One Fish, Two Fish: Choosing Optimal Edge Topologies for Real-Time Autonomous Fish Surveys

Jonathan Tsen, Jason Anderson (advisor), Kate Keahey (advisor), Leonardo Bobadilla (advisor)
FATEC Shunji Nishimura, The University of Chicago, Argonne National Laboratory, Florida International University

Jupyter Notebook Code

Explanation of the project on Youtube

The oceans represent more than seventy percent of the Earth's surface, and marine ecosystems are central to many global challenges. However, monitoring the environment through state-of-the-art devices that require the use of GPUs presents unique challenges for the development of computer vision, artificial intelligence, and cloud computing technologies that allow marine scientists to collect and process data in greater volumes than ever before and offer solutions that can improve the efficiency of real-time data acquisition and analysis to address complex biological and ecological issues.

Background

- Monitoring the environment and ensuring the sustainable use of marine resources presents unique challenges which span socio-economic balances, technology development, and dynamic ecological processes that span spatial and temporal scales.
- We aim to develop indices of fish abundance and map distributions to important habitats in the bay in real time and leverage the capabilities of the AVs to improve survey efficiency to better understand the Biscayne Bay ecosystem.

Problem Statement

- What is the best strategy for collecting and analyzing data from the autonomous vehicles and can we leverage cloud computing resources to improve access to data products in real-time?
- How does the resolution of video data and quality of network connection influence which strategy is best?
- In this work we compare configurations in which the vehicle is equipped with an intelligent edge device versus configurations that perform similar computations in the cloud.

Metrics

Response time: how long from the time of video frame capture until the result can be received from the AI model. This has two components:

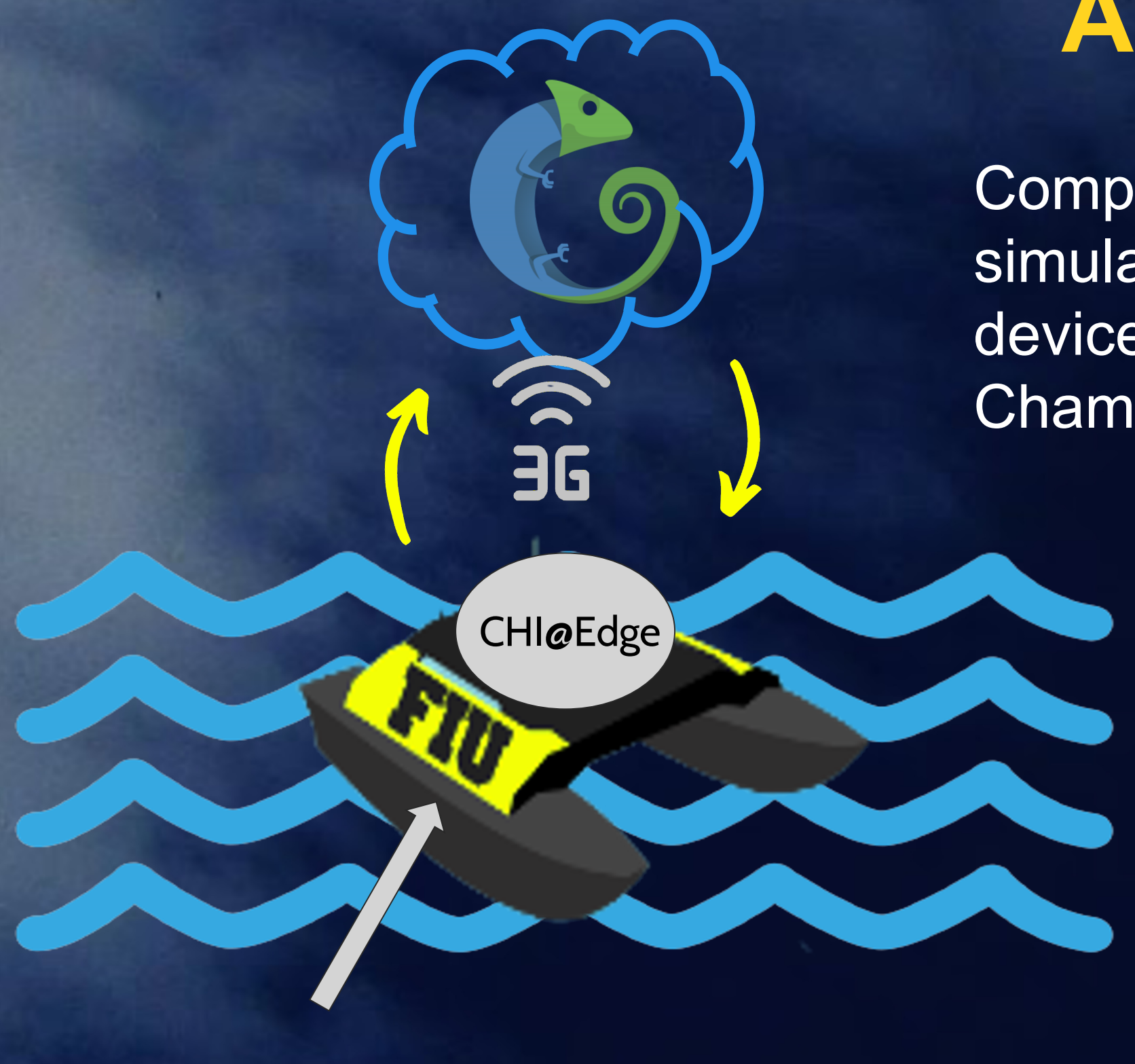
- **Runtime:** Time it takes for the model to run in a frame.
- **Transfer time:** Time it takes for video to be uploaded and returned from IoT to Chameleon Cloud.

