Evaluating Edge and Cloud Computing for Automation in Agriculture

Alberto Najera*, Harkirat Singh†, Chandra Shekhar Pandey‡, Fatih Berkay Sarpkaya‡, Fraida Fund‡, Shivendra Panwar‡
*University Heights High School, Bronx, New York, USA
†Francis Lewis High School, Queens, New York, USA
‡NYU Tandon School of Engineering, Brooklyn, New York, USA
{an3957, hs5076, cp3793, fbs6417, ffund, panwar}@nyu.edu

***Abstract*—Thanks to advancements in wireless networks, robotics, and artificial intelligence, future manufacturing and agriculture processes may be capable of producing more output with lower costs through automation. With ultra fast 5G mmWave wireless networks, data can be transferred to and from servers within a few milliseconds for real-time control loops, while robotics and artificial intelligence can allow robots to work alongside humans in factory and agriculture environments. One important consideration for these applications is whether the "intelligence" that processes data from the environment and decides how to react should be located directly on the robotic device that interacts with the environment - a scenario called "edge computing" - or whether it should be located on more powerful centralized servers that communicate with the robotic device over a network - "cloud computing." For applications that require a fast response time, such as a robot that is moving and reacting to an agricultural environment in real time, there are two important tradeoffs to consider. On the one hand, the processor on the edge device is likely not as powerful as the cloud server, and may take longer to generate the result. On the other hand, cloud computing requires both the input data and the response to traverse a network, which adds some delay that may cancel out the faster processing time of the cloud server. Even with ultra-fast 5G mmWave wireless links, the frequent blockages that are characteristic of this band can still add delay. To explore this issue, we run a series of experiments on the Chameleon testbed emulating both the edge and cloud scenarios under various conditions, including different types of hardware acceleration at the edge and the cloud, and different types of network configurations between the edge device and the cloud. These experiments will inform future use of these technologies and serve as a jumping off point for further research.**

I. Introduction

New technology in areas such as wireless networks, robotics, and artificial intelligence, can help increase productivity in agriculture and in similar applications. However, there remains the question of where the intelligence underlying these advances should be placed - should inference happen on edge devices, or in the cloud? A key factor influencing this decision is the required response time, which depends on the application - for example, an autonomous vehicle moving at high speeds is likely to require a very fast response time.

We consider both paradigms (illustrated in Fig. 1) with respect to the response time of the service. In edge computing, the intelligence is placed at the edge where the result of the computation will be translated into action, so there is no delay due to transferring data or results across a network. However, because the edge device is not as powerful, the time required for a machine learning model to return a result (inference time) may be longer. In contrast, a cloud computing service may have a faster inference time because the service runs on a powerful server, typically with GPU acceleration. But in addition to inference time, the total response time for the cloud service also includes network transfer time, since the input data is sent over a network to the machine learning model, and the prediction of the model is sent back over a network to the edge where it will be translated into action.

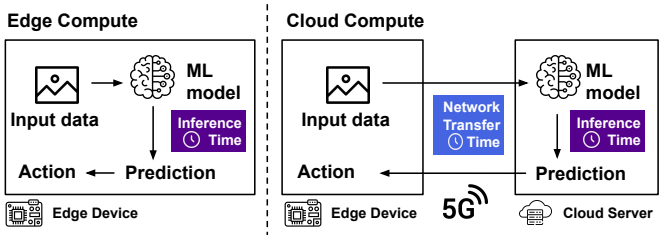

Fig. 1. Edge and cloud computing paradigms.

Since the overall response time depends on the network and compute resources that are deployed, the preferred option will depend on business considerations: what is the inference task, what is the required response time for this inference task, and what is the cost to deploy the network and compute resources necessary to achieve this response time. Previous studies have considered these tradeoffs for specific applications, for example [1] for an environmental science workload.

In this work, we consider the specific application of person detection in an outdoor agricultural setting, where machine learning models may be deployed on edge devices or accessed in the cloud via a private 5G mmWave network. We train a model for this task and then compute the response time for different deployment options, and report the results as a case study for consideration of these tradeoffs. All artifacts necessary to reproduce this work on Chameleon [2], a public cloud computing research testbed, are made available at [3].

II. Methodology

In this section, we describe the details of our experimental methodology.

Dataset: We used the National Robotics Engineering Center agricultural person-detection dataset [4], which includes images from the point of view of the front of a vehicle in an orchard settings. Each frame is labeled as "person" (in front of the vehicle) or "no person".

Model: Using the training subset of this data, we train an object detection model on Google's Teachable Machine [5] using its default settings. The trained model is available at [3].

Network scenarios: We use `netem` and `tc htb` to emulate a mmWave wireless network link, using traces collected and published in [6]. The data includes four trace scenarios: a static link, a link with short blockages (e.g. a person walking through the signal path), a link with long blockages (e.g. a person or vehicle walking through the signal path), and a scenario with both mobility and blockages.

