Towards Real-time Advanced Vision Analytics over the Cloud

Yiding Wang *HKUST*
Weiyan Wang *HKUST*
Junxue Zhang *HKUST*
Junchen Jiang *University of Chicago*
Kai Chen *HKUST*

Abstract

Advanced vision analytics are gaining in popularity for their key roles in the modern vision applications. Unfortunately the advanced applications which demand high computing resources and quality video input are currently unable to utilize high-performance cloud computing. This is due to the real-time interaction and inference accuracy requirements, along with the varied and limited wireless bandwidth.

In this paper, we discuss the challenges faced by advanced vision analytics over the cloud and present a real-time video streaming and cloud inference framework that achieves both low latency and high inference accuracy. We utilize backend model-aware deep neural network (DNN)-based super-resolution (SR) to recover the quality of degraded data on the cloud server which achieves minor analytics accuracy loss and $\sim$6.8$\times$ bandwidth reduction compared with previous work. We address several challenges of applying SR in a cloud streaming video inference framework, from both machine learning and system perspectives. We integrate our framework and backend analytics stacks to further reduce computation overhead.

1 Introduction

Driven by modern vision applications such as autonomous driving and robotics, many advanced computer vision topics and models are raised, e.g. multiple object detection [19, 20, 22], semantic segmentation [11, 27] and instance segmentation [7, 9, 23] on high-resolution scenes. Advanced vision applications require high-resolution data to ensure accuracy and fast inference for real-time interaction. Unlike traditional object tracking applications [3, 25, 26] which are already widely deployed over the cloud, latest real-time advanced vision applications e.g. autonomous driving adopts edge computing to avoid the network latency and accuracy degradation. Therefore future industry-level scalability could be restricted by the cost of expensive edge devices, e.g. on-car chips [13].

Cloud computing is a growing trend. Due to the limitations of edge devices and the large-scale deployment of vision analytics applications, DNN-based models are often deployed in data centers, serving inference requests with streaming data from clients over the cloud. Offloading heavy computing tasks from low-cost client devices to data centers can greatly improve the inference performance with advanced models, relax the hardware requirements for edge devices, and achieve large-scale deployment at low cost. However, the varied and limited bandwidth of wireless networks make it challenging to ensure the latency and quality for real-time video streaming and advanced vision analytics.

Our goal is to achieve high accuracy and low latency for the streaming video inference over the cloud. The straightforward degradation techniques to reduce bandwidth consumption and latency such as downsampling and reducing frame rate at fixed intervals [25] are not desirable. This is because degraded data hurts accuracy greatly in our target applications, e.g. small object detection and pixel-level semantic segmentation. Constrained by the latency and accuracy requirements in such a real-time and accuracy-critical setting, it is extremely challenging to let advanced vision analytics applications take advantage of the various benefits of cloud computing.

In this work, we present a cloud framework to satisfy both latency and accuracy requirements of such advanced applications. To reduce the data rate while retaining accuracy, we deploy a content-aware SR model on the cloud server to restore the quality of the video streams for inference. However, the vanilla SR model doesn't perform well on reconstructing small details e.g. distant pedestrians which are important for critical applications e.g. autonomous driving. We use the hybrid criteria to train SR, which consist of standard perceptual evaluation criteria and a novel analytics perception criteria. This can enhance the performance of SR for improving the inference accuracy of the backend analytics model. Besides the machine learning perspective, we found that directly applying SR in a cloud analytics system could result in unstable performance. We handle the uncertainty of SR through the system-level adaptive controller. We also integrate our streaming video inference framework with server analytics stacks to reduce computation overhead.

We train the SR model offline which takes various downsampled inputs, and the output is a super-resolved high-resolution video. Training the SR uses the same data as the vision analytics model training, so no extra data is required. The hybrid criteria training works in a fine-tuning method. First we train the base SR model with the difference between the original and the super-resolved video, which is a standard evaluation method. Then we fine-tune the model with the inference accuracy loss of SR output on the vision analytics model as showed in Figure 2. In this way we improve the small detail reconstruction of SR model which benefits the backend model. The efficient SR model can super-resolve the streaming video on the cloud server with very low overhead.

The multi-scale structure in the high-resolution vision analytics are used to speed-up computation [19, 27] or improve accuracy [16, 22]. We integrate our framework and this kind of models to reduce more computation overhead by reusing low-resolution data. Besides downsampling, we adaptively select useful frames for instance-level tasks with a 2-level frame selector to reduce both the video streaming latency and computation overhead while keeping good trackability. The integration of the framework and analytics stacks are stated in Section 3.2.

