

Bridging the Edge-Cloud Barrier for Real-time Advanced Vision Analytics

Yiding Wang
HKUST

Weiyan Wang
HKUST

Junxue Zhang
HKUST

Junchen Jiang
University of Chicago

Kai Chen
HKUST

Abstract

Advanced vision analytics plays a key role in a plethora of real-world applications. Unfortunately, many of these applications fail to leverage the abundant compute resource in cloud services, because they require high computing resources and high-quality video input, but the network connections between visual sensors (cameras) and the cloud servers do not always provide sufficient and stable bandwidth to stream high-fidelity video data in real time.

This paper presents CloudSeg, an edge-to-cloud framework for advanced vision analytics that co-designs the cloud-side inference with real-time video streaming, to achieve both low latency and high inference accuracy. The core idea is to send the video stream in low resolution, but recover the high-resolution frames from the low-resolution stream via a *super-resolution* procedure tailored for the actual analytics tasks. In essence, CloudSeg trades additional cloud-side computation (super-resolution) for significantly reduced network bandwidth. Our initial evaluation shows that compared to previous work, CloudSeg can reduce bandwidth consumption by $\sim 6.8\times$ with negligible drop in accuracy.

1 Introduction

Recent years have seen an explosive growth of real-world vision-based applications, primarily driven by advances in traditionally challenging vision tasks, e.g. multiple object detection [21, 24], semantic segmentation [14, 30], instance segmentation [8, 25] and panoptic segmentation [12, 13]. To obtain adequate inference accuracy, these tasks often require *both* high computation power and high-resolution images (or video streams). This, however, poses a fundamental challenge to real-time vision-based applications. On the one hand, many video analytics tasks have been optimized for cloud environments (e.g. [10, 29]). This seems to suggest one should send data via the bandwidth-limited connection to the cloud in the hope that the sophisticated cloud-side model can still extract enough information from the limited data. This hope, unfortunately, turns out to be illusory for advanced vision analytics

tasks; while reducing video resolution (or frame rate) does save bandwidth, it will nevertheless inflict non-trivial drop in inference accuracy [4, 28]. On the other hand, some real-time advanced vision applications, e.g. autonomous driving, put expensive hardware accelerators [15] on edge devices to perform local inference. However, this approach does not make much economic sense when future applications require large-scale deployment, e.g. fleets of delivery vehicles [23].

In this paper, we present *CloudSeg*, an edge-to-cloud video analytics framework that optimizes for both high accuracy and low latency. CloudSeg lowers the quality in which the video is sent to the cloud, but it then runs a *super-resolution* (SR) procedure at the cloud server to reconstruct high-quality videos before executing the actual video analytics (video segmentation, object detection, etc.). This approach is in the same spirit of prior applications of SR where high-quality images are needed when only low-quality images are available [7]. What’s new is that we found it can potentially strike a desirable balance between accuracy and latency in the edge-to-cloud analytics setting. Essentially, running SR uses much less cloud resource and cause less delay than the actual inference, and it could restore the video quality so that video analytics task could achieve the same accuracy as if the video is streamed in high quality.

That said, we found that current SR models do not always perform as well as expected. This is because traditional SR models seek to retain pixel-level details (i.e., minimizing visual quality loss), which does not always retain the information needed by vision analytics. A notable example of such mismatch is the recovery of small details such as distant pedestrians. Traditional SR models, trained to uniformly recover all pixels to meet a given target quality, may fail to recover enough details for small object than for big objects, thus making small objects hard to identify or segment. However, these small objects are crucial (just as other large objects) to the accuracy of vision tasks and the practicality of applications e.g. autonomous driving.

To address the limitations of SR, we train our SR model in such a way that it reduces both quality loss as well as the

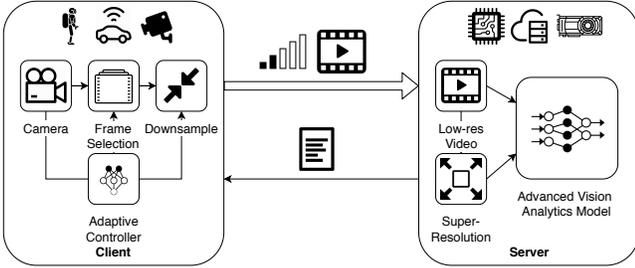


Figure 1: *CloudSeg framework overview*

accuracy loss of the analytics task. Given an existing SR model, which is essentially a deep neural network (DNN), we use an additional training process to fine-tune the weights of the SR model to minimize the accuracy loss of the super-resolved frames on the cloud-side analytics model, as showed in Figure 2. To this end, the fine-tuning process uses the difference of inference accuracy between the original frames and the super-resolved frames as the loss function (§3.1).

