https://doi.org/10.2352/EI.2026.38.16.AVM-108

Low Power Automotive Vision Using Hybrid Sensors on NPUs

Sean Fausz, Kamal Rana, Austin Xiong, Shijie Xiao, Zhongyang Huang and Bo Mu; OmniVision Technologies, Santa Clara, CA/USA

Abstract

Automotive vision is a key component of advanced driver assistance systems (ADAS), enhancing road safety and improving vehicle operation for drivers. A critical requirement for automotive vision is achieving faster detections to ensure higher levels of safety. However, faster object detections using CMOS Image Sensors (CIS) are limited by their frame rate. While increasing the CIS frame rate enables faster object detection, it also results in higher sensor data rates and significantly increases power consumption. In our previous work [1], we demonstrated that utilizing event-based pixels—offering sparse spatial resolution but high temporal resolution—with low CIS framerate provides an effective alternative solution for faster object detections in automotive vision. Using hybrid sensor data (low CIS framerate + event-based sensor (EVS)) achieves comparable performance to high CIS framerate but with reduced data rates and power consumption. Specifically, in our previous study, we showed that using 7 fps CIS data combined with EVS data delivers the same performance as 20 fps CIS data, but with 40% lower data rate [1]. In this work, we implement post-training quantization (PTQ) and quantization aware training (QAT) techniques to automotive vision models trained on hybrid sensor data (CIS+EVS). This enables automotive vision models using hybrid (CIS+EVS) sensors to reduce both sensor data rates and power consumption during inference, particularly when deployed on Neural Processing Units (NPUs).

Introduction

In advanced driver assistance systems (ADAS) there are levels of driving automation provided and defined by SAE [12]. Level 0 represents no driving automation; an example of this level is an emergency braking system. Level 1 provides driver assistance features such as adaptive cruise control or lane keeping. Level 2 enables partial driving automation, including functions like a parking assist system. Level 3 introduces conditional driving automation, where the vehicle can handle tasks such as highway driving but requires the driver to remain on standby and ready to take control. Level 4 offers high driving automation, as seen in self-driving taxis operating in urban areas. Finally, level 5 represents full autonomous driving under all conditions. ADAS levels 0-2 are becoming standard and commonplace in new vehicles and the prevalence of ADAS levels 3 and 4 are starting to increase for select applications and uses. With the growing number and complexity of these features being incorporated into vehicles, the required bandwidth increases. For example, faster object detections using CMOS Image Sensors (CIS) are limited by their frame rate. While increasing the CIS frame rate enables faster object detection, it also results in higher sensor data rates and significantly increases power consumption. Therefore, the need to minimize automotive system bandwidth while preserving performance and safety becomes increasingly important in response to the growing number and complexity of features.

Event-based vision sensor (EVS) [3] is an attractive solution for ADAS. EVS pixels output events when a change in illumination is observed. These events are encoded to show where the event occurred (the pixel's row and column location), the timestamp of when the change happened, and whether the change was an increase or decrease (polarity). These events are asynchronous, giving high temporal resolution and lower latency characteristics [2]. We developed a hybrid (CIS+EVS) sensor [2] which integrates a CIS sensor and an EVS into a single device, enabling low power operation and high temporal resolution image reconstruction, such as super slow-motion video [6] and deblurring [7]. In our previous work [1], we demonstrated how such a hybrid sensor (CIS+EVS) can achieve object detection performance comparable to pure CIS while providing faster detection and reducing the data rate by 40%. We aim to expand upon this previous work and show additional places to save data rate and power.

In this work we show how overall system bandwidth can be reduced through the deployment on the edge, specifically with a neural processing unit (NPU). NPUs are specialized hardware specifically designed for machine learning. An NPU's architecture is optimized for common operations in neural networks such as matrix multiplications, which allow NPUs to achieve higher energy efficiency and better performance on quantized models [16]. With increasing ADAS levels, the data needed to be processed for real-time inference needs to be optimized to handle the larger bandwidth. An NPU can do this as it accelerates AI inference by utilizing parallel workloads, continuous low-power operation, and handling quantized models which consume less energy.

