of online model distillation for high quality, low-cost video semantic segmentation. On each video frame, a compact model is run, producing a pixel-level segmentation. This compact student model is periodically adapted using predictions from a high-quality teacher model (such as MRCNN [13]). Since the student model is trained online (adapted just-in-time for

| Input Size | Operation | s | r | c |
|---|---|---|---|---|
| 1280 x 720 | conv 3x3 | 2 | | 8 |
| 640 x 360 | conv 3x3 | 2 | | 8 |
| 320 x 180 | enc_block 1 | 2 | | 64 |
| 160 x 90 | enc_block 2 | 2 | | 64 |
| 80 x 45 | enc_block 3 | 2 | | 128 |
| 40 x 23 | dec_block 3 | 1 | 2 | 64 |
| 80 x 45 | dec_block 2 | 1 | 2 | 32 |
| 160 x 90 | dec_block 1 | 1 | 4 | 32 |
| 640 x 360 | conv 3x3 | 1 | | 32 |
| 640 x 360 | conv 3x3 | 1 | 2 | 32 |
| 1280 x 720 | conv 1x1 | 1 | | 32 |

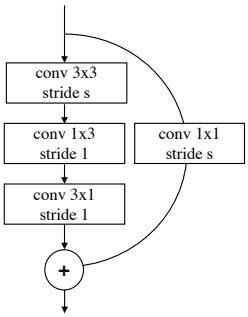


Figure 2: Left: JITNet architecture. Right: encoder/decoder block details. s = stride, r = resize, c = output channels.

future use), we refer to it as "JITNet". To make online distillation efficient in practice, our approach must: 1) use a student network that is fast for inference and fast for adaptation, 2) train this student *online* using imperfect teacher output, and 3) determine when and how to ask the teacher for labels as new frames arrive. We next address each of these challenges in turn.

### 3.1. JITNet Architecture

Efficient online adaptation requires a student architecture that (1) is efficient to evaluate even when producing high resolution outputs and (2) is amenable to fast learning. The ability to make high-resolution predictions is necessary for handling real-world video streams with objects at varying scales. Fast and stable adaptation is necessary for learning from the teacher in a small number of iterations.

Our JITNet architecture is a compact encoder-decoder [1] composed of three modified ResNet [14] blocks. To reduce computation, we replace the second 3×3 filter in each block with a separable filter (1×3 followed by a 3×1) and also limit the number of channels for high resolution feature maps. To ensure fast training, we add skip connections from each encoder block to the corresponding decoder block. This allows the gradient signal to propagate efficiently to lower layers. We include diagnostic experiments to evaluate the impact of these skip connections in supplemental.

Table 1 gives the parameter count, number of floating-point operations, and runtime of both JITNet and MRCNN on a frame of 720p video on an NVIDIA V100 GPU. (We provide both inference and training costs for JITNet.) Compact segmentation models, such as those based on MobileNet V2 [35, 43], are 3-4× slower than JITNet at high resolution and are not designed for fast, stable online training. We evaluate the MobileNet V2 architecture as the student model and demonstrate that online distillation is viable using off-the-shelf architectures. However, we find that

| Model | FLOPS (B) | | Params (M) | Time (ms) | |
|---|---|---|---|---|---|
| | Infer | Train | | Infer | Train |
| JITNet | 15.2 | 42.0 | 3 | 7 | 30 |
| MRCNN | 1390.0 | - | 141 | 300 | - |

Table 1: FLOPS (inference, training), parameter count, and runtime for both JITNet and MRCNN. JITNet has 47× fewer parameters and requires 91× (inference) and 34× (training) fewer FLOPS than MRCNN inference.

JITNet is more suitable for achieving both higher accuracy and efficency. We also evaluate JITNet variants on standard semantic segmentation to ground it relative to other efficiency-oriented architectures. Both studies are included in the supplemental.

### 3.2. Online Training with Gradient Descent

Online training presents many challenges: training samples (frames) from the video stream are highly correlated, there is continuous distribution shift in content (the past may not be representative of the future), and teacher predictions used as a proxy for "ground truth" at training can exhibit temporal instability or errors. The method for updating JITNet parameters must account for these challenges.

