



Perception Workload Characterization and Prediction on the Edges with Memory Contention for Connected Autonomous Vehicles

Presented By: Sihai Tang

Authors: Sihai Tang, Shengze Wang, Song Fu, and Qing Yang

Department of Computer Science and Engineering

University of North Texas

Outline

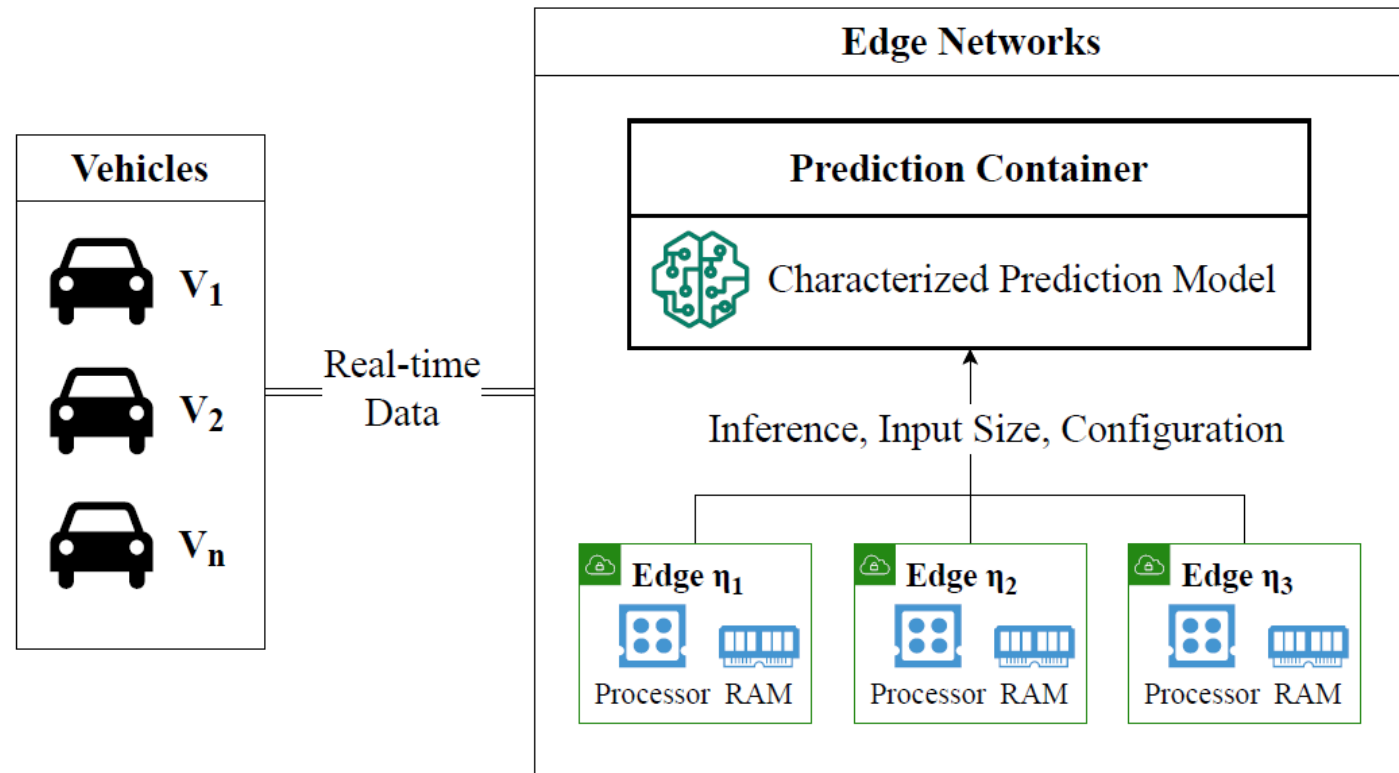
- Introduction and Motivation
- Preliminary Experimentation and Data Challenges
- Methodology
- Characterization and Findings
- Layer and Memory
- Conclusions and Future Works



Introduction

- Perception plays a vital role in the operation of Autonomous Vehicles (AVs), ensuring the safety and efficiency of these vehicles.
- Deep Neural Networks (DNNs) are the preferred choice for this module due to their accuracy and speed.
 - DNNs, such as YOLO, SSD, Faster RCNN, DeepLab, and LaneNet, are extensively researched and deployed for tasks like object detection and image segmentation.
- However, the deployment of DNNs as Edge workloads presents challenges.
 - Continuous Training Cycles, Expensive Data, Limited Resource Compared to Cloud....

Autonomous
vehicle to
Edge node
interaction



Challenges

- Limited Resources on Edge for CAV tasks
 - HD Maps, Fusion Detection, Bandwidth Saturation Tasks...
- Power Constraints, and Task Overload Queue
- When it comes to Perception workloads, variance in Edge platform and hardware is especially challenging.
- Traditional Techniques such as model or architectural optimization cannot keep up!