Post training quantization (PTQ) refers to reducing a model's numerical precision after it has been trained. For example, converting a model with floating point 32 (FP32) values to INT8. This greatly reduces power and decreases the model size, allowing better deployment but can incur some accuracy loss. However, PTQ does not require retraining, unlike quantization aware training (QAT), which performs quantization during the initial training process. The quantization nodes simulate INT8 during forward and backward passes, helping the model be more robust to quantization noise. This approach is more complex than PTQ and requires long training times, but it consistently provides the highest accuracy for quantized models [16].

In this work, we used our prior works dataset [1] that fused CIS and EVS data provided by the DSEC dataset [5]. For evaluation, we used the YOLOX-nano model for inference on GPU. We performed PTQ and QAT on that model to port it to an NPU. We compared the performance and power consumption between the GPU and NPU

models and observed a 10% performance drop in exchange for saving 99% power consumption. Our results showed that an automotive vision model deployed on an NPU can immensely save power with minimal performance drop. The use of an NPU with a hybrid (CIS+EVS) sensor offers a promising solution to mitigating demands in ADAS bandwidth driven by the increasing prevalence and complexity of these systems.

Figure 1: *Sample CIS Images from DSEC Dataset showing some of the different types of object classifications.*

Dataset

In this work, we utilized DSEC: A Stereo Event Camera Dataset for Driving Scenarios [5]. The dataset is publicly available from the DSEC dataset website. It contains driving dataset for various scenarios, including both conventional CIS frames (see Figure 1) and event data (see Figure 2).

The event data has spatial resolution of 640 × 480 and the RGB images are downsampled to match the EVS resolution. The dataset includes 20 frame rate per second with variable exposure time based on the scene and captured in a 12-bit raw format for CIS images. The dataset contains numerous sequences or clips of data captured but, in our work, we use only a subset of the available dataset. The sequences we use for training are from "zurich_city" the "16-21a" sets and we use "thun_02_a" data for testing. The dataset is stereo; however, we only use the data from the left camera.

We used the same modified hybrid dataset (CIS+EVS) as our previous study. To reiterate it here, we downsampled the 20fps CIS images to 7fps. We used these CIS images when available. In between the blind times of the CIS images, we upsampled back to 20fps by creating hybrid (CIS+EVS) frames. These were three-channel input images. One channel contained the EVS mask information, another held the grayscale CIS image of the most recent CIS image, and the third remained empty. Although we reused the most recent CIS image available, the EVS information was continuously updated at regular time intervals (see Figure 3). This ensured that even in the absence of new CIS frames, the model still received updated event-based information, enhancing detection accuracy and temporal consistency.

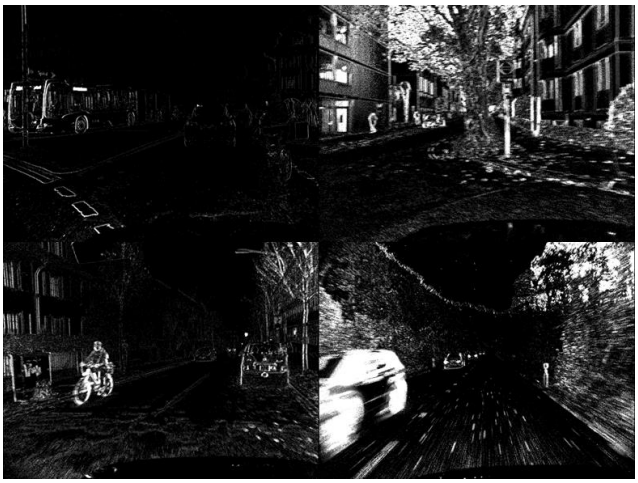


Figure 2: *Sample EVS images obtained from DSEC Dataset. These images are created by accumulating EVS data over a given time interval. Both polarities are added to one channel.*