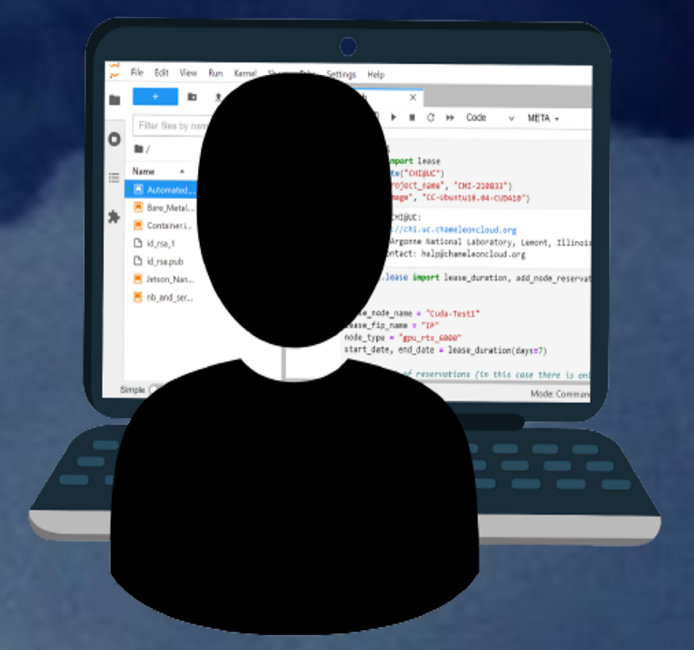
Size of the Model: Space required to store the model.