Our main contributions are:

1. Design the first cloud framework to serve advanced vision analytics applications over the wide-area network. We tackle the latency/accuracy trade-off with the enhanced super-resolution framework.

2. Address the challenges of applying SR in the cloud vision analytics framework from ML and system perspectives. We improve the accuracy of critical details for real-world advanced applications and handle the uncertainty of the learning-based components.

3. Integrate the streaming video inference framework with server analytics stacks and adaptively select the efficient knob policy to manage the varied wireless bandwidth.

2 Background & Related Work

2.1 What's New for Advanced Vision Analytics?

We consider that the vision analytics tasks which serve demanding applications and require high computing resources, high quality data, high inference accuracy and low latency are *advanced*. 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, real-time interaction requires low latency.

Advanced vision analytics models generally have higher inference overhead due to their complex structures and high-resolution inputs. For example, state-of-the-art real-time object detection model SSD [14] can run at 300×300 in speed of 59 FPS (frames per second), while real-time semantic segmentation model ICNet [27] runs at 27 FPS on a 2048×1024 resolution input on the same device.

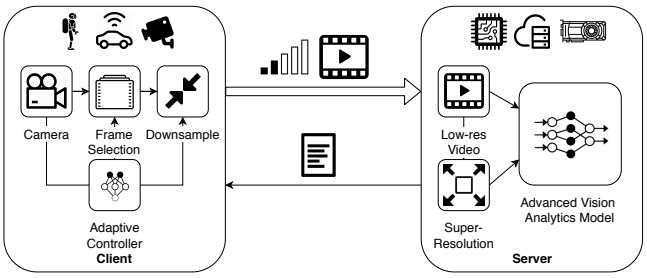

Figure 1: Framework overview

From a cloud perspective, for such advanced vision applications, the quality requirement for input data makes the latency/accuracy trade-off much more challenging. Generally object detection can achieve high accuracy using relatively low-resolution input [14]. Some applications e.g. pedestrian detection require high-resolution input, but bandwidth can be saved by reducing the frame rate [25]. Previous real-time cloud video analytics frameworks for object detection mainly focus on data degradation and knob policy, e.g. streaming the most important frames [3] or cropped areas [18] where objects are likely to appear to the server. Previous work also adopt an adaptive degradation strategy [25] in order to mitigate this trade-off. Resolution degradation barely works in advanced vision applications because high quality input is necessary to ensure accuracy. For instance-level recognition tasks, a higher frame rate than pedestrian detection is also required to ensure trackability because scenes change fast for a moving car.

Previous solutions for real-time object detection are not capable to tackle the trade-off faced by real-time cloud advanced vision applications because the bandwidth consumption can hardly be compressed. We compare our framework with the degradation method used by AWStream [25] in terms of bandwidth consumption in Section 4.2.

2.2 Video Streaming for Vision Analytics

AWStream [25] 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. Glimpse [3] reduces bandwidth consumption by sending only trigger frames based on the pixel-level changes between frames to the server. NoScope [10] uses a simple frame difference detector to select the frames that are likely to have objects to complex vision inference models. SimpleProto [18] proposes a server-driven framework that the server-side analytics logic determines how to stream the video around frame selection, area cropping and resolution.

2.3 Super-Resolution for Vision Analytics

Super-resolution (SR) reconstructs a high-resolution scene from a low-resolution scene. It has promising applications in video streaming [24] and vision analytics [6]. Recently deep neural network-based models [8, 12] gain great performance on this task. CARN [2] is a state-of-the-art real-time super-resolution model which implements a cascading mechanism upon a residual network for fast inference with accurate results.

Previous work [6] shows that SR can improve the inference accuracy on low-resolution video. We find an important insight that, besides the standard perceptual evaluation criteria, analytics perception criteria is also useful for improving the performance of vision analytics with SR. In this work, we enhance the SR with backend model-aware fine-tuning. We also observe that SR has its inherent uncertainty when applied in a cloud analytics framework. Through our system-level design, our framework can handle the worst cases caused by the unreliability of SR as stated in Section 3.3.