Inference devices: We measured inference time on three device types, as shown in Table I. Raspberry Pi 4 and Coral Dev Board are considered edge devices. We included two devices capable of hardware acceleration (Coral Dev Board with TPU, and cloud server with NVIDIA RTX6000 GPU). The cloud server is on the CHI@UC site of the Chameleon [2] testbed, and the edge device is on the the CHI@Edge [7] site.

Device	Inference type
Raspberry Pi 4	CPU
Coral Dev Board	CPU, TPU
Cloud Server (RTX6000)	GPU, GPU + TensorRT optimization

TABLE I
DEVICE TYPES USED FOR MEASURING INFERENCE TIME.

Other inference optimizations: For the cloud server deployment, we also consider a model with TensorRT [8] optimizations for fast inference.

III. RESULTS

The results of our experiments are illustrated in Figure 2. The fastest overall response time is achieved by the Coral Dev Board edge device with TPU hardware acceleration. However, this deployment option would require a TPU-equipped edge device on each vehicle, which could be costly. When the model is deployed on edge devices *without* hardware acceleration, the response time is 40-50 ms. The cloud server equipped with GPU with TensorRT optimization also achieves a sub-10 ms *median* response time. However, during instances of blockage of the mmWave link, the response time could increase substantially, so this alternative would be acceptable only if occasional response times above 10 ms could be tolerated.

IV. ARTIFACT DESCRIPTION

The Github repository at [3] includes the following items:

- trained models in Keras and TFLite format
- a small set of images used for measuring data transfer time across the emulated network
- and detailed instructions for running our experiment on Chameleon [2].

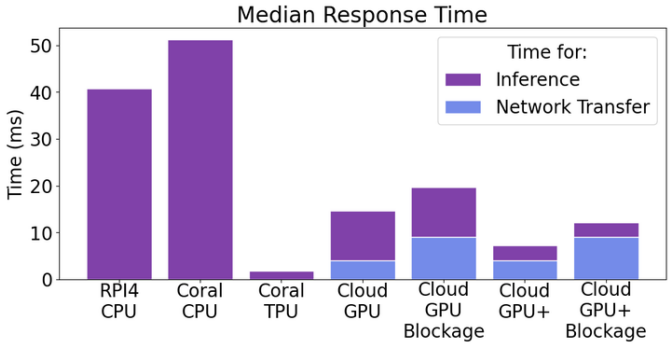

Fig. 2. Experiment results: median response time including inference time in both edge and cloud scenarios, and network transfer time for cloud scenarios.

These artifacts may be used in combination with other pre-existing artifacts for Chameleon [9], [7], [10] to reproduce our experimental results.

ACKNOWLEDGMENT

This project is supported by the New York State Center for Advanced Technology in Telecommunications (CATT), NYU WIRELESS, the ARISE program at the NYU Tandon Center for K12 STEM Education, the Pinkerton Foundation, and the National Science Foundation under Grant No. OAC-2230079.

REFERENCES

[1] J. Tsen, J. Anderson, L. Bobadilla, and K. Keahey, "One fish, two fish: Choosing optimal edge topologies for real-time autonomous fish surveys," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'21 Poster)*, 2021.

[2] K. Keahey, J. Anderson, Z. Zhen, P. Riteau, P. Ruth, D. Stanzione, M. Cevik, J. Colleran, H. S. Gunawi, C. Hammock, J. Mambretti, A. Barnes, F. Halbah, A. Rocha, and J. Stubbs, "Lessons learned from the chameleon testbed," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. USENIX Association, Jul. 2020, pp. 219–233. [Online]. Available: https://www.usenix.org/conference/atc20/presentation/keahey

[3] A. Najera, "Evaluating edge cloud computing for automation in agriculture," https://github.com/bert0bert/agricultural-automation, 2023.

[4] Z. Pezzementi, T. Tabor, P. Hu, J. K. Chang, D. Ramanan, C. Wellington, B. P. Wisely Babu, and H. Herman, "Comparing apples and oranges: off-road pedestrian detection on the national robotics engineering center agricultural person-detection dataset," *Journal of Field Robotics*, vol. 35, no. 4, pp. 545–563, 2018.

[5] M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru, and A. Chen, "Teachable machine: Approachable web-based tool for exploring machine learning classification," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–8. [Online]. Available: https://doi.org/10.1145/3334480.3382839

[6] A. Srivastava, F. Fund, and S. S. Panwar, "An experimental evaluation of low latency congestion control for mmwave links," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 352–357.

[7] K. Keahey, J. Anderson, M. Sherman, Z. Zhen, M. Powers, I. Brunkan, and A. Cooper, "Chameleon@edge community workshop report," 2021.

[8] H. Vanholder, "Efficient inference with tensorrt," in *GPU Technology Conference*, 2016.

[9] F. Fund, "Using cloud servers for GPU-based inference," https://chameleoncloud.org/experiment/share/3546a1d7-ea72-4b58-80eb-8cd95ff8965b, 2023.

[10] ——, "Network emulation," https://chameleoncloud.org/experiment/share/61d0c7e0-f932-4531-977f-5a88ff3c71d4, 2023.