Acknowledgements

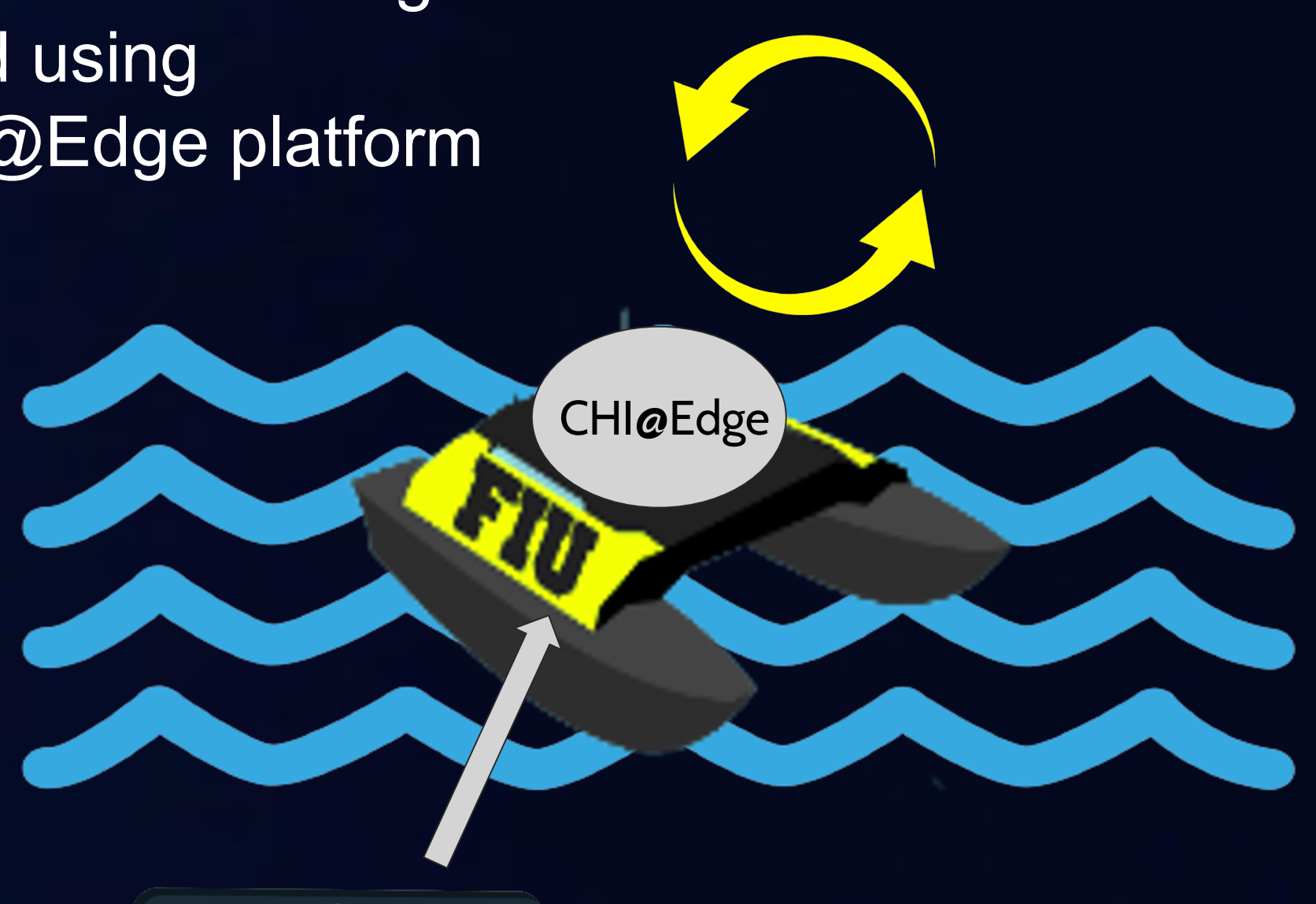
Approach

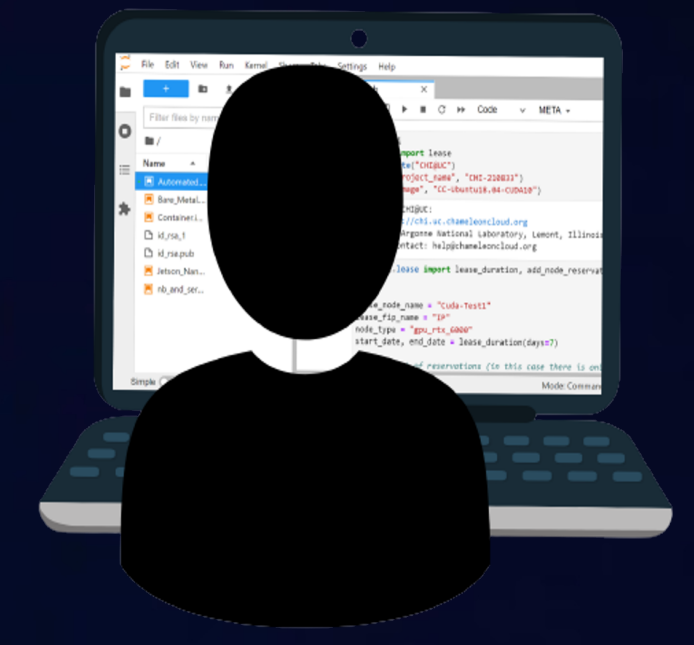
Compared performance using simulated workloads on real edge devices configured using Chameleon's CHI@Edge platform





- Raspberry Pi 4 on AV
- YOLOv4 model on Chameleon cloud GPU
- 3G connection between AV and cloud