Motivation and Literature

- Traditional Scheduling
- Middleware with layer by layer
 - MASA, Deepeye and DART
- Architectural Enhancements
 - DAMO-YOLO

Begin	Interval a	Interval b	Interval c	Interval d (Detector)			
	Object Appears	Successful Capture	Image arrives at queue	Fetch	Fetch	Fetch	Inference
						Display	

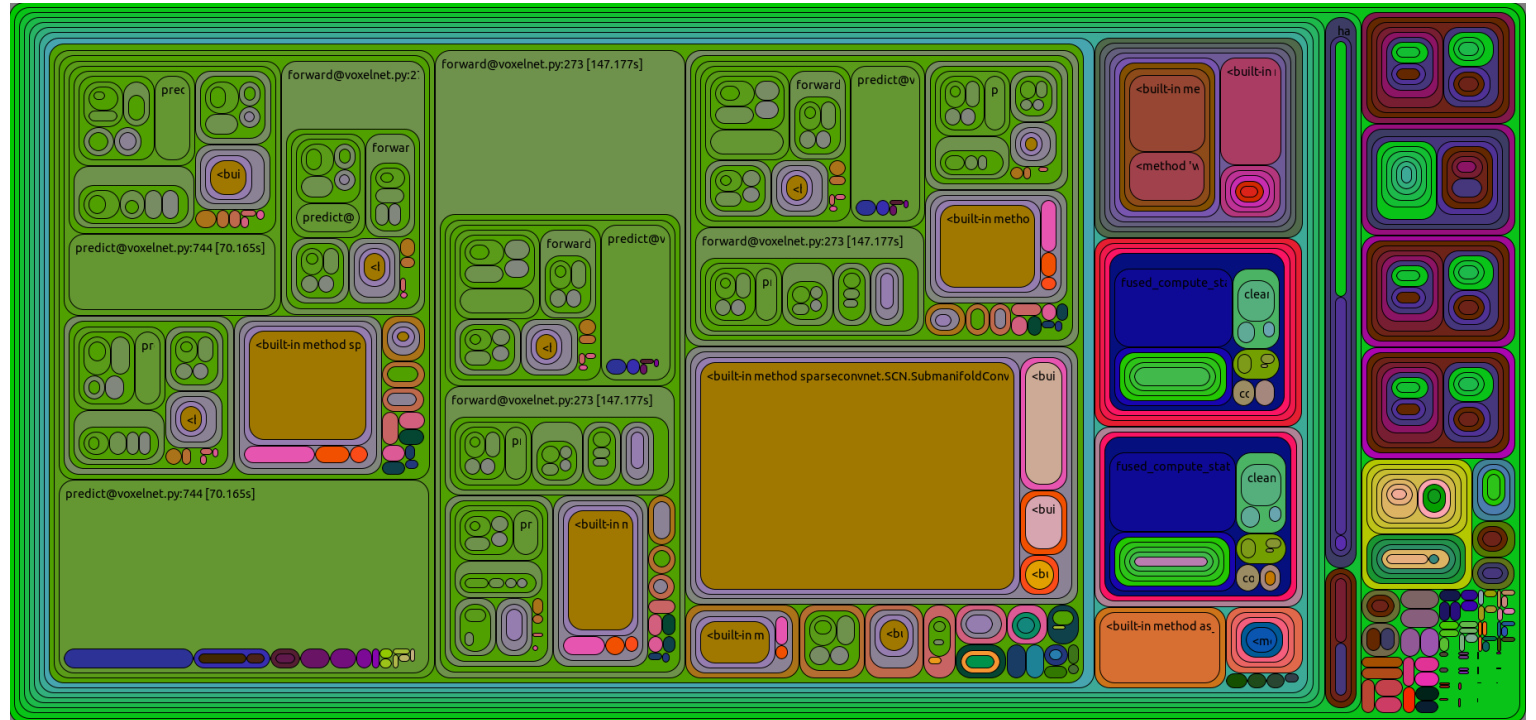
Time

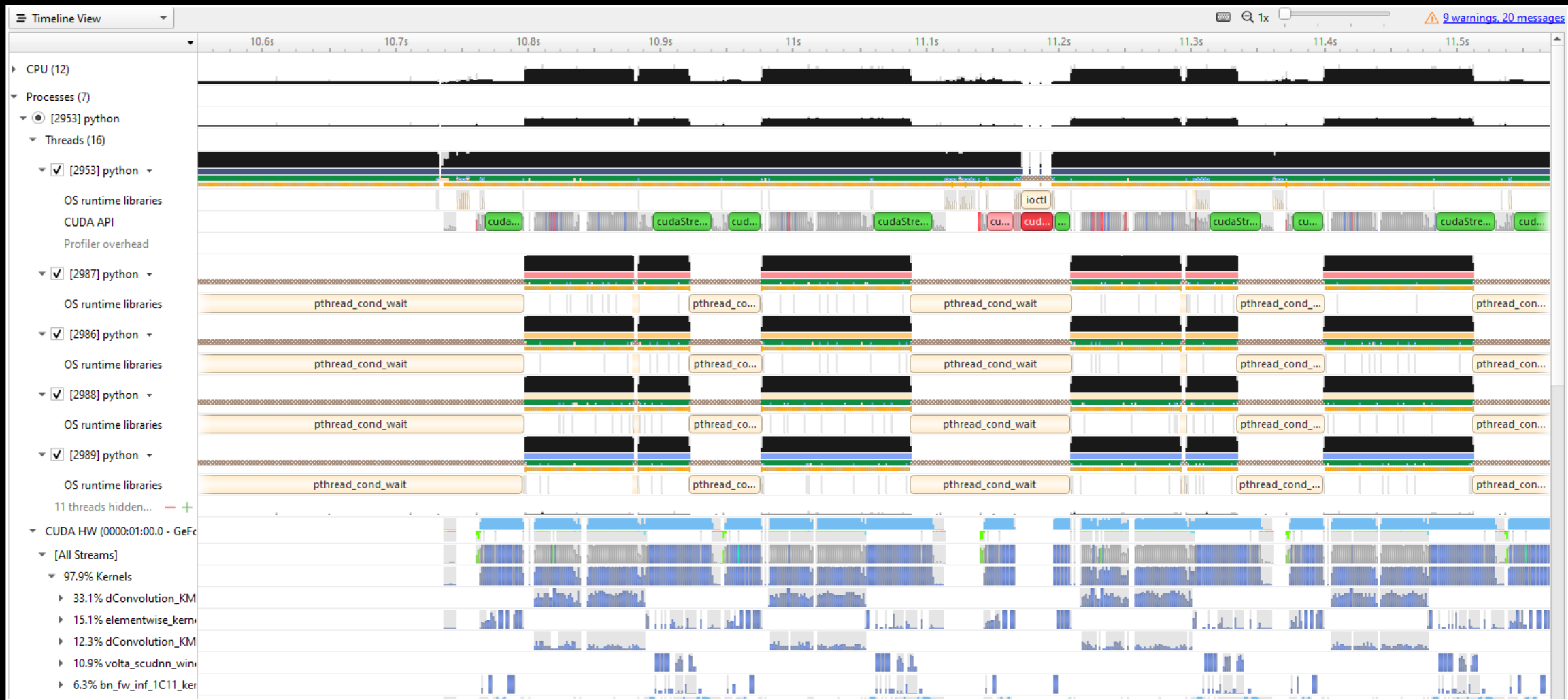
Optimization

Begin	Interval a	Interval b	Interval d (Detector)			
	Object Appears	Successful Capture	Fetch	Fetch	Inference	Inference

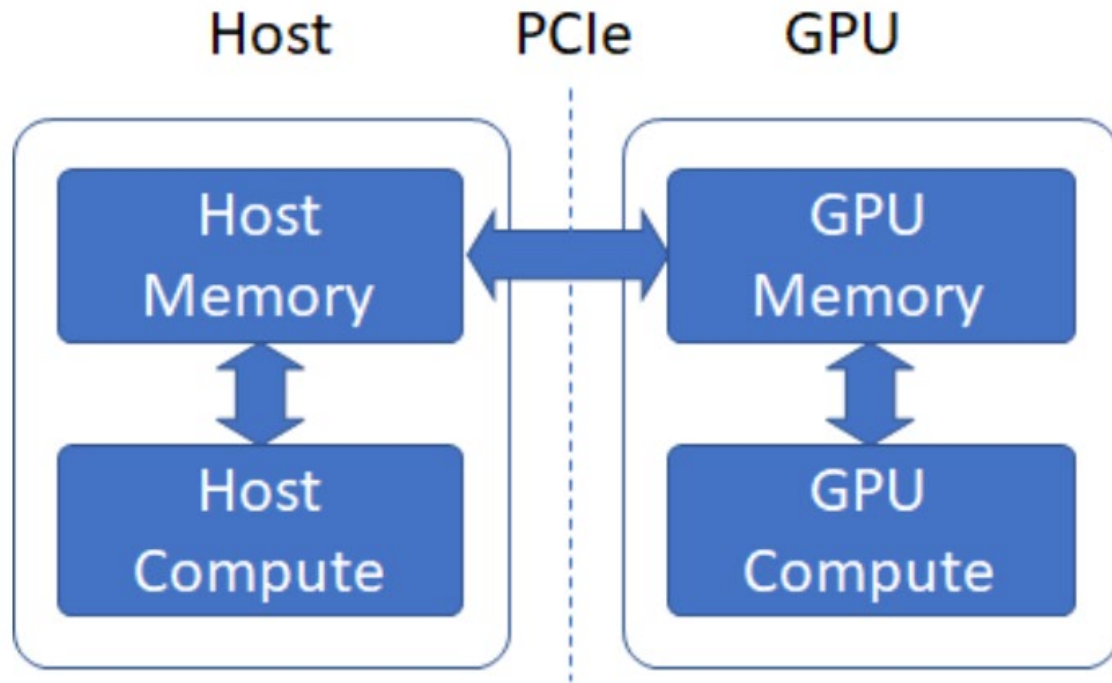
Time

Preliminary Characterization (CPU)