3 Design

Here we present our real-time video streaming and cloud inference framework for advanced vision analytics that achieves both low latency and high inference accuracy under limited bandwidth. A client device generates a high-resolution video and streams it to the server through a wireless wide-area network after processing. The server will also process the video then run inference with the DNN model. The server returns the inference results which can be encoded in the text files and are small in size compared with video. In this section, we illustrate the design of our 3 key components which are backend model-aware super-resolution, framework integration with analytics stacks and the adaptive controller.

3.1 Backend Model-aware Super-Resolution

Our framework extends the state-of-the-art efficient super-resolution model CARN [2] to meet the challenges in serving advanced vision analytics applications over the cloud. We offline train the SR model on the same dataset which was used to train the backend vision model with an inference accuracy-oriented method and deploy the SR model in front of the video inference model on the server side. Through this, we minimize the accuracy loss while drastically reducing the streaming latency.

Originally the super-resolution model CARN is trained with pairs of encoded high-resolution (HR) and low-resolution (LR) frames. The only target metrics is the structural similarity of the original HR frame and super-resolved (SR) frame, as illustrated on the left-hand side of Figure 2. Here we propose a new criteria to train an SR model for specific vision tasks towards better analytics performance. As Figure 2 shows,

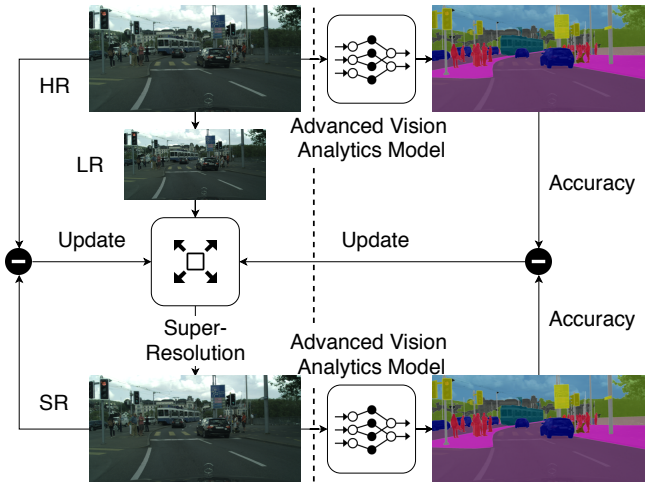

Figure 2: Train SR model offline with the new criteria

besides scene similarity loss, we add the inference performance loss of the backend vision model to update the SR model. From a machine learning perspective, this is similar to regularization. Here we use a semantic segmentation model ICNet [27] to illustrate the training pipeline.

CARN [2] adopts a patch-based CNN model. Patch is a 64×64 or larger subsection of an input image by randomly cropping, flipping and rotating it. For the backend model-aware training stated above, due to the different targets of SR and segmentation, the patch sizes are different, e.g. patch in ICNet is 720×720. In our applications, patching is useful to understand small features such as edges. We learn that a small patch is not optimal for accurate ICNet inference, so we can't directly run vision analytics inference during SR training with same batch of data. What's more, the advanced vision analytics training requires more effort on data annotation, so the dataset volume is smaller. For every 30 frames (1.8 s) the Cityscapes [4] dataset only provides one annotated frame with pixel-level ground-truth labels for semantic segmentation.

Since the training of the SR model does not requiring any annotation, we can utilize all frames from the training dataset to train an SR model based on traditional image similarity. Then we fine-tune the model with annotated frames based on semantic segmentation accuracy which is measured by the mIoU scores. The improvement is showed in Section 4.

3.2 Vision Analytics Stacks Integration

Besides using super-resolution for general advanced vision applications, we make special optimizations for models using two real-time machine learning techniques and integrate them with our framework. They are instance-level recognition models with *key frame feature propagation* for high-frame-rate input, and models with *downsampling branches* or *multi-scale structure* for high-resolution input.

2-Level Frame Selection For instance-level tasks [7, 23], skipping stale frames can save bandwidth while keeping high trackability. In real-time video analytics models [11, 21, 28], key frame feature propagation can save computation overhead. Here we define the frames which are necessary to stream as *useful frames*, such that *key frames* can be seen as the most useful frames. Intuitively, when the video is experiencing rapid changes, useful and key frames are more concentrated than when the scene is stable, so the criteria for useful and key frames is the pixel deviation of the current frame from that of the previous useful or key frame.

One low-latency video semantic segmentation model [11] devises a small and fast neural network which takes the differences between the low-level features of the current frame and the previous key frame as input, and predicts the deviation. If the predicted deviation goes beyond a pre-defined threshold, the current frame is set as a key frame, instead of selecting key frames with fixed intervals or simple heuristics.