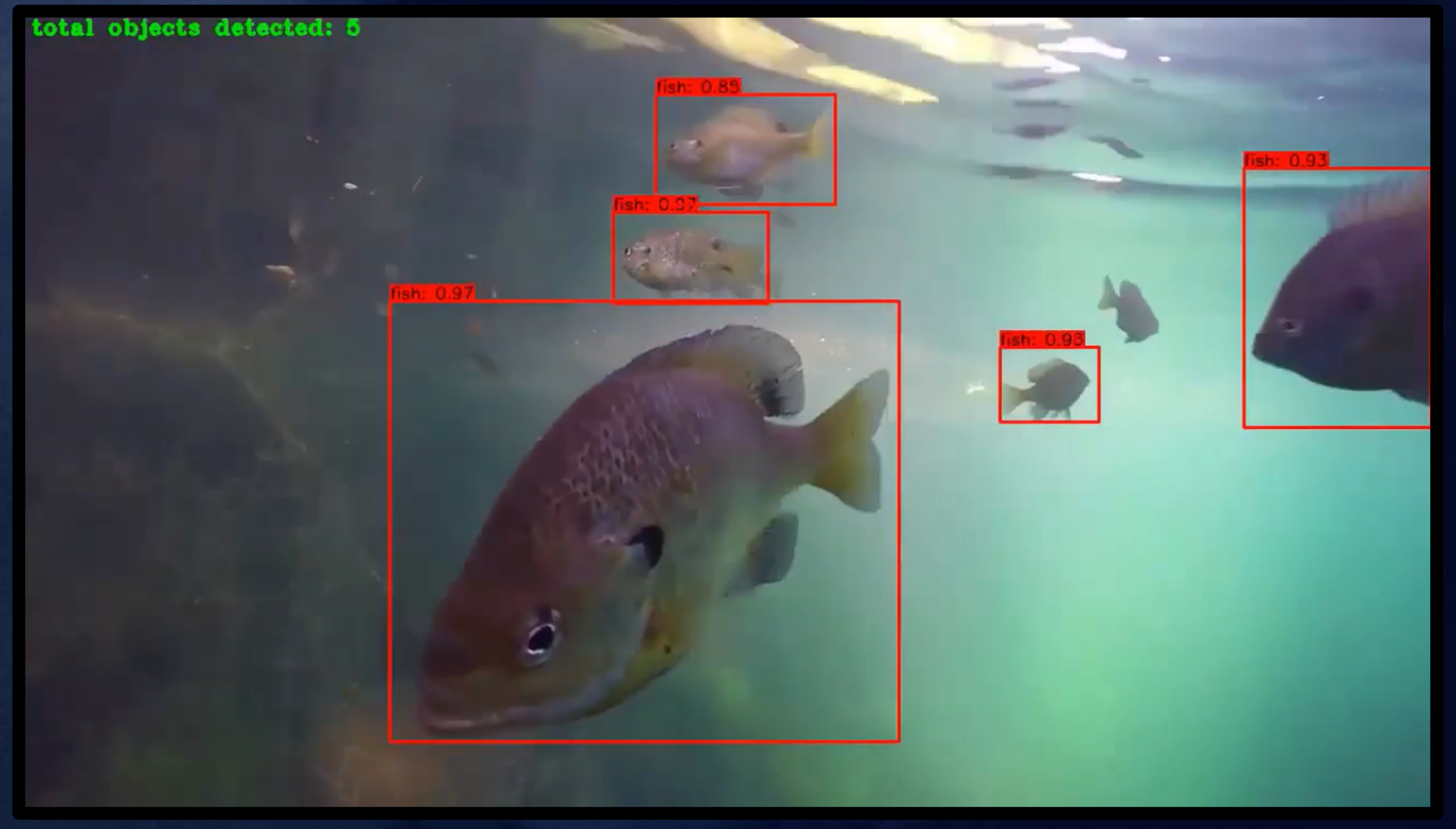




- NVIDIA Jetson Nano on AV
- Tiny-YOLOv4 model on the Jetson Nano
- No network connection required

Chameleon's CHI@Edge platform was used to provision an edge device of the same type that will be used on theAV, as well as provision cloud resources for our experiment.

We evaluate two versions of YOLO: **YOLOv4** (deployed on the cloud resources) and **Tiny-YOLOv4** (used on the edge device).