GPU Characterization



	Encoding	Transfer Rate(bits/sec)	Throughput (MB/sec)
PCI 1.0	8b/10b	2.5 Gb/sec	$8/10 \cdot 2.5/8 = 250$ MB/sec
PCI 2.0	8b/10b	5 Gb/sec	$8/10 \cdot 5/8 = 500$ MB/sec
PCI 3.0	128b/130b	8 Gb/sec	$128/130 \cdot 8/8 = 984.5$ MB/sec

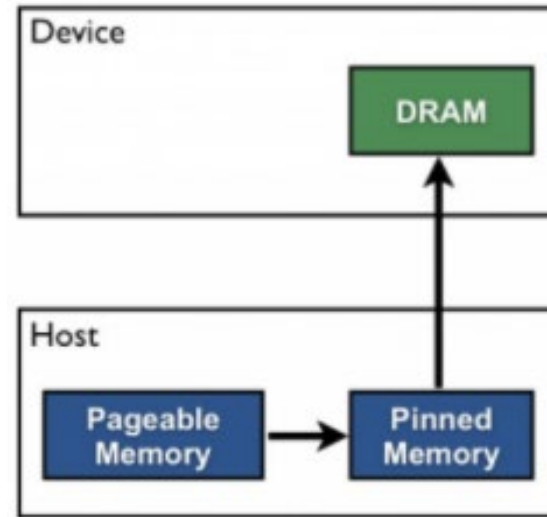
Optimization Analysis - Hardware

- Both hardware GPUs are PCIe Gen 3
- 1060 - 1 Lane ; 2070 - 2 Lane

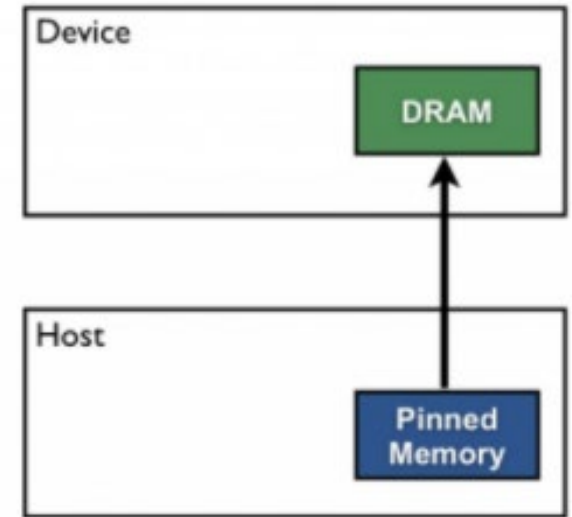
Optimization Direction

- In a theoretically perfect scenario, we can achieve the optimal speed to the right.
- But this varies with hardware and code
 - Can cause system instability if letting the system decide