To generate target labels for training, we use the instance masks provided by MRCNN above a confidence threshold, and convert them to pixel-level semantic segmentation labels. All pixels where no instances are reported are labeled as background. On most video streams, this results in a significantly higher fraction of background compared to other classes. This imbalance reduces the ability of the student model to learn quickly, especially for small objects, due to most of the loss being weighted on background. We mitigate this issue by weighting the pixel loss in each predicted instance bounding box (dilated by 15%) five times higher than pixels outside boxes. This weighting focuses training on the challenging regions near object boundaries and on small objects. With these weighted labels, we compute the gradients for updating the model parameters using weighted cross-entropy loss and gradient descent. Since training JITNet on a video from a random initialization would require significant training to adapt to the stream, we pretrain JITNet on the COCO dataset, then adapt the pretrained model to each stream.

When fine-tuning models offline, it is common to only update a few layers or use small learning rates to avoid catastrophic forgetting. In contrast, for online adaptation, the goal is to minimize the cost of adapting the JITNet model so that it maintains high accuracy for current and near future video content. Rapidly specializing the compact JITNet to the temporal context retains high accuracy at low-cost. *Therefore, we update all layers with high learning rates.* Empirically, we find that gradient descent with high momentum (0.9) and learning rate (0.01) works remarkably

well for updating JITNet parameters. We believe high momentum stabilizes training due to resilience to teacher prediction noise. *We use the same parameters for all online training experiments.*

**Algorithm 1:** Online distillation

**Input:** $S_{0...n}$, $u_{max}$, $\delta_{min}$, $\delta_{max}$, $a_{thresh}$, $\theta_0$
**Output:** $P_{0...n}$

$\delta \leftarrow \delta_{min}$
**for** $t \leftarrow 0$ **to** $n$ **do**
  **if** $t \equiv 0 \pmod{\delta}$ **then**
    $L_t \leftarrow$ MaskRCNN($S_t$)
    $u \leftarrow 0$, $update \leftarrow$ **true**
    **while** $update$ **do**
      $P_t \leftarrow$ JITNet($\theta_t$, $S_t$)
      $a_{curr} \leftarrow$ MeanIoU($L_t$, $P_t$)
      **if** $u < u_{max}$ **and** $a_{curr} < a_{thresh}$ **then**
        $\theta_t \leftarrow$ UpdateJITNet($\theta_t$, $P_t$, $L_t$)
      **else**
        $update \leftarrow$ **false**
      $u \leftarrow u + 1$
    **if** $a_{curr} > a_{thresh}$ **then**
      $\delta \leftarrow \min(\delta_{max}, 2\delta)$
    **else**
      $\delta \leftarrow \max(\delta_{min}, \delta/2)$
  **else**
    $P_t \leftarrow$ JITNet($\theta_t$, $S_t$)
  $\theta_{t+1} \leftarrow \theta_t$

### 3.3. Adaptive Online Distillation

Finally, we need to determine *when* the student needs supervision from the teacher. One option is to run the teacher at a fixed rate (e.g., once every $n$ frames). However, greater efficiency is possible using a dynamic approach that adapts JITNet with teacher supervision only when its accuracy drops. Therefore, we require an algorithm that dynamically determines when it is necessary to adapt JITNet without incurring the cost of running the teacher each frame to assess JITNet's accuracy.

Our strategy is to leverage the teacher labels on prior frames not only for training, but also for *validation*: our approach ramps up (or down) the rate of teacher supervision based on recent student accuracy. Specifically, we make use of exponential back-off [11], as outlined in Algorithm 1. Inputs to our online distillation algorithm are the video stream ($S_t$), maximum number of learning steps performed on a single frame ($u_{max}$), the minimum/maximum frame strides between teacher invocations ($\delta_{min}$, $\delta_{max}$), a desired accuracy threshold ($a_{thresh}$), and the initial JITNet model parameters ($\theta_0$).

The algorithm operates in a streaming fashion and processes the frames in the video in temporal order. The teacher is only executed on frames which are multiples of the current stride ($\delta$). When the teacher is run, the algorithm computes the accuracy of the current JITNet predictions ($P_t$) with respect to the teacher predictions ($L_t$). If JITNet accuracy is less than the desired accuracy threshold (mean IoU), the model is updated using the teacher predictions as detailed in the previous section. The JITNet model is trained until it either reaches the set accuracy threshold ($a_{thresh}$) or the upper limit on update iterations ($u_{max}$) per frame. Once the training phase ends, if JITNet meets the accuracy threshold, the stride for running the teacher is doubled; otherwise, it is halved (bounded by minimum and maximum stride). The accuracy threshold is the only user-exposed knob in the algorithm. As demonstrated in our evaluation, modifying the threshold's value allows for a range of accuracy vs. efficiency trade-offs.