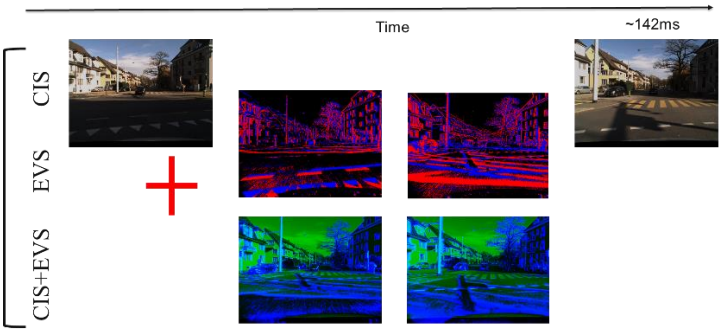


Figure 3: *The hybrid images from the 7fps CIS (grayscale)+EVS dataset were created by down sampling the 20fps CIS images to 7fps CIS images. Event data was grouped based on an accumulation time and added to the blue channel as an event mask for the hybrid frames over a consistent time interval. The previous CIS image is converted to grayscale and added to the green channel, while the red channel was left empty. This process is repeated for the next blind time between two CIS frames.*

Related Work

Most object detection models for automotive vision are trained on RGB images produced by CMOS image sensors, optionally with other forms of data, with a focus on detection accuracy and inference speed under various road environments and lighting conditions [8,9]. However, such works provide limited insight into the data transmission and power consumption of their implementations, most of which are GPU-based. These details are significant, as energy efficiency is a key constraint in practical deployment scenarios.

Prior work partially addressed these power efficiency concerns by leveraging hybrid CIS and EVS data, achieving reduced data transmission compared to pure CIS approaches [1]. Although this lowered the overall power consumption, the network was still implemented on a GPU system, which is not well-suited for deployment on resource-constrained edge devices.

GPUs have been the dominant platform for accelerating deep neural networks, but increasing demand for energy-efficient inference on edge devices has led to the emergence of NPUs as a promising alternative, specifically optimized for low-power neural network execution. Some previous works have implemented object detection models on a multi-neural processing unit system, focusing on system-level optimizations to achieve higher throughput and reduced latency, and compared these performance metrics against GPU counterparts [10]. However, no comprehensive analysis of overall power consumption between the NPU and GPU systems was provided, even though increased throughput is often associated with higher power usage.

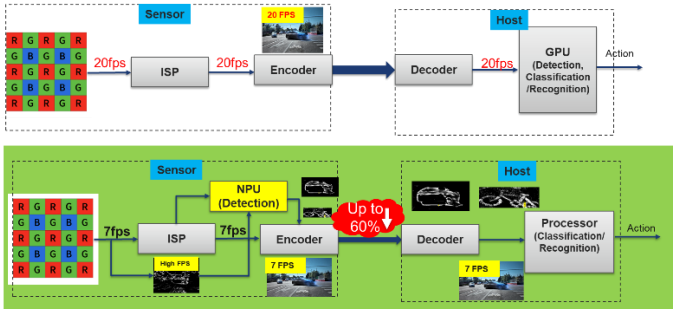


Figure 4: *The proposed system pipeline (bottom) shows how hybrid data will be input into an NPU to perform detections and outputs the low frame rate CIS images and just the event data of detected objects to the host process. This proposed pipeline will save data rate at the sensor by using the hybrid (CIS+EVS) sensor, save bandwidth by using the NPU, and significantly reduce the data transmission to the host by outputting just the event data inside a detection's bounding box in comparison to the standard pipeline (top).*

Methods

Our proposed pipeline aimed to save data rate at the sensor, overall system bandwidth by performing detections on an NPU, and data transmission to the host (see Figure 4). In our previous work we showed how a hybrid (CIS+EVS) sensor can achieve similar performance with 40% lower data rate. In this work, we aimed to expand upon the savings seen from the sensor side and save bandwidth through running inference for detections on an NPU.