Pageable Data Transfer

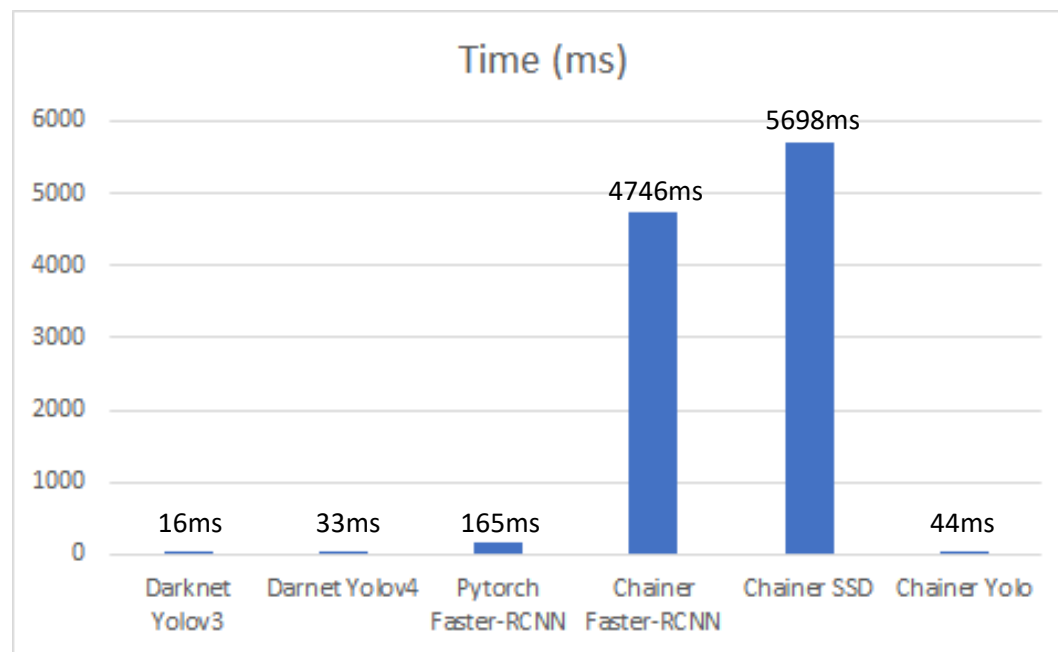
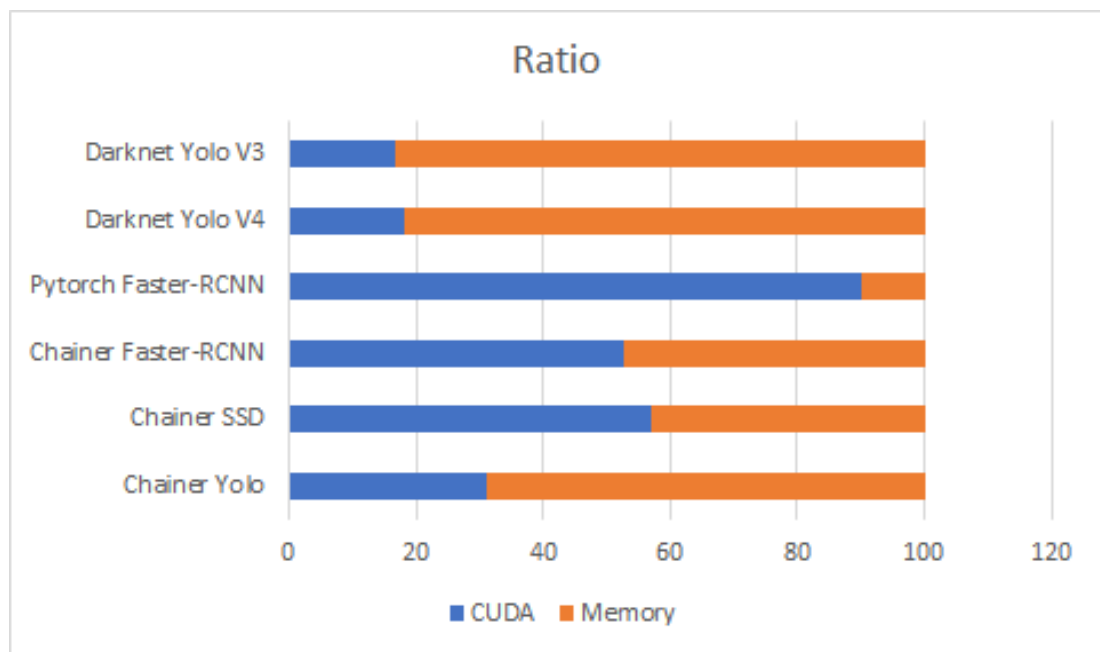


Pinned Data Transfer



Data Size (MB)	20	40	80	120	160
Transfer BW (page-locked) (GB)	11.36	11.44	12.02	12.09	12.13
Transfer BW (pageable) (GB)	5.06	5.31	5.39	5.35	5.04

Initial Results



Initial Modeling without Scenario Dividing

- It proved impossible to fully predict the behavior or explain the deeper attribute impacts based on the attributes alone.
- The impacts of Memory are very apparent, but the data cannot fight the skew in many modeling methods.

