In-Car Cognition with Edge Artificial Intelligence Accelerators

Ming-Chih Lin, Sio-Sun Wong, Yu-Kwen Hsu, Ti-Hua Yang, Pi-Sui Hsu, He-Wei Lee, Wei-Chen Tsai, Hsiu-Tzu Wu, Sung-Lin Yeh

Cloud + AI, Microsoft

{tolin, devinwon, yukwenh, anneyang, v-sunhsu, v-timlee, v-catsai, v-joaw, v-bryeh}@microsoft.com

Abstract

With sensors, computer vision, and AI, autonomous vehicles are becoming a reality that provides different levels autonomous. Although most of the cognition for vehicles focuses on external environment, some automotive manufacturers start thinking about using these technologies for in-car cognition to bring more value-added services. We received inquiries from car manufacturers to build an In-Car Cognition (ICC) component running on Internet of Things (IoT) edge device with Azure IoT Edge, where ICC component needs to detect and analyze passengers and specific objects within the vehicle. We built five different solutions using different kinds of software, models, and hardware, including Microsoft Azure Cognitive Service Containers, Azure Custom Vision, Open Neural Network Exchange (ONNX) models, ONNX Runtime, Intel Open Visual Inferencing and Neural Network Optimization (OpenVINO), Dlib / OpenCV, etc. running on Intel based and ARM based devices with AI accelerators.

Introduction

With sensors, computer vision, and artificial intelligence, autonomous vehicles are becoming a reality that provides different levels autonomous. Although most of the cognition for vehicles focus on external environment, some automotive manufacturers start thinking about using these technologies for in-car cognition to bring more value-added services. We received inquiries from car manufacturers to build an In-Car Cognition (ICC) component running on edge device to detect and analyze passengers and specific objects within the vehicle with the features below:

- Each detected / perceived passenger should be recognized
- His / her emotions (neutrality / happiness / surprise / sadness / anger / disgust / fear / contempt) should be analyzed, and the age (baby / child / teenager / adult) and gender (female / male) estimated.
- According to the position of the face, the respective seat in the vehicle should be assigned.
- If no passenger is detected on a seat (no perception of a face), the corresponding seat should be analyzed for animals (dog / cat).

Approach

We evaluated Microsoft tool chains and some well-known methods from community and identified the possible classification / detection models. The models and services we used for our solution includes: Azure Cognitive Service Containers, Open Neural Network Exchange (ONNX) models from Azure Custom Vision (Salvaris et al. 2018) and ONNX model zoo, Intel Open Visual Inferencing and Neural Network Optimization (OpenVINO) models from OpenVINO model zoo, MTCNN face detector (Zhang et al. 2016), and Dlib (King 2009) / OpenCV (Bradski and Kaehler 2008) for face detection and identification.

To deploy and manage the ICC solution, the solution is designed to be deployed with Azure IoT Edge, which is a fully managed service built on Azure IoT Hub, and can deploy AI, services, and business logic, etc. to run on Internet of Things (IoT) edge devices via standard containers as shown in Figure 1.

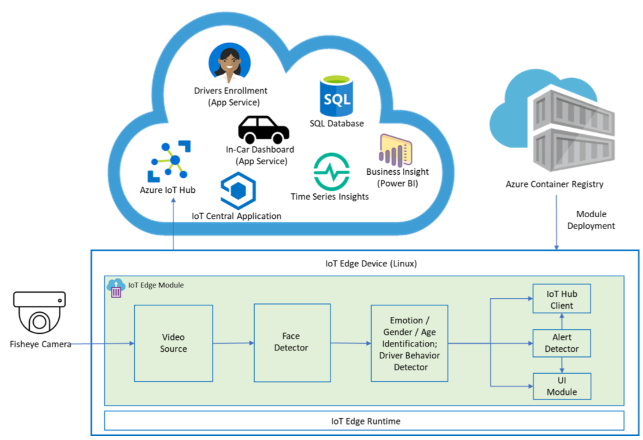

Figure 1: The architecture of ICC

Results

We built five different solutions using different kinds of software, models, and hardware, including Azure Cognitive

Service Containers, Azure Custom Vision, ONNX models, ONNX Runtime, Intel OpenVINO, Dlib / OpenCV, etc. running on Intel-based and ARM-based devices. The 1st one is using Azure Cognitive Service Containers, running on an Intel NUC edge device. The 2nd one is using pure ONNX models with ONNX Runtime, running on an Intel NUC edge device. The 3rd one is using the models from Intel OpenVINO, running on an Aaeon UP2 edge device with Intel Movidius Myriad X. The 4th one is using the mixed solution including Dlib / OpenCV and ONNX models, running ARM based Nvidia Jetson Nano. The 5th one is Quanta Q22 with Ambarella CV22 system on chip, which is an ARM based device that combines image processing, video encoding, and computer vision processing in a single, low power design. Table 1 shows the performance of the solutions we built. The cell with different colors means the inference time. All of them are running with Azure IoT Edge.