Even when consecutive video frames contain significant motion, their overall appearance may not change significantly. Therefore, it is better to perform more learning iterations on the current frame than to incur the high cost of running the teacher on a new, but visually similar, frame. The maximum stride was chosen so that the system can respond to changes within seconds (64 frames is about 2.6 seconds on 25 fps video). The maximum updates per frame is roughly the ratio of JITNet training time to teacher inference cost. We set $\delta_{min}$ and $\delta_{max}$ to 8 and 64 respectively, and $u_{max}$ to 8 for all experiments. *We include further discussion and an ablation study of these parameters, choices in network design, and training method in supplemental.*

## 4. Long Video Streams (LVS) Dataset

Evaluating fast video inference requires a dataset of long-running video streams that is representative of real-world camera deployments, such as automatic retail checkout, player analysis in sports, traffic violation monitoring, and wearable device video analysis for augmented reality. Existing large-scale video datasets have been designed to support training high-quality models for various tasks, such as action detection [23, 41], object detection, tracking, and segmentation [33, 46], and consist of carefully curated, diverse sets of short video clips (seconds to a couple minutes).

We create a new dataset designed for evaluating techniques for efficient inference in real-world, long-running scenarios. Our dataset, named the Long Video Streams dataset (LVS), contains 30 HD videos, each 30 minutes in duration and at least 720p resolution. (900 minutes total; for comparison, YouTube-VOS [46] is 345 minutes.) Unlike other datasets for efficient inference, which consist of streams from fixed-viewpoint cameras such as traffic cameras [19], we capture a diverse array of challenges: from fixed-viewpoint cameras, to constantly moving and zooming television cameras, and hand-held and egocentric video. Given the nature of these video streams, the most commonly occurring objects include people, cars, and animals.

It is impractical to obtain ground truth, human-labeled segmentations for all 900 minutes (1.6 million frames) of the dataset. Therefore, we curate a set of representative

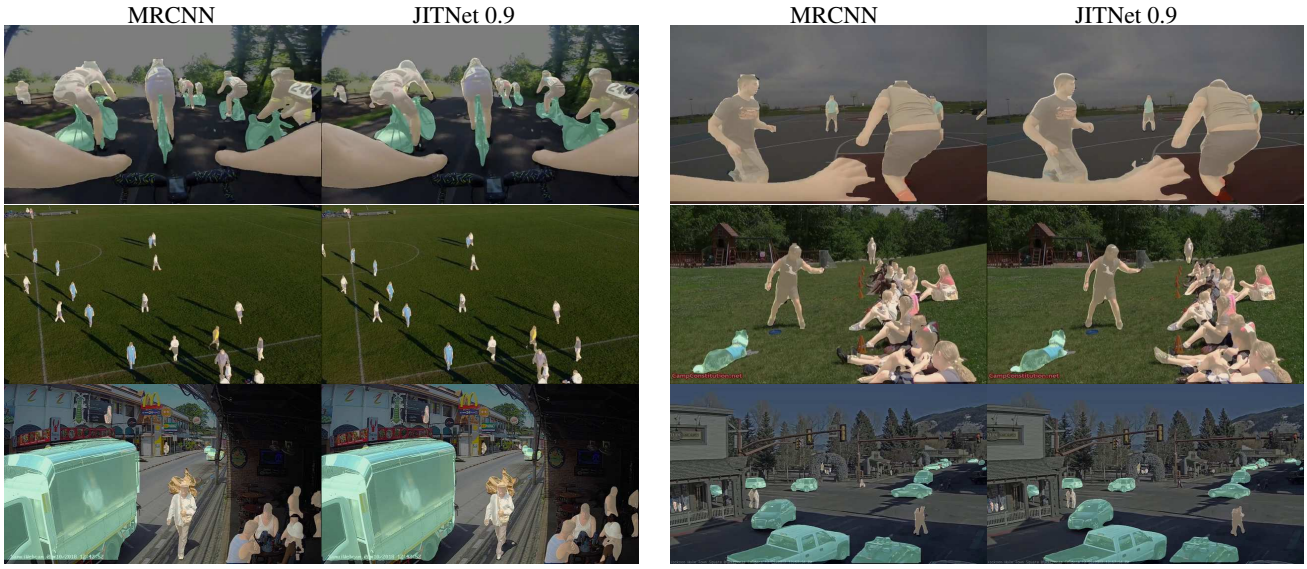


Figure 3: Frame segmentations generated by MRCNN (left) and JITNet 0.9 (right) from a subset of videos in the LVS dataset.

videos and use MRCNN [13] to generate predictions on all the frames. (We evaluated other segmentation models such as DeepLab V3 [6] and Inplace ABN [3], and found MRCNN to be produce the highest quality labels.) We use the highest-quality MRCNN [7] without test-time data augmentation, and provide its output for all dataset frames to aid evaluation of classification, detection, and segmentation (semantic and instance level) methods. Figure 3 shows a sampling of videos from the dataset with their corresponding MRCNN segmentations (left image in each group). We refer readers to supplemental for additional dataset details and visualizations of MRCNN predictions.