Regression and Classification

13 Methods used

- Linear Regression
- Gaussian Process
- Isotonic Regression
- Multilayer Perceptron (MLP) Regressor
- MLP Base
- MLP CS
- Pace Regression
- Radial Basis Function (RBF) Network
- RBF Regressor
- Simple Linear Regression
- SMOreg (SVM)
- Correlated Nystrom Views(XNV)

Attributes Tested:

Filter Number

Filter Size

Stride length

Input Size

Input Depth/Feature Dimension

Output Size

Output Depth

Target: BFlops

Regression and Classification

13 Methods used

- Linear Regression |0.82
- Gaussian Process |0.76
- Isotonic Regression |0.85
- Multilayer Perceptron (MLP) Regressor |0.88
- MLP Base |0.84
- MLP CS |0.84
- Pace Regression |0.80
- Radial Basis Function (RBF) Network |0.43
- RBF Regressor |0.87
- Simple Linear Regression |0.85
- SMOreg (SVM) |0.81
- Correlated Nystrom Views(XNV) |0.90

filters	Size	Stride	input_xy	input_depth	output_xy	output_depth	BF
32	9	1	102400	3	102400	32	0.177
64	9	2	102400	32	25600	64	0.944
64	1	1	25600	64	25600	64	0.21
64	1	1	25600	64	25600	64	0.21
32	1	1	25600	64	25600	32	0.105
64	9	1	25600	32	25600	64	0.944
64	1	1	25600	64	25600	64	0.21
64	1	1	25600	128	25600	64	0.419
28	9	2	25600	64	6400	128	0.944
64	1	1	6400	128	6400	64	0.105
64	1	1	6400	128	6400	64	0.105
64	1	1	6400	64	6400	64	0.052
64	9	1	6400	64	6400	64	0.472
64	1	1	6400	64	6400	64	0.052
64	9	1	6400	64	6400	64	0.472
64	1	1	6400	64	6400	64	0.052
28	1	1	6400	128	6400	128	0.21


Regression and Classification

13 Methods used

- **Linear Regression** | **0.82**
- Gaussian Process | 0.76
- Isotonic Regression | 0.85
- **Multilayer Perceptron (MLP) Regressor** | **0.88**
- MLP Base | 0.84
- MLP CS | 0.84
- Pace Regression | 0.80
- Radial Basis Function (RBF) Network | 0.43
- RBF Regressor | 0.87
- Simple Linear Regression | 0.85
- SMOreg (SVM) | 0.81
- **Correlated Nystrom Views(XNV)** | **0.90**

BF =

0.058	* size +
0	* input_xy +
-0.0002	* input_depth +
-0	* output_xy +
0.0004	* output_depth +
-0.0033	



Methodology of Characterization

- From the Preliminary empirical analysis, we found several factors that required deeper analysis for meaningful Characterization.
- Single Stage YOLOv3 and Two-Stage Faster-RCNN are chosen as the representative for each category of network.
- To characterize the workload behavior, we chose the following potential variables:
 - Processor Resource
 - RAM memory resource
 - Workload Size calculated from service type and input size
 - Time to process the workload

Setup and Scenarios

- To profile our ML methods, we simulate the high-end Edge nodes with a machine equipped with an Intel Core i7-10750H, Nvidia GeForce RTX 2070, 16 GB of DDR4 RAM, and a 1 TB NVMe SSD.
 - The total operating power constraint for the laptop is set to 250 Watts.
- For the lower-end Edge node, we opted for the Nvidia Jetson Xavier NX. It supports nine optimized power budgets to cap the CPU core numbers and their frequencies.
 - Power modes in our experiments include 20W, 15W, or 10W TDP with six, four, or two CPU cores.

Experimental Analysis (Low-End)