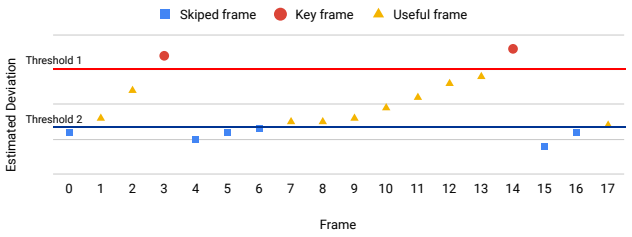

Figure 3: Adaptive 2-level frame selection

In our framework, to serve instance-level recognition applications with key frame feature propagation, we adapt this fast neural network to a 2-level frame selector and deploy it on the client. It aims to save the bandwidth consumption by skipping stale frames without hurting trackability and works together with the super-resolution method.

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 such that they can be updated according to network conditions and application requirements.

For a general instance-level task, only useful frames will be filtered out by the lower threshold to save bandwidth. If the backend analytics model works with the key frame feature propagation technique, two thresholds both apply. Key frames and useful frames will be streamed to the server with tags. The 2-level frame selection on the client reduces the streaming and repeated computation overheads, since the backend model doesn't need to select key frames again on the tagged frames.

Reuse Low-resolution Data The multi-scale structure let the model process high-resolution input in different resolutions (image pyramid) to improve the detection performance for small objects [16, 22], or save computation while keeping accuracy [19, 27]. Here we take ICNet [27] as an example to show that how we integrate such models and our framework to reuse the low-resolution data received by the server and avoid the overheads of repeated super-resolving and downsampling.

ICNet achieves real-time inference by building an inference path that employs information in the low-resolution frames along with details from the high-resolution frames to achieve both low latency and high accuracy. For example, ICNet downsamples the 2048×1024 (HR) input by $2\times$ (MR) and $4\times$ (LR) respectively to feed the cascading neural network. We found that for such model with the downsampling technique, a naive server-side workflow to process the LR frames is to let the SR model upsample the LR input by $4\times$, then let ICNet downsample HR to MR and LR to run inference with its multiple branches. The backend model and our framework working separately introduces repeated computation and the data quality loss.

Integrating the framework and ICNet can fix this issue by reusing LR data. The SR model can directly super-resolve LR to MR and HR then feed all the three frames to ICNet without the downsampling process, as illustrated in Figure 1. For different input resolutions, the framework can apply the most suitable super-resolution and downsampling policy.

3.3 Worst Case Handling with Adaptive Controller

Applying super-resolution in the vision analytics framework generally can handle the latency/accuracy trade-off as shown in Section 4, but there are still bad cases, even the average of results over a period of time is good. From our observation, this can be caused by variance of 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) in the same setting of Section 4 on a 30-second clip. Such extremely low accuracy (≤ 0.6) is unfavorable for the real-world applications, even the average is not that bad. It can be fixed by streaming a higher-resolution video to the backend model or even bypassing SR, as the red dashed line shows.

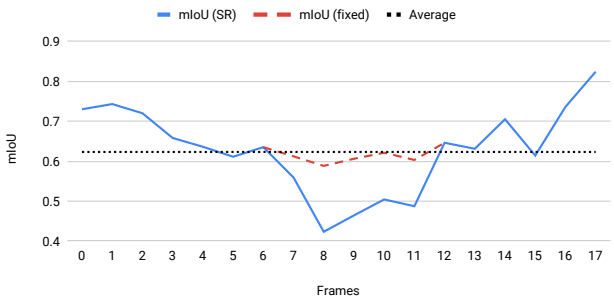

Figure 4: Variance of the inference performance with SR

We propose an adaptive controller to handle the variance of network conditions, the real-world scene changes and the performance drop of SR model. It gathers information from the transport layer e.g. bandwidth and network latency as well as the application performance from the application layer e.g. inference accuracy and computation latency. Through offline/online accurate 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. Our framework can avoid the worst cases caused by the uncertainty of the learning-based components and the environment.

4 Preliminary Results

We implement a prototype of our framework and conduct experiments on the Cityscapes [4] dataset. We use ICNet [27] which is designed for semantic segmentation as our backend model. Preliminary results show that our framework can achieve real-time advanced vision analytics over the cloud with low bandwidth consumption and accuracy loss.