7fps hybrid (CIS+EVS) data was input into this pipeline (see Figure 5) and sent to the NPU. The NPU used in this work is designed for ADAS applications and provides up to 2 TOPS of compute capability while supporting YOLOv3-based models, among others [13]. It achieves a throughput of 2048 MAC operations per cycle at 500 MHz, enabling efficient real-time processing [13]. The MAC input format uses an 8/16-bit dynamic fixed-point representation with a global 4-bit exponent, whereas the MAC output is represented in 16-bit fixed point with a global 4-bit exponent [13].

To implement object detection in real applications the model needs to be able to fit on hardware. In our previous work, we used YOLOv3, a large model with many parameters and operations (140.7 GFLOPS). This network is far too large to be on hardware for real-time inference. Therefore, we used a much smaller network YOLOX-nano (referred to as nano for conciseness).

The nano model has only 1.08 GFLOPS and is derived (YOLOX in general) from YOLOv3. The key architectural differences between YOLOv3 and nano include the detection head, anchor strategy, label assignment, and backbone/neck design. YOLOv3 uses a coupled detection head, whereas nano introduces a decoupled head consisting of three branches: a classification branch, a regression branch, and an IoU branch. YOLOv3 also employs anchor-based detection, requiring predefined anchor box dimensions obtained through k-means clustering and anchor tuning, while nano adopts an anchor-free approach that simplifies both training and inference. For label assignment, YOLOv3 relies on static IoU-threshold-based method, whereas nano uses SimOTA, a dynamic label-assignment strategy. In terms of architecture, YOLOv3 is built on the Darknet-53 backbone (where 53 represents the number of layers), while nano uses the lighter YOLOPAFPN with depthwise separable convolutions, enabling significantly reduced computational cost [14.] Ultimately, nano is designed for low-power inference and applicable for edge devices like NPUs.

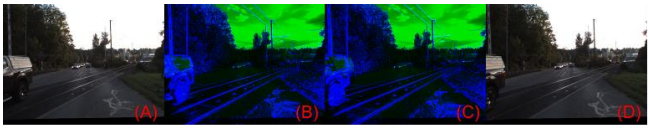


Figure 5*: Visualization of hybrid (CIS+EVS) data input to the pipeline. Image (A) shows a 7fps CIS image. Image (B) is a hybrid (CIS+EVS) image at the first time-interval in the blind time between the 7fps CIS frames. Image (C) is the next hybrid (CIS+EVS) in the CIS blind time. Image (D) is the following 7fps CIS image captured. For hybrid (CIS+EVS) images, the most recent CIS image is converted to grayscale and added to the green channel, event data (both polarities) are added to the blue channel, and the red channel is left empty.*

Instead of training from scratch, we used a pre-trained model on the COCO dataset. The COCO dataset has 80 classes, so we modified the model file by filtering out the classes we do not need and keeping the classes that are in our dataset (car, person, truck, bus, motorcycle, and bicycle). We then retrained from that modified model file with our custom hybrid (CIS+EVS) dataset to learn the weights needed for this specialized data. After training we ran inference on the model to get the performance score (mean Average Precision – mAP) on GPU. The inference average power was also measured.

To convert this nano model to NPU, PTQ was performed to reduce the model's numerical precision before QAT was implemented to retrain the model with quantization. We started with FP32 on YOLOv3 in our previous work. In conversion to nano the numerical precision was changed to FP16, and lastly, the NPU was run with INT8.

Results

We compared the object detection performance with the hybrid (CIS+EVS) dataset, as well as the difference in power consumption running inference on GPU and NPU. We compared performance using the standard measurement of mAP 0.5:0.95, denoting the average score at that range of IoU thresholds. For GPU power, we measured by logging the power usage every 100ms during inference and throttled the images to 20fps to get the average power.

The average power was then converted to the same units to that of the NPU power measurements (mW @ 1fps). We saw a slight performance drop of 10% for a 99% savings in average power consumption (see Figure 6).