Solution	**Microsoft Azure Cognitive Service Containers**	**ONNX models with ONNX Runtime**	**Intel OpenVINO**	**Nvidia Jetson Nano**	**Ambarella CV22**
Device	Intel NUC (AMD64)	Intel NUC (AMD64)	Aaeon UP2 (AMD64)	Nvidia Jetson Nano (ARM64)	Quanta Q22 (ARM64)
Face Detection	Face container	Face detector model	face-detection-adas-0001	Dlib / OpenCV	MTCNN
Age		ONNX model	age-gender-recognition-retail-0013	ONNX model	N/A
Gender		ONNX model	age-gender-recognition-retail-0013	ONNX model	Dlib
Emotion		Emotion FerPlus	emotions-recognition-retail-0003	Emotion FerPlus	Dlib
Face Identification	Face container	ArcFace	face-reidentification-retail-0095	Dlib / OpenCV	N/A
Non-People Detection	N/A	Tiny YOLO v2	SSD MobileNet v2 COCO	Tiny YOLO v2	YOLO v3
Driver Distraction	N/A	N/A	facial-landmarks-35-adas-0002	Dlib / OpenCV	Dlib / OpenCV

Table 1: Performance of the solutions. Inference time:
■ < 100ms, ■ 100 - 500ms, ■ 500 - 1,000ms, ■ > 1,000ms

Azure Cognitive Service Containers provide an easy solution running cognitive services in edge device with containers. However, it requires high compute resources and has only AMD64 containers. In addition, there's no containers for non-people detection or driver distraction yet.

Pure ONNX solution with ONNX Runtime may run on different kinds of devices. However, the face detector ONNX model we use is more complex and thus it takes a longer inference time. In addition, different versions of ONNX models and ONNX Runtime may not be compatible and thus we cannot get ArcFace running in this solution.

OpenVINO solution uses the Intel OpenVINO models. However, the models cannot be converted into other formats and the hardware is limited to Intel related accelerators.

The mixed solution can run on ARM64 devices, and the ONNX models perform well with ONNX Runtime. However, it takes longer time for Dlib / OpenCV for face detection and face identification.

For the age and gender identification, we collected 23,334 faces from UTKFace (Zhang et al. 2017). For age identification, there are 18,879 adult, 1,885 baby, 1,503 child, and 1,067 teenager images. For gender identification, there are 11,163 female and 12,194 male images. The ONNX model is trained using Azure Custom Vision with general (compact) domain and evaluated using 3-fold cross validation with Precision, Recall, and Average Precision. The results illustrate Azure Custom Vision advanced training can achieve better results than the normal training.

Age	**Training with general (compact) domain**	**Advanced Training with general (compact) domain**
Precision	89.4%	**93.8%**
Recall	85.4%	**93.6%**
Average Precision	94.1%	**96.4%**

Gender	**Training with general (compact) domain**	**Advanced Training with general (compact) domain**
Precision	86.5%	**92.6%**
Recall	86.4%	**92.4%**
Average Precision	94.2%	**97.7%**

Table 2: Performance of the age and gender ONNX models

We also created custom dashboards to show the results from different IoT edge devices with Azure App Service, and a Power BI dashboard to show insights.

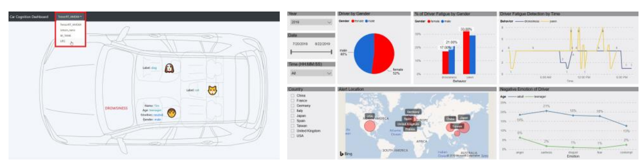

Figure 2: Dashboards

References

Bradski, G., and Kaehler, A. 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O'Reilly Press.

King, D. E. 2009. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* 10:1755–1758.

Salvaris, M.; Dean, D.; and Tok, W. H. 2018. Cognitive Services and Custom Vision. In *Deep Learning with Azure*. Apress, Berkeley, CA.

Zhang, K.; Zhang, Z.; Li, Z.; and Qiao, Y. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23(10):1499–1503.

Zhang, Z.; Song, Y.; and Qi, H. 2017. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*).