Rethinking distributed ML at the edge for automotive: an empirical study

Simone Mangiante
Vodafone Group R&D

Guenter Klas
Vodafone Group R&D

Over the last few years, the telecom industry has brought two major innovations into public mobile networks: 5G and edge computing. Car manufacturers realise that some computation can be offloaded from a vehicle to the network in order to keep power consumption and retail cost low, whilst complementing the vehicle's own mission-critical and autonomous on-board systems [1, 3]. One of the most debated use cases is computer/machine vision at the network edge to advance the performance of assisted and automated driving [2]. Our hypothesis is that edge computing, if suitably distributed within mobile networks, can be beneficial in augmenting the in-vehicle capabilities. More specifically, we seek an answer to this question: **can edge computing effectively bring benefits to in-car Machine Learning (ML)?**

We introduce an architecture that aims to opportunistically (i.e. when connectivity is available) complement in-car computing with high quality data from the network. In our scenario cars perform a single machine vision task (cars and pedestrians detection) and are connected through cellular radio. The *Device* (vehicle), the *Edge* of the mobile network, and the *Cloud* execute state-of-the-art ML models [5, 7] *well suited* for their running environment and pre-trained on relevant datasets like COCO [6] and KITTI [4]. Since the three locations will always differ in compute resources, it is futile to normalise them or explore an endless list of features: we call Object Detector Module (ODM) the particular combination of ML model, location and hosting infrastructure.

To assess the contribution of ODMs to the overall problem solving, we implemented a Device (in-car) Python application which captures frames from a dashboard camera and can process them locally or send them to the Edge and Cloud (in JPEG format) through a commercial 4G wireless connection. We set a minimum accuracy threshold of **0.9** which the three ODMs must exceed for a detection to be counted as *High-Confidence Detection* (HCD). Focusing on HCDs only, we can easily work on the metric of cumulative number of HCDs, since at that high confidence we assume ODMs never report false positives. In order to keep the cost of in-car equipment low, the Device mounts an NVIDIA Jetson Nano, one of the cheapest GPU boards able to achieve a near real-time video object detection. Our Edge is located in AWS London (`eu-west-2`), the closest AWS region to where we are (South-East England). The AWS Rekognition service in AWS Dublin (`eu-west-1`) acts as our Cloud. As ODMs process data in parallel at their own maximum throughput (driven by the frame rate of the camera on the vehicle), they compete for making the highest value contribution. As expected, the Device managed to process more frames than the Edge and the Cloud in the same time period. In order to perform a fairer comparison between ODMs we submitted 3992 randomly chosen frames to all three modules and measured per-frame, per-ODM HCDs and response time. Edge and Cloud ODMs are more accurate than the Device, thanks to better models and more compute resources: the Edge reported more HCDs than the Device in 79% of total frames. This value confirms the better accuracy of the Edge ODM, but does not take into account its non real-time, more variable response time.

An autonomous vehicle will value HCDs more than low-confidence ones. Equally, to a car real-time insight is likely more valuable than stale information. Most research in deep learning for object detection evaluates model performance through separate metrics measuring inference speed (typically FPS) and detection precision (typically mAP) [5]. In contrast, we aim to assess relative ODM contributions through a single metric, realising that none of the ODMs will ever be perfect in both traditional metrics (high FPS *and* mAP). In the extremes, the Device struggles with mAP due to resource constraints, whilst the Cloud struggles with FPS due to network latency. In order to take into account both HCDs accuracy and response time in a single value, we defined a composite metric which shows that, in our collected dataset, the Edge was capable to add value to the vehicle. Expert automotive stakeholders can tune this metric by varying a parameter and make it tougher or more performance demanding for the Edge.

Our findings suggest it is indeed possible to enhance the overall system performance of automotive use cases like object detection by leveraging edge computing in mobile networks for the execution of high-performance ML models.

References

[1] Giuseppe Avino, Paolo Bande, Pantelis A Frangoudis, Christian Vitale, Claudio Casetti, Carla Fabiana Chiasserini, Kalkidan Gebru, Adlen Ksentini, and Giuliana Zennaro. A mec-based extended virtual sensing for automotive services. *IEEE Transactions on Network and Service Management*, 16(4):1450–1463, 2019.

[2] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

[3] Mustafa Emara, Miltiades C Filippou, and Dario Sabella. Mec-assisted end-to-end latency evaluations for c-v2x communications. In *2018 European Conference on Networks and Communications (EuCNC)*, pages 1–9. IEEE, 2018.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.