We further integrate CloudSeg with analytics models using the popular *pyramid structure* [16, 24, 30] to reduce unnecessary downsampling overhead by reusing low-resolution data (§3.2). Besides, we adaptively select *useful frames* for instance-level tasks with a 2-level frame selector to further reduce overhead while keeping good trackability. Finally, to cope with the bandwidth fluctuation, inspired by prior work [28], we adapt the video resolution and frame rate to the available bandwidth (§3.3). Our preliminary results show that CloudSeg on average can save $\sim 6.8\times$ bandwidth compared to a recently proposed baseline [28] while achieving same inference accuracy.

2 Background

2.1 Requirements of advanced vision analytics

This work considers *advanced* vision analytics tasks that require *low latency* and *high inference accuracy*. For example, for autonomous driving and multiple object detection applications, small and distant objects still matter so high-resolution input is necessary; for autonomous driving and robotics applications, high-frame-rate input is essential to ensure trackability because scenes generally change fast and real-time interaction requires low latency.

To achieve desirable accuracy, these advanced vision analytics needs to run highly complex models, increasingly in the form of deep neural networks (DNNs), with expensive hardware (GPUs) as well as on high-resolution inputs. For example, state-of-the-art real-time object detection model SSD [17] can run at 300×300 in speed of 59 FPS (frames per second), while real-time accurate semantic segmentation model IC-Net [30] runs at 27 FPS on a 2048×1024 resolution input, both on Nvidia Titan X.

2.2 Video streaming for vision analytics

In many real-time video analytics applications, it is, however, fundamentally challenging to colocate expensive compute resources with high-fidelity video data considering scalability and cost. With more edge devices deployed in geographically distributed locations, how to collect their video streams to cloud for analytics without using too much bandwidth has attracted much attention.

The conventional wisdom has been that an edge device should compress its video, via pixel-level (spatial) downsampling and frame-level (temporal) downsampling, and ensure that sufficient information is retained, so that the cloud server can still run the vision analytics model on the downsampled video and produce highly accurate inference as if the video is not compressed. Specifically, AWStream [28] learns a Pareto-optimal policy and adaptively selects a data rate degradation strategy to meet the accuracy and bandwidth trade-off over the wide-area network for video object detection. FilterForward [3] filters relevant video frames on the edge with small neural networks to save bandwidth and it shares the same spirit of prior filter-based frameworks [4, 11, 20].

As we will see in §4.1, while this approach [28] works to some extent, it ultimately imposes a hard trade-off: at some point, when the frame rate needs to be retained high for advanced applications, more aggressive video downsampling always inflict a non-trivial drop in accuracy. As a result, it cannot be directly applied to serve advanced vision analytics.

2.3 Super-resolution for vision analytics

Our solution is based on the recent advance in super-resolution (SR) techniques. Ideally, a SR model can reconstruct a high-resolution scene from a low-resolution scene, by inferring details based only on information in the low-resolution input. Recently, DNN-based SR models have significantly improved the performance [2, 9]. Prior work has shown that SR is a promising approach to improving video streaming quality [27] and boosting vision analytics accuracy [7] when only low-resolution videos are available.

Our work differs from the prior work in two important aspects. First, we show that by applying SR on the downsampled video, the resulting reconstructed high-resolution video can usually produce almost the same accuracy as if the video was not downsampled. Although such result is not surprising, it suggests that SR could serve as an architectural role of “glue” between the video encoding stack (for saving bandwidth) and the video analytics (for maximizing accuracy). Second, through experiments, we also shed light on the limitations of current SR models, which are tailored to retain visual-based human-perceptual information, rather than maximizing analytics accuracy. Instead, we present a new way of training SR models such that the resulting model maximizes both the post-SR visual quality and the analytics accuracy.

3 Design

We present CloudSeg, a new edge-to-cloud framework for real-time advanced video analytics. The workflow of CloudSeg is illustrated in Figure 1. On the edge side, the sensor (camera) adaptively downsamples a high-resolution video, and streams it to the cloud server via network. On the cloud side, the server then processes the video, runs (DNN-based) inference, and finally returns the inference results to the edge device. CloudSeg consists of three main components, which we will explain next.