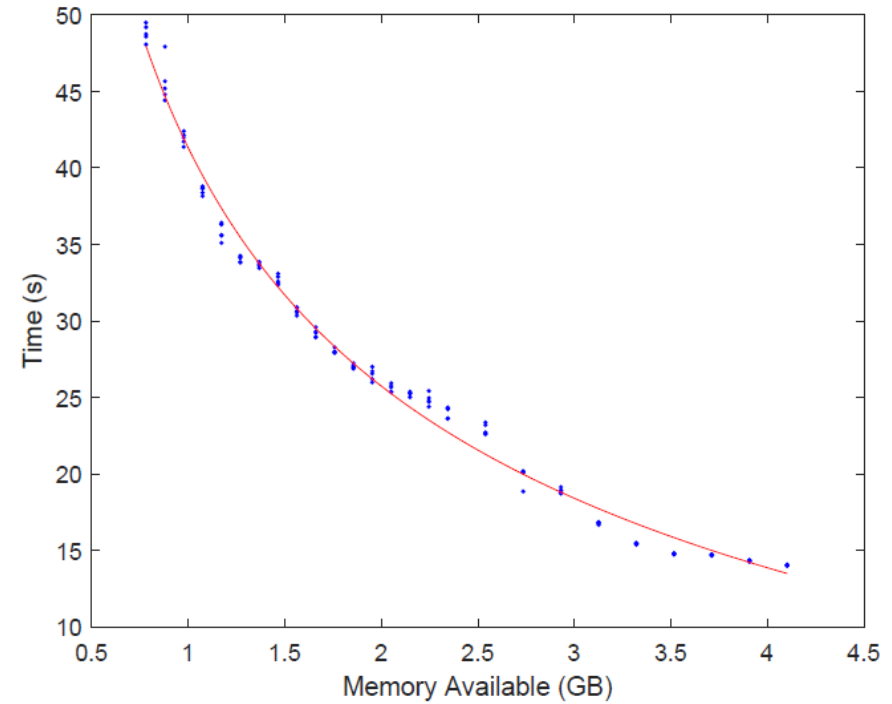
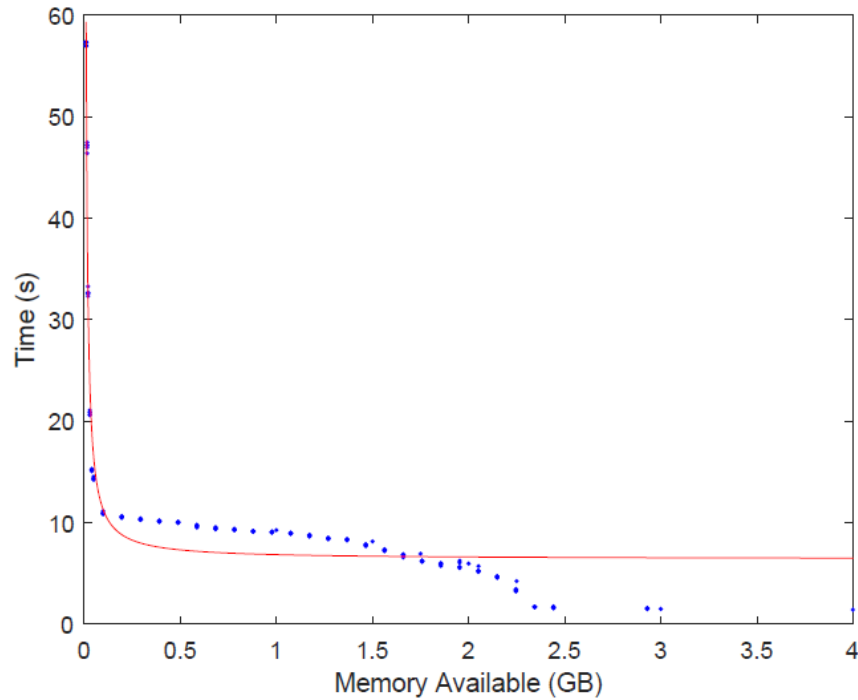


Fig. 4. Memory Contention on a low-end Edge between single-stage and two-stage perception. Left(a) represents YOLO and Right(b) represents Faster R-CNN