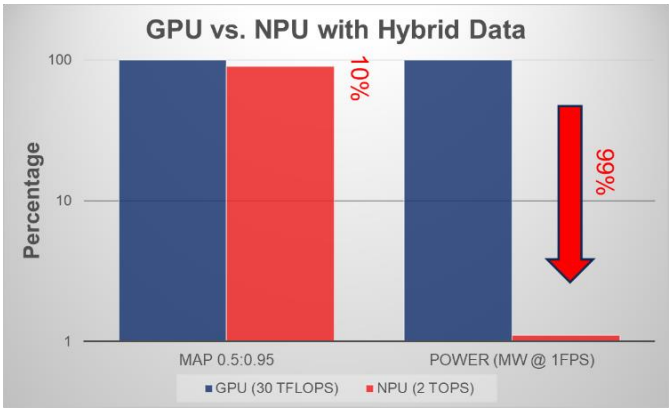


Figure 6: *Comparison of GPU against NPU in terms of performance and power consumption. By quantizing the model to run on NPU, power can be saved by 99% at the cost of a 10% decrease in performance. The GPU used has 30 Tera Floating Point Operations Per Second (TFLOPS). While the NPU that was used has 2 Tera Operations Per Second (TOPS).*

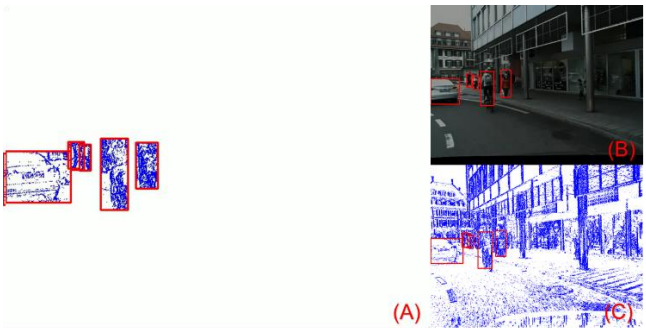


Figure 7: *Visualization of how transmitted data to the host can be reduced by performing detection on an NPU. Image (A) shows the detection results on NPU. The bounding boxes of the detection results are shown in red. We filter out the events outside of a detection bounding box. This is to show what we can send to the host minimizing the transmission of the data up to an estimated 60% depending on the scene. For comparison, image (C) shows the full event data and the background events that do not belong to a detection area. We still send CIS images in the pipeline (B). However, this is at 7fps compared to the more standard 20fps.*

As mentioned in the pipeline, we can decrease the data transmitted to the host through our hybrid (CIS+EVS) data. While a conventional pipeline would transmit, for example, 20fps CIS images, our proposed pipeline instead transmits only 7fps CIS images with low data rate EVS information of detected objects. We previously showed how this approach reduces the data rate by 40% on the sensor [1]. Furthermore, because object detection is performed directly on the NPU rather than the host processor—and due to the asynchronous nature of event data—we can further optimize the data transmission to minimize the amount of data that needs to be sent. This can be done by sending only the events that are found to be within a detection's bounding box area. This is shown visually in Figure 7.

One of the main reasons for the drop in performance from running detections on GPU in comparison to NPU comes from smaller objects not being detected. This is due to decreasing the model size and increasing the power efficiency. Some of these smaller objects can be detected on GPU, but with the quantization done on the NPU, enough numerical precision is lost that these more challenging objects score is too low or becomes lost and is not detected (see Figure 8). These more challenging objects may still be detected, but only by lowering the score threshold, which introduces more noise.