3.1 Analytics-aware super-resolution

To address the challenges of serving advanced vision analytics applications over the cloud as well as the limitations of analytics-agnostic SR discussed in §2, CloudSeg trains the SR model with a novel approach so that the resulting model maximizes both the post-SR visual quality and the analytics accuracy. We first train the SR model offline on the same dataset which was used to train the actual vision model, then fine-tune the SR model with an accuracy-oriented metric to further improve the inference accuracy especially on critical details. The resulting SR model is used to reconstruct high-resolution (HR) images from the low-resolution (LR) input images before feeding them to the actual inference model on the cloud server.

We use a state-of-the-art super-resolution model CARN [2] to illustrate our method (Figure 2), which involves two steps:

- *Base SR training:* We use a semantic segmentation model ICNet [30] as the vision-task model. Originally, CARN is trained to minimize the quality loss (structural similarity index, or SSIM) between the original HR frame and the resulting super-resolved (SR) frame.
- *Analytics-aware fine-tuning:* Next, we further train the SR model to improve the accuracy of specific vision tasks. We calculate the difference of inference accuracy between running the vision model on the original HR image and running it on the SR image. This difference is then used as the loss function which the new SR model is trained to minimize.

To better tailor SR model for our purposes, CloudSeg also uses different training parameters make the resulting SR model more amenable to video analytics. CARN adopts a patch-based CNN model, where a patch is any fixed-sized (e.g., 64×64) region which will undergo different forms of random deformations (cropping, flipping, rotation). To make the SR training aware of the analytics task, we use a much more fine-grained patch than that used in ICNet (720×720). In our applications, small patches are crucial to identifying and retaining small details, such as distant pedestrians. However, CloudSeg applies the fine-grained patch only when fine-tuning SR model weights, for practical reasons. First, a small

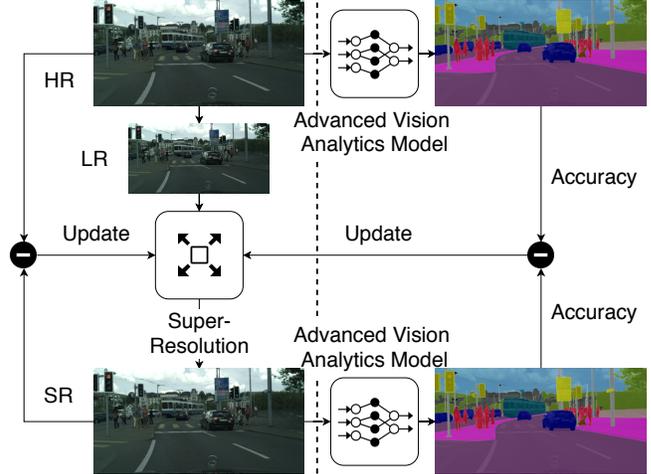


Figure 2: High-level workflow of how to train the SR models in CloudSeg, and feed the SR frames to the inference model.

patch is not well-suited for accurate ICNet segmentation inference, so during the SR fine-tuning, we use the same patch size as the original ICNet. Note CloudSeg does not exactly rely on quality recovery since our ultimate goal is accurate vision inference, so a larger patch size is well-suited for the analytics-aware fine tuning.

We also found another simple yet effective change that can boost the training of the base SR model. Many machine-learning dataset, including the one we use (Cityscapes [5]) to train segmentation model, have many labeled images, but contains even more massive unlabeled images, which are collected but not manually labeled. These unlabeled images do not add value to the training of any machine-learning model, but they are as useful as labeled images when training the SR model! Compared to the naive approach where both SR and the vision analytics model are trained on the same labeled dataset (a small subset of the whole image set), incorporating the unlabeled images in the training of the base SR model can significantly boost the effectiveness (quality recovery) of the trained SR model.

3.2 Using super-resolved data in vision models

Having super-resolved images is functionally sufficient to meet the need of advanced vision analytics. However, the rapid advances of computer vision have given rise to a variety of vision analysis techniques, many of which, fortunately, offer new opportunities to improve inference latency when super-resolution process is *implemented* appropriately. Here, we briefly explain two examples.