4.1 Backend Model-aware Super-Resolution

We compare the similarity criteria (PSNR, SSIM) and the inference accuracy criteria (mIoU) of a semantic segmentation task using the SR model with and without analytics perception criteria fine-tuning. HR is the 2048×1024 frame. We get the LR frame by resizing HR to 512×256 with bilinear, which is the default resize algorithm of TensorFlow [1], and the video size is deducted by 13.3×. Then we upsample LR to the original resolution with three methods: bilinear, content-aware SR and backend model-aware SR. The standard inference model ICNet is trained on the Cityscapes [4] training set and mIoU is tested on the validation set. The mIoU of HR matches the performance claimed in the ICNet repository[1]. PSNR and SSIM are both calculated over the RGB channels, so the exact values are different from the original paper, which are calculated over the luminance channel. Our fine-tuned SR model achieves a better inference accuracy compared with the original SR. It improves the reconstruction of small details e.g. sharper edges of people in the distance which are important for the target advanced vision applications.

Metrics	LR-Bilinear	SR	SR-FT	HR
PSNR	31.00	35.21	35.44	—
SSIM	0.936	0.970	0.968	—
mIoU	0.582	0.633	0.649	0.675

Table 1: Performance of different upsampling methods

[1] https://github.com/hszhao/ICNet

4.2 Bandwidth Consumption

Cityscapes [4] dataset videos are 2048×1024 and 17 FPS, consisting of 8-bit RGB frames. Following the state-of-the-art streaming analytics framework AWStream [25], videos are encoded in H.264. In this setting, the original 2048×1024 video consumes 10 Mbps bandwidth. With the SR method introduced in our framework, a video can be adaptively downsampled by different factors. Here we downsample the video by 4× to 512×256. It consumes 750 kbps bandwidth, which is 13.3× smaller compared to original high-resolution video.

We further compare the bandwidth consumption of our framework with AWStream. Note that for a pixel-level semantic segmentation task here, we stream all the frames, and frames are only degraded on resolution. To achieve the same accuracy as our framework, AWStream can only downsample the video to 1440×720. It consumes 5.1 Mbps bandwidth which is 6.8× larger than ours, as shown in Figure 5.

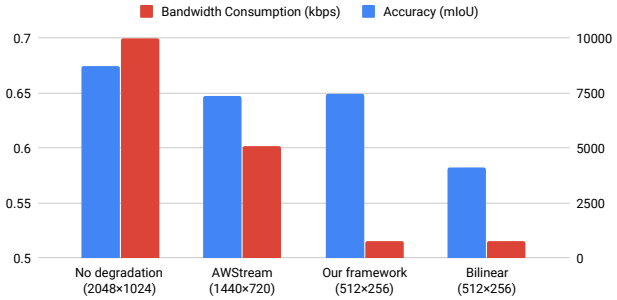

Figure 5: Bandwidth consumption to achieve comparable accuracy

4.3 Inference Latency

Besides network latency which is greatly reduced by our SR model, the major latency comes from the DNN model inference on the cloud server. 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, our framework can run in real time.

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

Table 2: Inference time per frame

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 [15, 17]. 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 [5]. We propose a method to handle the uncertainty to ensure the performance of the framework, and this is still an important topic for further research.

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.

Online improvement of the framework We can improve the performance of DNN-based models online on the server with the raw data as reference. The raw data also serves online profiling. If the client streams raw data, it is collected to train the SR model and the vision analytics model. The logic is decided by the adaptive controller considering the available bandwidth and inference 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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Vladimir Iglovikov, Selim Seferbekov, Alexander Buslaev, and Alexey Shvets. Ternausnetv2: Fully convolutional network for instance segmentation. *arXiv preprint arXiv:1806.00844*, 2018.

[10] 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.

[11] 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.

[12] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE conference on computer vision and pattern recognition (CVPR) workshops*, volume 1, page 4, 2017.

[13] 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.

[14] 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.

[15] 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.

[16] Mahyar Najibi, Bharat Singh, and Larry S Davis. Autofocus: Efficient multi-scale inference. *arXiv preprint arXiv:1812.01600*, 2018.

[17] 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.

[18] 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.

[19] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. Sbnet: Sparse blocks network for fast inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8711–8720, 2018.

[20] 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.

[21] 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.

[22] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9333–9343, 2018.

[23] 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.

[24] 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.

[25] Ben Zhang, Xin Jin, Sylvia Ratnasamy, John Wawrzynek, 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.

[26] 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.

[27] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[28] 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.