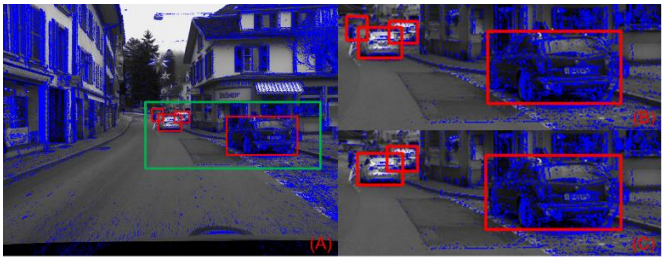


Figure 8: *Visualization of the performance drop from GPU to NPU. Image (A) shows the full scene with GT bounding boxes in red and a green box to show the zoomed in area for images (B) and (C). Image (B) shows the detections on GPU. There is a small object partially occluded that is detected. Image (C) shows the detections on NPU. The small object partially occluded is not detected.*

Conclusion and Discussion

In this work, we demonstrate the advantages of a hybrid (CIS+EVS) sensor and NPU system for automotive vision. The advantages seen come from comparing the data rate of the sensor, the overall system bandwidth, and the data transmission to the host.

With the increasing ADAS components along with the complexity of the ADAS, the power and bandwidth needed to support these features continue to grow. Therefore, the ability to reduce these metrics is crucial.

In our previous work, we showed the benefits of a hybrid (CIS+EVS) sensor, consisting of a mix of high spatial resolution CIS images along with high temporal resolution EVS data. The combination of this data brings similar performance (comparing 7fps CIS+EVS against 20fps CIS) in automotive vision but with faster detections and approximately 40% lower data rate [1]. We expand upon these findings to show the big picture of a low power automotive vision system.

An NPU, a specialized hardware, is optimized for machine learning operations. As a result, this edge device can run quantized models that not only reduce energy consumption but also improve overall energy efficiency. We took advantage of this and used an NPU to run inference on our hybrid (CIS+EVS) data. We performed object detection on a YOLOX-nano network on GPU with our hybrid (CIS+EVS) data. We took this model and performed PTQ and QAT to port the model onto an NPU so performance and power could be compared. Our results showed a massive 99% savings in power for the cost of 10% decrease in performance.

With the object detections being done by the NPU rather than the host, we explained how additional data transmitted can be reduced using a hybrid (CIS+EVS) sensor. In a traditional pipeline for example, 20fps CIS images are sent to the host to perform detections. Not only is this 20fps data larger in comparison to our hybrid (CIS+EVS), but it is difficult to be further reduced because CIS is in frames and the detections are still done on the host. With the NPU performing detections and the asynchronous nature of EVS, we can filter out the events that are not included in a bounding box area of a detected object. We estimate that this filtering, depending on the scene, can reduce up to 60% of the data needing to be transmitted to the host for further processing.

Overall, we showed a hybrid (CIS+EVS) sensor and NPU system can perform automotive vision at low power. We plan to continue to optimize the model for hybrid (CIS+EVS) data by looking at different ways to improve the data quality, optimizing the fusion of the CIS and EVS data, and changing the network architecture to be optimized for this hybrid data.

References

[1] K. Rana, *et al.*, "Automotive Vision using Hybrid Event Sensors," *Electronic Imaging, Autonomous Vehicles and Machines*, 2026.

[2] M. Guo et al., "A 3-Wafer-Stacked Hybrid 15-MPixel CIS + 1-MPixel EVS With 4.6-GEvent/s Readout, In-Pixel TDC, and On-Chip ISP and ESP Function," *IEEE Journal of Solid-State Circuits*, vol. 58, no. 11, pp. 2955–2964, Nov. 2023, doi: 10.1109/JSSC.2023.3303154.

[3] G. Gallego *et al*., "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[4] D. Gehrig and D. Scaramuzza, "Low-latency automotive vision with event cameras," *Nature*, 629(8014), pp.1034-1040, 2024, doi: 10.1038/s41586-024-07409-w

[5] M. Gehrig, *et al.*, "DSEC: A Stereo Event Camera Dataset for Driving Scenarios," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp.4947-4954, July 2021, doi: 10.1109/LRA.2021.3068942.