High-End

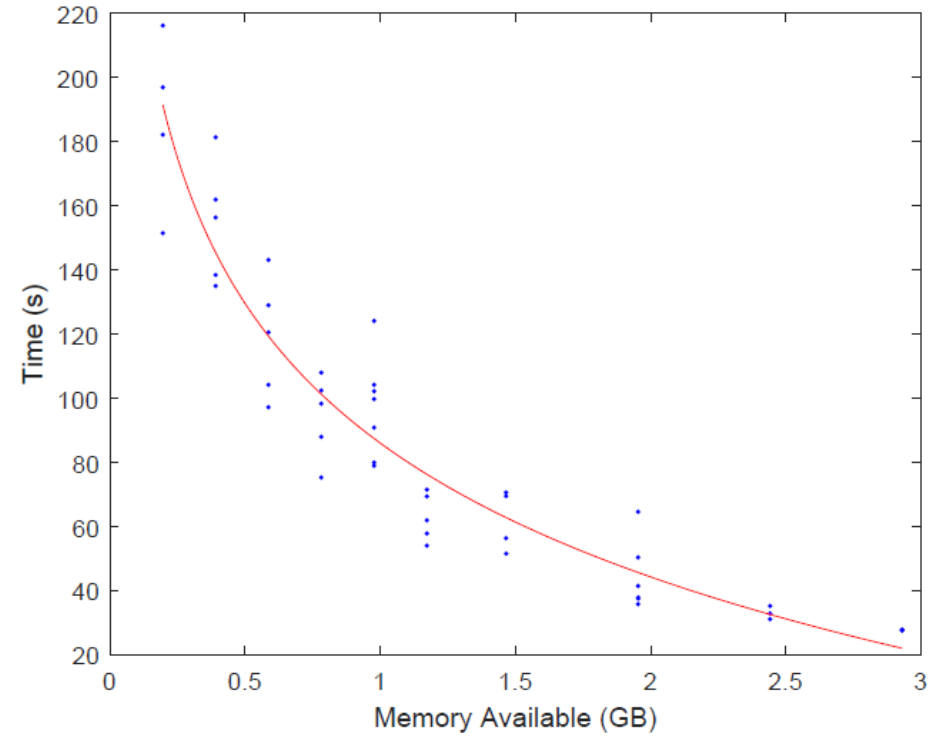
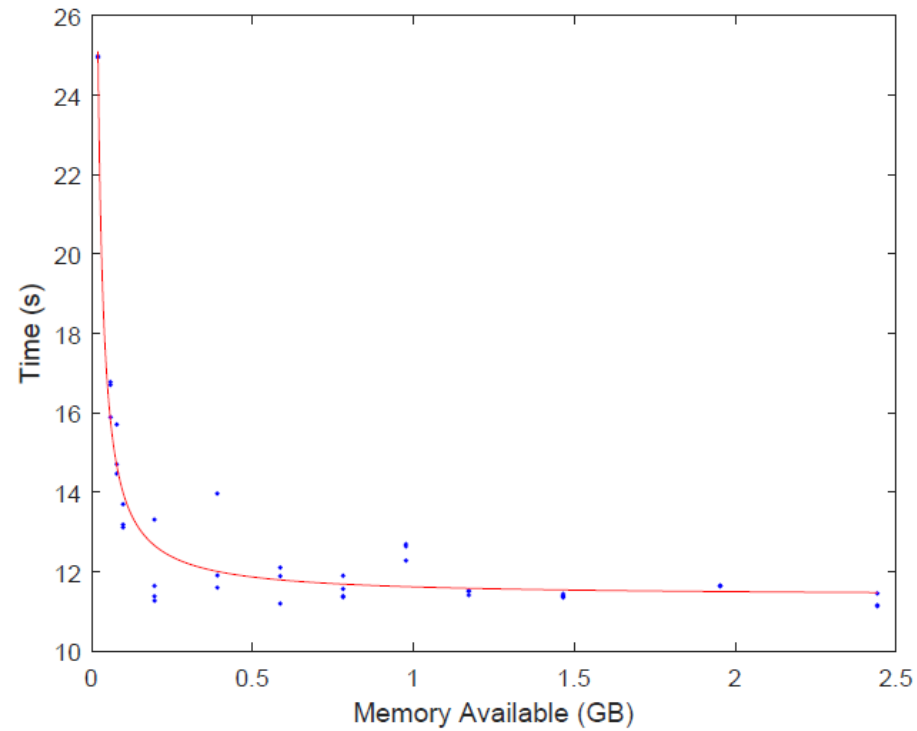
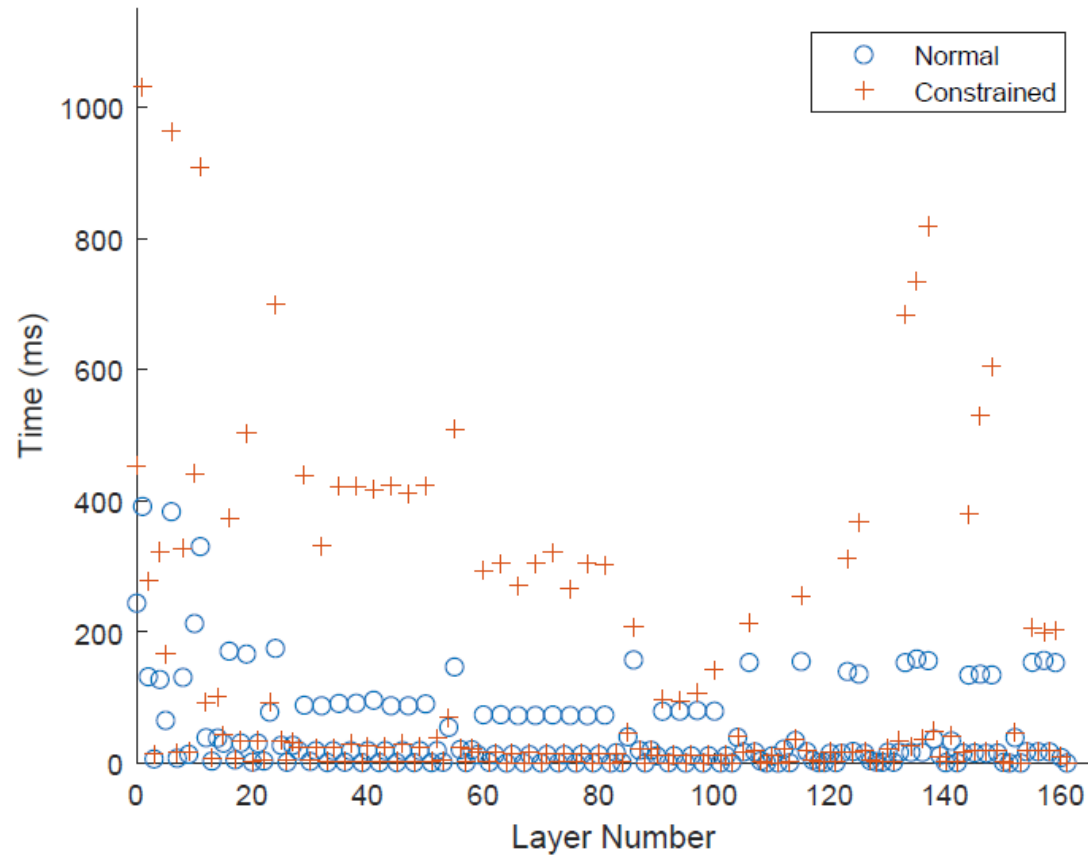


Fig. 5. Memory Contention on a high-end Edge between single-stage and two-stage perception. Left(a) represents YOLO and Right(b) represents Faster R-CNN

Layer by Layer



YOLO Layer	Normal	Contention	Percentage Increase
0 Convolution	0.2705s	0.4792s	77%
1 Convolution	0.5176s	14.311s	2665%
6 Convolution	0.5046s	14.879s	2849%
9 Routing	0.0108s	0.1249s	1053%
10 Convolution	0.2669s	0.5167s	93%
11 Convolution	0.4157s	0.7581s	82%
16 Convolution	0.2113s	0.2816s	33%
20 Shortcut	0.0015s	0.0057s	271%

TABLE I
SINGLE-STAGE PERCEPTION CNN LAYERS WITH AND WITHOUT MEMORY CONTENTION.

Faster R-CNN Layer	Normal	Contention	Percentage Increase
3 Relu Activation	1.8892s	22.1668s	1173.34%
5 Convolution	0.5995s	3.4108s	468.9%
7 Convolution	1.1588s	28.272s	2339.77%
12 Convolution	0.9201s	6.551s	12.96%
14 Convolution	0.9158s	9.6611s	954.93%

TABLE II
TWO-STAGE PERCEPTION CNN LAYERS WITH AND WITHOUT MEMORY CONTENTION.