## 5. Evaluation

To evaluate online distillation as a strategy for efficient video segmentation, we compare its accuracy and cost with an alternative motion-based interpolation method [48] and an online approach for video object segmentation [4]. While our focus is evaluating accuracy and efficiency on long video streams (LVS), we also include results on the DAVIS video benchmark [33] in supplemental.

### 5.1. Experimental Setup

Our evaluation focuses on both the efficiency and accuracy of semantic segmentation methods relative to MRCNN. Although MRCNN trained on the COCO dataset can segment 80 classes, LVS video streams captured from a single camera over a span of 30 minutes typically encounter a small subset of these classes. For example, none of the indoor object classes such as appliances and cutlery appear in outdoor traffic intersection or sports streams. Therefore, we measure accuracy only on classes which are present in the stream and have reliable MRCNN predictions. Our evaluation focuses on object classes which can independently move, since stationary objects can be handled efficiently using simpler methods. We observed that MRCNN often confused if an instance is a car, truck, or a bus, so to improve temporal stability we combine these classes into a single class "auto" for both training and evaluation. Therefore, we only evaluate accuracy on the following classes: bird, bike, auto, dog, elephant, giraffe, horse, and person. Table 2 shows the classes that are evaluated in each individual stream as an abbreviated list following the stream name.

All evaluated methods generate pixel-level predictions for each class in the video. We use mean intersection over union (mean IoU) over the classes in each video as the accuracy metric. All results are reported on the first 30,000 frames of each video (≈16-20 minutes due to varying fps) unless otherwise specified. Timing measurements for JITNet, MRCNN (see Table 1), and other baseline methods are performed using TensorFlow 1.10.1 (CUDA 9.2/cuDNN 7.3) and PyTorch 0.4.1 for MRCNN on an NVIDIA V100 GPU. All speedup numbers are reported relative to wall-clock time of MRCNN. Note that MRCNN performs instance segmentation whereas JITNet performs semantic segmentation on a subset of classes.

### 5.2. Accuracy vs. Efficiency of Online Distillation

Table 2 gives the accuracy and performance of online distillation using JITNet at three different accuracy thresholds: JITNet 0.7, 0.8, and 0.9. Performance is the average speedup relative to MRCNN runtime, *including the cost of teacher evaluation and online JITNet training*. To provide intuition on the speedups possible on different types of videos, we organize LVS into categories of similar videos and show averages for each category (e.g., Sports (Moving)