[6] R. Jiang *et al*., "EVS-Assisted Joint Deblurring, Rolling-Shutter Correction and Video Frame Interpolation Through Sensor Inverse Modeling," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2024, pp. 25172-25181, doi: 10.1109/CVPR52733.2024.02378.

[7] D. Saito, *et al.*, "Building End-to-End Deblur Image Quality Evaluation Simulation for Hybrid-EVS-CIS Sensor Images," *Electronic Imaging*, pp 240–1 - 240-7, 2025, https://doi.org/10.2352/EI.2025.37.9.IQSP-240

[8] M. Meyer and G. Kuschk, "Deep Learning Based 3D Object Detection for Automotive Radar and Camera," *2019 16th European Radar Conference (EuRAD)*, Paris, France, 2019, pp. 133–136.

[9] Y. Zhang, *et al.*, "Real-Time Vehicle Detection Based on Improved YOLO v5," *Sustainability*, vol. 14, no. 19, pp. 12274, 2022, doi:10.3390/su141912274.

[10] S. Oh, Y. Kwon, and J. Lee, "Optimizing Real-Time Object Detection in a Multi-Neural Processing Unit System," *Sensors*, vol. 25, no. 5, pp. 1376, 2025, doi: 10.3390/s25051376.

[11] C.-Y. Chan, "Advancements, prospects, and impacts of automated driving systems," *International Journal of Transportation Science and Technology*, vol. 6, no. 3, pp. 208–216, 2017, doi: 10.1016/j.ijtst.2017.07.008.

[12] SAE, " SAE Levels of Driving Automation™ Refined for Clarity and International Audience," SAE International. Access: Mar. 20, 2026. [Online]. https://www.sae.org/news/blog/sae-levels-driving-automation-clarity-refinements

[13] Omnivision, " OMNIVISION Advanced Digital Imaging, Analog, and Display Solutions," Omnivision Technologies. Assess: Mar. 20, 2026. [Online]. https://www.ovt.com/products/#asic

[14] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," *arXiv preprint arXiv:2107.08430*, 2021, doi: 10.48550/arXiv.2107.08430.

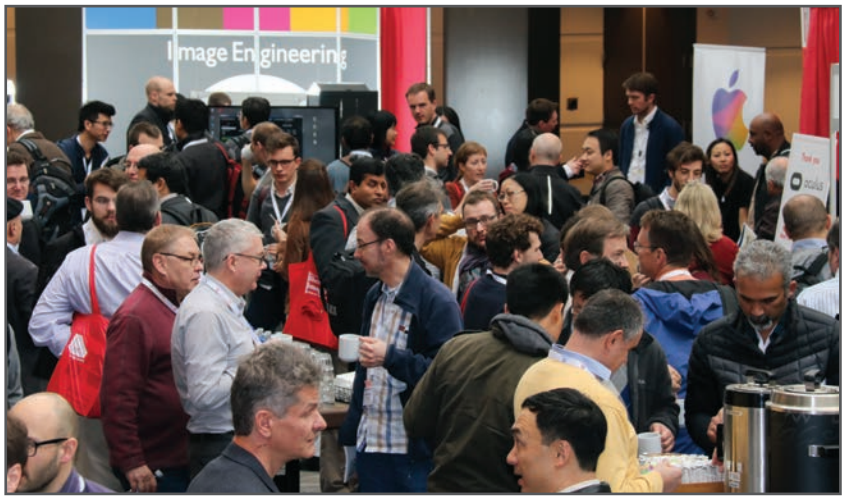
[15] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018, doi: 10.48550/arXiv.1804.02767.

[16] M. Nagel, *et al.*, "A White Paper on Neural Network Quantization," *arXiv preprint arXiv:2106.08295*, 2021, doi: 10.48550/arXiv.2106.08295.

JOIN US AT THE NEXT EI!

electronic IMAGING

Imaging across applications . . . Where industry and academia meet!



• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

ADVANCING IMAGING SCIENCE 75 1947 IS&T 2022 imaging.org