Characterization

- With the Extensive data gathered, multiple attributes and models were applied.
- We found that while it is possible to characterize the workload effectively, it required scenarios to fully capture the extent of behaviors analyzed.

$$c: \{1, \dots, n\} \text{ where } \beta_{Available} = \beta_{Max} - \sum_i^n c_i$$

such that processing time for task: a_i in the task queue c can be calculated as follows

$$\left\{ \begin{array}{l} c(a_i) = \text{Scenario 1} \quad \text{for } \beta_i^{threshold} > \beta_i^{Required} \\ \quad \quad \quad \&\& \beta_i^{Required} < \beta_i^{Available} \\ c(a_i) = \text{Scenario 2} \quad \text{for } \beta_i^{threshold} > \beta_i^{Required} \\ \quad \quad \quad \&\& \beta_i^{Required} > \beta_i^{Available} \\ c(a_i) = \text{Scenario 3} \quad \text{for } \beta_i^{threshold} < \beta_i^{Required} \end{array} \right.$$

Conclusion

- By characterizing and generalizing our findings, we provide valuable insights into the performance and potential of Edge devices for machine learning workloads.
- We Identified that convolutional layers, along with Routing, Shortcut, and ReLU activation layers, are the predominant layers affected by factors such as memory availability. Opening Future research Possibilities.
 - 2849% increase for convolutional layers, over 1053% increase for Routing layers, over 1173.34% increase for ReLU, and over 271% for Shortcut layers.
- Through Targeting the Attributes, we can effectively Utilize the Edge without Prior Tuning or Optimizing the models for each individual device.
- Each Scenario would give the scheduler important information for CAV tasks.



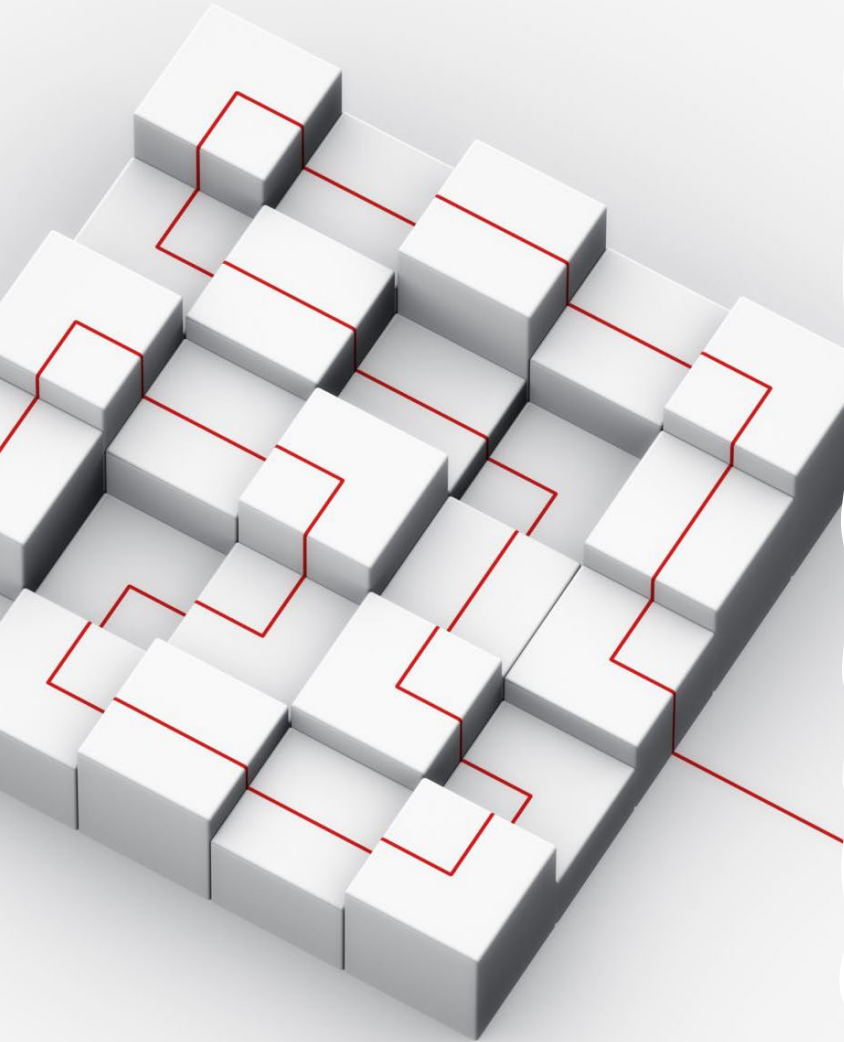
Discussion

- There are Several Unusual Findings
- In lower end devices, such as raspberry pi, we found a perplexing issue where the object detection model was able to successfully complete the detection task, but results differed wildly between two pis.
 - Both devices were same in hardware and was ran with the SAME microSD card. Containing both os and model.
 - One would correctly mark and classify objects where the other would mark almost all objects and non-objects as fire hydrants.



Continued.

- Over subsidized workloads will trigger the OOM killer Linux kernel on the Nvidia Jetson, but not on the other platforms with the same Linux Kernel and OOM policy.
- The Discrepancy between theoretical computation load and the actual workload latency eliminates theory based characterization and modeling.



Future works

- Currently, I am working on expanding the research using NAS searching and transformers.
- Characterizing the models for Edge use case can be useful for Cloud as well.
- Layer by Layer information and the memory intensive layers can prove significant to both architecture design and NAS searching research.
- The bottlenecks identified in the paper can also help with privacy and security research topics.