| Video | Offline | Flow [48] | | Online Distillation | | |
|---|---|---|---|---|---|---|
| | Oracle (20%) | Slow (2.2×) (12.5%) | Fast (3.2×) ( 6.2%) | JITNet 0.7 | JITNet 0.8 | JITNet 0.9 |
| **Overall** | 80.3 | 76.6 | 65.2 | 75.5 (17.4×, 3.2%) | 78.6 (13.5×, 4.7%) | 82.5 (×7.5, 8.4%) |
| Category Averages | | | | | | |
| Sports (Fixed) | 87.5 | 81.2 | 71.0 | 80.8 (24.4×, 1.6%) | 82.8 (21.8×, 1.8%) | 87.6 (10.4×, 5.1%) |
| Sports (Moving) | 82.2 | 72.6 | 59.8 | 76.0 (20.6×, 2.1%) | 79.3 (14.5×, 3.6%) | 84.1 (6.0×, 9.1%) |
| Sports (Ego) | 72.3 | 69.4 | 55.1 | 65.0 (13.6×, 3.7%) | 70.2 (9.1×, 6.0%) | 75.0 (4.9×, 10.4%) |
| Animals | 89.0 | 83.2 | 73.4 | 82.9 (21.7×, 1.9%) | 84.3 (19.6×, 2.2%) | 87.6 (14.3×, 4.4%) |
| Traffic | 82.3 | 82.6 | 74.0 | 79.1 (11.8×, 4.6%) | 82.1 (8.5×, 7.1%) | 84.3 (5.4×, 10.1%) |
| Driving/Walking | 50.6 | 69.3 | 55.9 | 59.6 (5.8×, 8.6%) | 63.9 (4.9×, 10.5%) | 66.6 (4.3×, 11.9%) |
| Subset of Individual Video Streams | | | | | | |
| Table Tennis (P) | 89.4 | 84.8 | 75.4 | 81.5 (24.7×, 1.6%) | 83.5 (24.1×, 1.6%) | 88.3 (12.9×, 3.4%) |
| Kabaddi (P) | 88.2 | 78.9 | 66.7 | 83.8 (24.8×, 1.6%) | 84.5 (23.5×, 1.7%) | 87.9 (7.8×, 6.3%) |
| Figure Skating (P) | 84.3 | 54.8 | 37.9 | 72.3 (15.9×, 2.8%) | 76.0 (11.4×, 4.1%) | 83.5 (5.4×, 9.4%) |
| Drone (P) | 74.5 | 70.5 | 58.5 | 70.8 (15.4×, 2.8%) | 76.6 (6.9×, 7.2%) | 79.9 (4.1×, 12.5%) |
| Birds (Bi) | 92.0 | 80.0 | 68.0 | 85.3 (24.5×, 1.6%) | 85.7 (24.2×, 1.6%) | 87.9 (21.7×, 1.8%) |
| Dog (P,D,A) | 86.1 | 80.4 | 71.1 | 78.4 (19.0×, 2.2%) | 81.2 (13.8×, 3.2%) | 86.5 (6.0×, 8.4%) |
| Ego Dodgeball (P) | 82.1 | 75.5 | 60.4 | 74.3 (17.4×, 2.5%) | 79.5 (13.2×, 3.4%) | 84.2 (6.1×, 8.2%) |
| Biking (P,Bk) | 70.7 | 71.6 | 61.3 | 68.2 (12.7×, 3.5%) | 72.3 (6.7×, 7.3%) | 75.3 (4.1×, 12.4%) |
| Samui Street (P,A,Bk) | 80.6 | 83.8 | 76.5 | 78.8 (8.8×, 5.5%) | 82.6 (5.3×, 9.5%) | 83.7 (4.2×, 12.2%) |
| Driving (P,A,Bk) | 51.1 | 72.2 | 59.7 | 63.8 (5.7×, 8.8%) | 68.2 (4.5×, 11.5%) | 66.7 (4.1×, 12.4%) |

Table 2: Comparison of accuracy (mean IoU over all the classes excluding background), runtime speedup relative to MRCNN (where applicable), and the fraction of frames where MRCNN is executed. Classes present in each video are denoted by letters (A - Auto, Bi - Bird, Bk - Bike, D - Dog, E - Elephant, G - Giraffe, H - Horse, P - Person). Overall, online distillation using JITNet provides a better accuracy/efficiency tradeoff than baseline flow based methods [48] and has accuracy comparable to oracle offline models.