Data reusing: In the context of semantics segmentation (and other complex vision tasks), a common approach to reducing inference delay (and saving cost) is through running

inference at multiple resolutions simultaneously. The idea is to produce most results using the LR images, and use HR images only when details are needed [16, 24, 30]. Now, if CloudSeg feeds the SR images to these models, it would cause the models to downsize the SR images (back) to LR images before running inference on both SR and LR images. Such repeated decoding/encoding is an obvious waste of resource and time!

Instead, CloudSeg can keep a copy of SR and LR images, feed both copies of each image to the inference model, and bypass the downsizing steps in the inference model to save inference delay. Moreover, doing so likely will also improve inference accuracy since downsizing SR back to LR is another lossy re-encoding, which means the resulting LR image contains less details than the LR image from which the SR was created.

Intelligent frame filtering: Besides the aforementioned per-frame optimization, it is also crucial to balance latency and accuracy along the temporal dimension. There has been plenty of frame-filtering schemes that discard redundant frames on the edge side, so only important frames (e.g., those that contains new objects) will be sent to the server [8, 25]. While they are useful in reducing bandwidth, these schemes are generally agnostic to the vision inference on the server-side [14, 22, 31]; the selected frames are not always important to the cloud-side analytics (e.g., no new objects are actually present).

CloudSeg attempts to address this issue through a more *analytics-aware* frame filtering. We use a state-of-the-art technique [14] to identify the *key frames* that contain most deviations relevant to the analytics, e.g., semantics segmentation. Intuitively, when the scene is changing rapidly, useful and key frames are more concentrated than when the scene is stable. Note that the notion of key frame is similar to that in video encoding, but the key frames in our context are more dependent on the specific vision tasks; rather than exhibiting substantial vision deviation (which is used in video coding, e.g., H264), the key frames in our context exhibit substantial content deviation that can potentially lead to different inference results.

In particular, CloudSeg uses a 2-level frame selector on the edge device. As Figure 3 shows, two thresholds target different frames: the higher one filters out *key frames* while the lower one filters out *useful frames*, and other stale frames will not be streamed to the server. Two thresholds are set by the adaptive controller in §3.3 such that they can be updated according to network conditions and application requirements. Useful frames and tagged key frames will be streamed to the server and are compatible with the *key frame feature propagation* structure [14, 22, 31]. For an instance-level model without key frame scheme, the selector falls back to a single-level useful frame filter to save bandwidth.

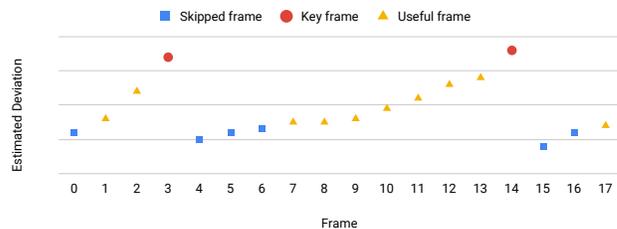


Figure 3: *Edge-side 2-level frame selection. Frames are classified in three categories: key frames and useful frames that will be analyzed by the server-side logic, and skipped frames that will be discarded on the client side.*

3.3 Adaptive bitrate controlling

While SR well handles the latency/accuracy trade-off in general (as shown in §4), it may fail in certain extreme cases such as those caused by variance of scenes, e.g., light and weather changes or glitches (worst cases) of SR. The blue line in Figure 4 shows the inference accuracy (mIoU) on a 30-second clip (experiment setting in §4). The minimal accuracy (≤ 0.6) is unacceptable for real-world applications, even the average is not that bad. This problem can be addressed by streaming a higher-resolution video to the backend model or even bypassing SR, as the red dashed line shows.

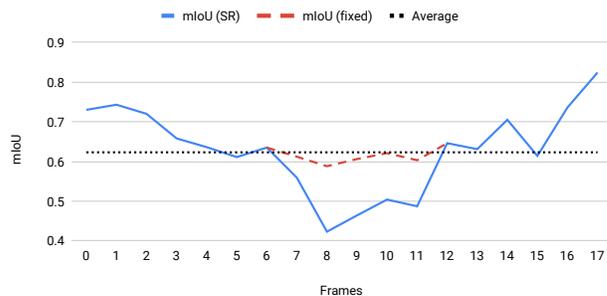


Figure 4: *Variance of the inference accuracy under CloudSeg and CloudSeg with fixed parameters.*

To that end, we adopt an adaptive bitrate controller, similar to prior work [28], to handle the variance of network conditions, real-world scene changes, or performance drop of SR. Basically, it gathers network information from the transport layer, e.g., bandwidth and network latency, as well as application performance from the application layer, e.g., inference accuracy and computation time. Through offline/online profiling and training, we can learn a model and find a suitable knob policy including downsampling rate, frame rate and frame thresholds with little overhead.