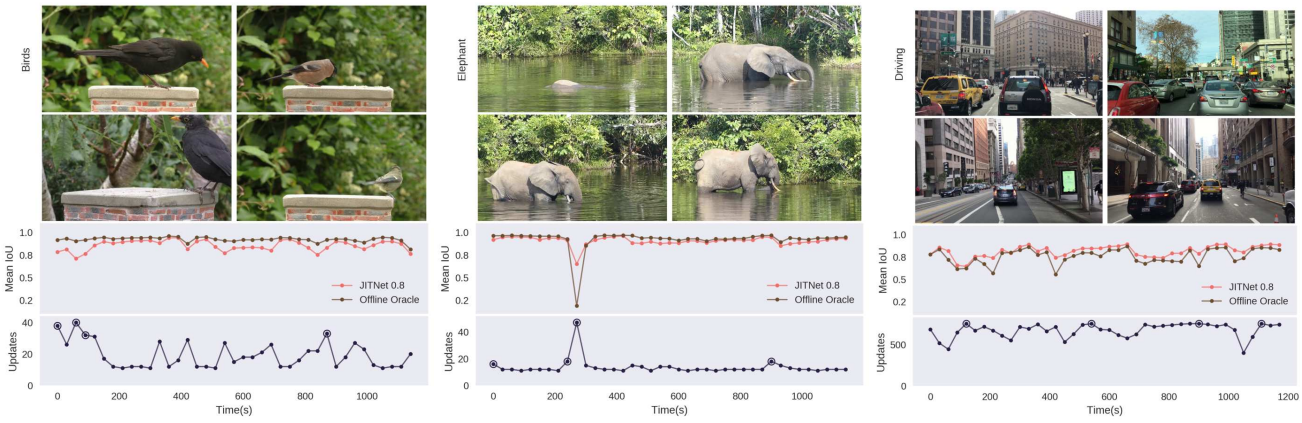


Figure 4: Top graph: the accuracy of JITNet 0.8 and Offline Oracle relative to MRCNN. Bottom graph: the number of updates to JITNet during online distillation. Plotted points are averages over a 30 second interval of the video. Images correspond to circled points in bottom plot, and show times where JITNet required frequent training to maintain accuracy.

displays average results for seven sports videos filmed with a moving camera), as well as provide per-video results for a selection of 10 videos. We also show the fraction of frames for which MRCNN predictions are used. For instance, on the Kabaddi video stream, JITNet 0.8 is 23.5 times faster than MRCNN, with a mean IoU of 84.5, and uses 510 frames out of 30,000 (1.7%) for supervision. Detailed results and videos for all streams, showing both MRCNN and JITNet predictions side-by-side for qualitative comparison, are provided in supplemental.

On average, across all sequences, JITNet 0.9 maintains 82.5 mean IoU with 7.5× runtime speedup (11.3× in FLOPs). In the lower accuracy regime, JITNet 0.7 is 17.4× faster on average (26.2× in FLOPs) while maintaining a mean IoU of 75.5. Mean IoUs in the table exclude the background class, where all the methods have high accuracy. As

Figure 5: Top: JITNet 0.9 predictions on a sequence of three frames which are roughly 0.13 seconds apart (4 frames apart) in the Figure Skating video. Bottom: Large deformations, object and camera motion prove challenging to flow based interpolation.

expected, when the accuracy threshold is increased, JITNet improves in accuracy but uses a larger fraction of teacher frames for supervision. Average speedup on sports streams from fixed cameras is higher than that for moving cameras. Even on challenging egocentric sports videos with significant motion blur, JITNet 0.9 provides 4.9× speedup while maintaining 75.0 mean IoU.

Although JITNet accuracy on the Sports (Fixed), Sports (Moving), Animals, and Traffic categories suggests potential for improvement, we observe that for streams with large objects, it is often difficult to qualitatively discern if JITNet or MRCNN produces higher quality predictions. Figure 3 displays sample frames with both MRCNN (left) and JITNet (right) predictions (zoom in to view details). The boundaries produced by JITNet on large objects (1st row) are smoother than MRCNN, since MRCNN generates low-resolution masks ($28 \times 28$) that are upsampled to full resolution. However, for videos containing small objects, such as traffic camera (Figure 3, 3rd row, right) or aerial views (2nd row, left), MRCNN produces sharper segmentations. JITNet's architecture and operating resolution would need to be improved to match MRCNN segmentations on small objects.

Streams from the Sports (Ego) category exhibit significant motion blur due to fast motion. Teacher predictions on blurred frames can be unreliable and lead to disruptive model updates. The Driving/Walking streams traverse a busy downtown and a crowded beach, and are expected to be challenging for online distillation since object instances persist on screen for only short intervals in these videos. Handling these scenarios more accurately would require faster methods for online model adaptation.

### 5.3. Comparison with Offline Oracle Specialization

The prior section shows that a JITNet model pre-trained only on COCO can be continuously adapted to a new video stream with only modest online training cost. We also compare the accuracy of just-in-time adaptation to the results of specializing JITNet to the contents of the each stream *entirely offline*, and performing no online training. To simulate the effects of near best-case offline pre-training, we train JITNet models on every 5th frame of the entire 20 minute test video sequence (6,000 training frames). We refer to these models as "offline oracle" models since they are constructed by pre-training on the test set, and serve as a strong baseline for the accuracy achievable via offline specialization. All offline oracle models were pre-trained on COCO, and undergo *one hour* of pre-training on 4 GPUs using traditional random-batch SGD. (See supplemental for further details.) Recall that in contrast, *online adaptation incurs no pre-training cost* and trains in a streaming fashion.

As shown in Table 2, JITNet 0.9 is on average more accurate than the offline oracle. Note that JITNet 0.9 uses only 8.4% of frames on average for supervision, while the oracle is trained using 20%. This trend also holds for the subcategory averages. This suggests that the compact JITNet model does not have sufficient capacity to fully capture the diversity present in the 20 minute stream.