4 Preliminary results

We implement a prototype of CloudSeg (without the intelligent frame filtering in §3.2) with semantic segmentation model ICNet [30] as the cloud-side vision model. We use Cityscapes [5] dataset, where videos are 2048×1024 and 17 FPS with 8-bit RGB frames. We empirically show that, compared to AWStream, a recently proposed baseline [28], CloudSeg can perform advanced vision analytics over the cloud with lower bandwidth consumption and lower latency with minor drop in accuracy.

4.1 Accuracy vs. bandwidth

For fairness, CloudSeg, as well as AWStream and baseline no-downsizing video encoding, stream videos to the server in H.264 form (which is used by AWStream). We measure accuracy in mean intersection-over-union (mIoU), higher the better. As shown in Figure 5, the original 2048×1024 video needs 10 Mbps and achieves the highest accuracy (0.67). In contrast, CloudSeg achieves slightly lower accuracy (0.65) with $13.3 \times$ less bandwidth consumption (750kbps or 512×256). AWStream is capable of getting the same accuracy as CloudSeg, but at the cost of using 5.1Mbps (1440×720 resolution).

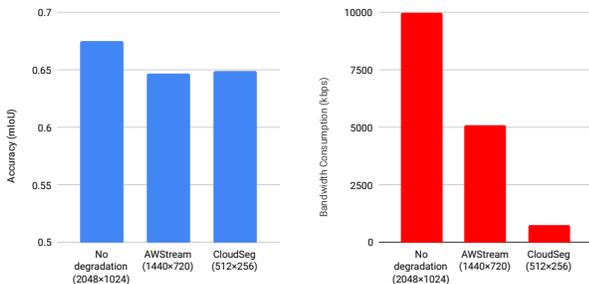


Figure 5: *Bandwidth consumption to achieve comparable accuracy. CloudSeg strikes a more favorable balance between bandwidth consumption and inference accuracy.*

4.2 Benefit of analytics-aware super-resolution

Next we compare the accuracy of the analytics-aware SR scheme in CloudSeg with a standard SR scheme (ICNet model [30] trained on the same Cityscapes dataset) and the bilinear algorithm which is the default image resizing algorithm of TensorFlow [1]. We first get the LR frames by resizing the original HR frames to 512×256 with bilinear. Then we up-sample the LR image to the original resolution using the three schemes. Table 1 compares the accuracy in mIoU across the schemes. We can see the accuracy of CloudSeg’s SR is much closer to that of the original HR images than the baselines.

This is in interesting contrast to the comparison among their quality loss (the last two rows of Table 1) measured in PSNR and SSIM [26]: CloudSeg exhibits very similar quality loss to the standard SR. This contrast is because, while CloudSeg fine-tunes the standard SR model by making it more aware of the video analytics task (increasing accuracy) without affecting its visual quality loss (§3.1). In other words, CloudSeg improves the reconstruction of small details e.g. sharper edges of people in the distance, which are important for the target advanced vision applications, but are not explicitly taken into account in the objective of standard SR.

Metrics	Bilinear	SR	CloudSeg	Original HR
Accuracy (mIoU)	0.582	0.633	0.649	0.675
PSNR	31.00	35.21	35.44	—
SSIM	0.936	0.970	0.968	—

Table 1: *Performance of different upsampling methods. CloudSeg achieves higher accuracy than other image-resizing baselines, while exhibiting similar visual quality loss, suggesting CloudSeg’s SR fine-tuning does improve inference accuracy on top of the standard the SR model.*

4.3 Inference delay

Besides reducing the network latency, CloudSeg also has low server-side inference delay as well. The server-side inference delay includes the SR image reconstruction and vision model inference. We test the average inference time of super-resolving Cityscapes frames from 512×256 to 2048×1024 and semantic segmentation (ICNet) on a single Nvidia V100 GPU. The results are showed in Table 2. The pipeline of SR and semantic segmentation works at 23.5 FPS. Considering that the framework overhead (e.g. image loading, client-side processing) takes a rather small fraction, CloudSeg can run in real time.