Figure 4 shows mean IoU of JITNet 0.8 and the offline oracle across time for three videos. The top plot displays mean IoU of both methods (data points are averages over 30 second time intervals). The bottom plot displays the number of JITNet model updates in each interval. Images above the plots are representative frames from time intervals requiring the most JITNet updates. In the Birds video (left), these intervals correspond to events when new birds appear. In comparison, the Elephant video (center) contains a single elephant from different viewpoints and camera angles. The offline oracle model incurs a significant accuracy drop when the elephant dips into water. (This rare event makes up only a small fraction of the offline training set.) JITNet 0.8 displays a smaller drop since it specializes immediately to the novel scene characteristics. The Driving video (right) is challenging for both the offline oracle and online JITNet since it features significant visual diversity and continuous change. However, while the mean IOU of both methods is lower, online adaptation consistently outperforms the offline oracle in this case as well.

### 5.4. Comparison with Motion-Based Interpolation

An alternative approach to improving segmentation efficiency on video is to compute teacher predictions on a sparse set of frames and interpolate the results using flow. Table 2 shows two baselines that propagate pixel segmentations using Dense Feature Flow [48], although we upgrade the flow estimation network from FlowNet2 [18] to modern methods. (We propagate labels, not features, since this was shown to be as effective [48].) The expensive variant (Flow (Slow)) runs MRCNN every 8th frame and uses PWC-Net [42] to estimate optical flow between frames. MRCNN labels are propagated to the next seven frames using the estimated flow. The fast variant (Flow (Fast)) uses the same propagation mechanism but runs MRCNN every 16th frame and uses a faster PWC-Net. Overall JITNet 0.7 is 2.8×

| Category | OSVOS (3.3%) | | JITNet 0.8 |
|---|---|---|---|
| | A | B | |
| **Overall** | 59.9 | 60.0 | 77.4 (14.5×, 4.6%) |
| Sports (Fixed) | 75.7 | 75.7 | 82.3 (24.0×, 1.6%) |
| Sports (Moving) | 69.1 | 69.3 | 78.7 (16.3×, 2.9%) |
| Sports (Ego) | 67.6 | 68.1 | 74.8 (9.5×, 5.9%) |
| Animals | 79.3 | 79.8 | 86.0 (19.7×, 2.1%) |
| Traffic | 22.3 | 21.9 | 70.8 (8.4×, 7.7%) |
| Driving/Walking | 36.7 | 36.3 | 66.8 (4.3×, 11.8%) |

Table 3: JITNet 0.8 generates higher accuracy segmentations than OSVOS on LVS and is two orders of magnitude lower cost. Percentages give the fraction of frames used for MRCNN supervision.

faster and more accurate than the fast flow variant, and JITNet 0.9 has significantly higher accuracy than the slow flow variant except in the Driving/Walking category.

Figure 5 illustrates the challenge of using flow to interpolate sparse predictions. Notice how the ice skaters in the video undergo significant deformation, making them hard to track via flow. In contrast, online distillation trains JITNet to learn the appearance of scene objects (it leverages temporal coherence by reusing the model over local time windows), allowing it to produce high-quality segmentations despite complex motion. The slower flow baseline performs well compared to online adaptation on rare classes in the Driving (Bike) and Walking (Auto) streams, since flow is agnostic to semantic classes. Given the orthogonal nature of flow and online adaptation, it is possible a combination of these approaches could be used to handle streams with rapid appearance shifts.

### 5.5. Comparison with Video Object Segmentation

Although not motivated by efficiency, video object segmentation (VOS) solutions employ a form of online adaptation: they train a model to segment future video frames based on supervision provided in the first frame. We evaluate the accuracy of the OSVOS [4] approach against JITNet on two-minute segments of each LVS video. (OSVOS was too expensive to run on longer segments.) For each 30-frame interval of the segment, we use MRCNN to generate a starting foreground mask, train the OSVOS model on the starting mask, and use the resulting model for segmenting the next 29 frames. We train OSVOS for 30 seconds on each starting frame, which requires approximately *one hour* to run OSVOS on each two-minute video segment. Since segmenting all classes in the LVS videos would require running OSVOS once per class, we run OSVOS on only one class per video (person or animal class in each stream) and compare JITNet accuracy with OSVOS on the designated class. (Recall JITNet segments all classes.) Furthermore, we run two configurations of OSVOS: in mode (A) we use the OSVOS model from the previous 30-frame interval as the starting point for training in the next interval (a form of continuous adaptation). In mode (B) we reset to the pre-trained OSVOS model for each 30-frame interval.