Model	Time (ms)	Frame-per-sec
Super-Resolution	6.2	161.3
Semantic Segmentation	36.3	27.5
Total	42.5	23.5

Table 2: *Server-side inference delay per frame. The additional SR procedure of CloudSeg adds only a small server-side overhead compared to the actual vision inference.*

Discussion

Can we do better under extremely low bandwidth? Our framework can greatly reduce the bandwidth consumption, but the bandwidth of wireless WAN could be extremely low that even our SR method can not recover sufficient inference accuracy. Again we turn to deep learning. Similar to SR, we consider utilize the computing resource of the cloud server to save bandwidth, with neural frame interpolation [18, 19]. We will investigate its impact on the tracking accuracy of instance-level tasks.

Handling uncertainty when applying ML for system

Applying learning-based techniques may increase the uncertainty of the real-world system, especially in applications e.g. autonomous driving [6]. Uncertainty handling deserves further research in systems with ML components.

Video QoE for vision analytics tasks Traditional QoE of video streaming is designed for user watching experience. From our preliminary results, vision analytics tasks may value different metrics than human audience. With a special QoE for vision tasks, the cloud analytics framework may save more bandwidth and achieve better performance.

References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.
- [2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [3] Christopher Canel, Thomas Kim, Giulio Zhou, Conglong Li, Hyeontaek Lim, David G. Andersen, Michael Kaminsky, and Subramanya R. Dulloor. Scaling video analytics on constrained edge nodes. In *2nd Conference on Systems and Machine Learning (SysML)*, 2019.
- [4] Tiffany Yu-Han Chen, Lenin Ravindranath, Shuo Deng, Paramvir Bahl, and Hari Balakrishnan. Glimpse: Continuous, real-time object recognition on mobile devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 155–168. ACM, 2015.
- [5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [6] Krzysztof Czarnecki and Rick Salay. Towards a framework to manage perceptual uncertainty for safe automated driving. In *International Conference on Computer Safety, Reliability, and Security*, pages 439–445. Springer, 2018.
- [7] Dengxin Dai, Yujian Wang, Yuhua Chen, and Luc Van Gool. Is image super-resolution helpful for other vision tasks? In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.
- [8] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: A real-time deep network for scene understanding. In *ICCV 2017-International Conference on Computer Vision*, page 11, 2017.
- [9] Yuchen Fan, Jiahui Yu, and Thomas S Huang. Wide-activated deep residual networks based restoration for bpg-compressed images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2621–2624, 2018.
- [10] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B Gibbons, and Onur Mutlu. Focus: Querying large video datasets with low latency and low cost. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 269–286, 2018.
- [11] 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.
- [12] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *arXiv preprint arXiv:1901.02446*, 2019.
- [13] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *arXiv preprint arXiv:1801.00868*, 2018.
- [14] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5997–6005, 2018.
- [15] Shih-Chieh Lin, Yunqi Zhang, Chang-Hong Hsu, Matt Skach, Md E Haque, Lingjia Tang, and Jason Mars. The

- architectural implications of autonomous driving: Constraints and acceleration. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 751–766. ACM, 2018.
- [16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.
- [17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [18] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 498–507, 2018.
- [19] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018.
- [20] Chrisma Pakha, Aakanksha Chowdhery, and Junchen Jiang. Reinventing video streaming for distributed vision analytics. In *10th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 18)*, 2018.
- [21] Vit Ruzicka and Franz Franchetti. Fast and accurate object detection in high resolution 4k and 8k video using gpus. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2018.
- [22] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork convnets for video semantic segmentation. In *European Conference on Computer Vision*, pages 852–868. Springer, 2016.
- [23] Matt Simon and Arielle Pardes. The prime challenges for scout, amazon’s new delivery robot | wired.
- [24] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9333–9343, 2018.
- [25] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020. IEEE, 2018.
- [26] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [27] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. Neural adaptive content-aware internet video delivery. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 645–661, 2018.
- [28] Ben Zhang, Xin Jin, Sylvia Ratnasamy, John Wawrzyniek, and Edward A Lee. Awstream: Adaptive wide-area streaming analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 236–252. ACM, 2018.
- [29] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J Freedman. Live video analytics at scale with approximation and delay-tolerance. In *NSDI*, volume 9, page 1, 2017.
- [30] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnets for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.
- [31] 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*, pages 2349–2358, 2017.