Table 3 compares the accuracy of both OSVOS variants to online distillation with JITNet. The table also provides model accuracy, runtime speedup relative to MRCNN, and the fraction of frames used by JITNet 0.8 for supervision in the two-minute interval. Overall JITNet 0.8 is more accurate than OSVOS and *two orders of magnitude* faster. On Traffic streams, which have small objects, and Driving/Walking streams with rapid appearance changes, OSVOS has significantly lower accuracy than JITNet 0.8. We also observe that the mode A variant of OSVOS (continuously adapted) performs worse than the variant which is re-initialized. We believe the JITNet architecture could be employed as a means to significantly accelerate online VOS methods like OnAVOS [44] or more recent OSVOS-S [28] (uses MRCNN predictions every frame).

## 6. Conclusion

In this work we demonstrate that for common, real-world video streaming scenarios, it is possible to perform online distillation of compact (low cost) models to obtain semantic segmentation accuracy that is comparable with an expensive high capacity teacher. Going forward, we hope that our results encourage exploration of online distillation for domain adaptation and self-supervised learning. More generally, with continuous capture of high-resolution video streams becoming increasingly commonplace, we believe it is relevant for the broader community to think about the design and training of models that are not trained offline on carefully curated datasets, but instead continuously evolve each day with the data that they observe from specific video streams. We hope that the Long Video Streams dataset serves this line of research.

**Acknowledgement** This research is based upon work supported in part by NSF Grant 1618903 and IIS-1422767, the Intel Science and Technology Center for Visual Cloud Systems (ISTC-VCS), the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R & D Contract No. D17PC00345, and the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0051, and a Google Faculty Fellowship. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

## References


[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 3

[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541. ACM, 2006. 1, 2

[3] Samuel Bulo, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of DNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5

[4] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 2, 5, 8

[5] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. 2

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5

[7] FAIR. Detectron Mask R-CNN. https://github.com/facebookresearch/Detectron, 2018. 5

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 2

[9] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008. 2

[10] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video CNNs through representation warping. *CoRR, abs/1708.03088*, 2017. 2

[11] Jonathan Goodman, Albert G Greenberg, Neal Madras, and Peter March. Stability of binary exponential backoff. *Journal of the ACM (JACM)*, 35(3):579–602, 1988. 4

[12] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2096–2109, 2016. 2

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017. 1, 2, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[15] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015. 2

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2

[17] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency maps with convolutional neural network. In *International Conference on Machine Learning*, pages 597–606, 2015. 2

[18] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017. 7

[19] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. Chameleon: Scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, pages 253–266, New York, NY, USA, 2018. ACM. 4

[20] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012. 2

[21] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. Noscope: Optimizing neural network queries over video at scale. *Proceedings of the VLDB Endowment*, 10(11):1586–1597, 2017. 2

[22] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork RNN. In *Proceedings of the International Conference on Machine Learning*, pages 1863–1871, 2014. 2

[23] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 4

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1

[25] Honghai Liu, Shengyong Chen, and Naoyuki Kubota. Intelligent video systems and analytics: A survey. *IEEE Trans. Industrial Informatics*, 9(3):1222–1233, 2013. 2

[26] Wei-Lwun Lu, Jo-Anne Ting, James J Little, and Kevin P Murphy. Learning to track and identify players from broadcast sports videos. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1704–1716, 2013. 2

[27] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3074–3082, 2015. 2

[28] K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. 2018. 8

[29] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016. 2

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 2

[31] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4293–4302. IEEE, 2016. 2

[32] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[33] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 5

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[36] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[37] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012. 2

[38] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 852–868. Springer, 2016. 2

[39] H. Shen, S. Han, M. Philipose, and A. Krishnamurthy. Fast video classification via adaptive cascading of deep models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[40] Assaf Shocher, Nadav Cohen, and Michal Irani. "Zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018. 2

[41] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[43] TensorFlow. TensorFlow DeepLab Model Zoo. https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md, 2018. 3

[44] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 2, 8

[45] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3119–3127, 2015. 2

[46] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. YouTube-VOS: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 4

[47] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. *Proceedings of the International Conference on Robotics and Automation*, 2018. 2

[48] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 7, 2017. 2, 5, 6, 7