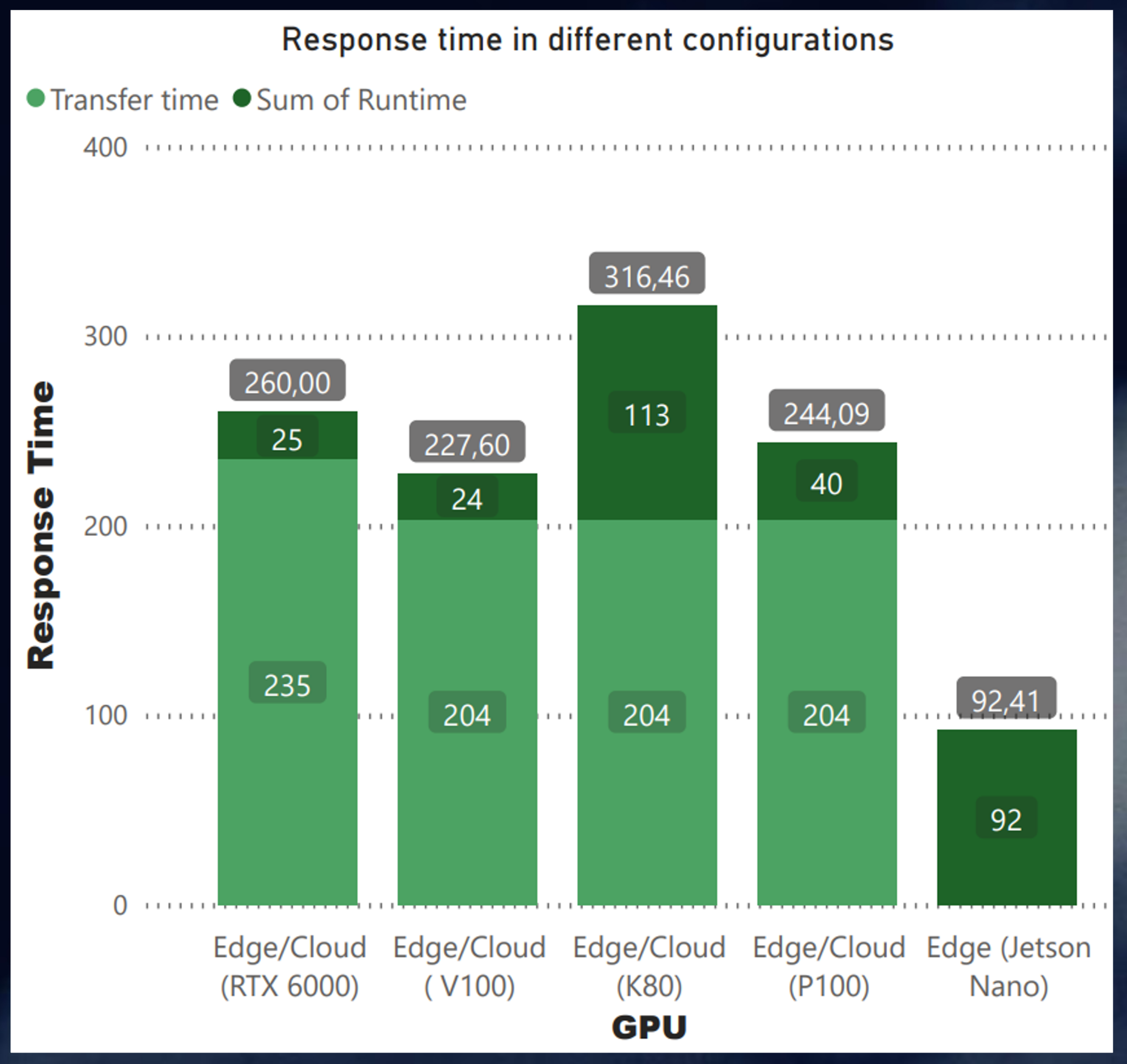


- Tiny-YOLOv4 has a smaller network size, less convolutional layers in the CSP backbone are compressed, the number of YOLO layers are two instead of three, and there are fewer anchor boxes for prediction. We did not observe a noticeable difference in reported accuracy of the model vs. YOLOv4 with our underwater training set.
- We tested results with a sample 1920x1080 image; this resolution is equivalent to what the underwater camera eventually will be capturing. Testing the image at a lower resolution of 1024x768, we saw a 1% reduction in Runtime and a decrease in standard deviation.
- We tested on real edge devices located at the University of Chicago.

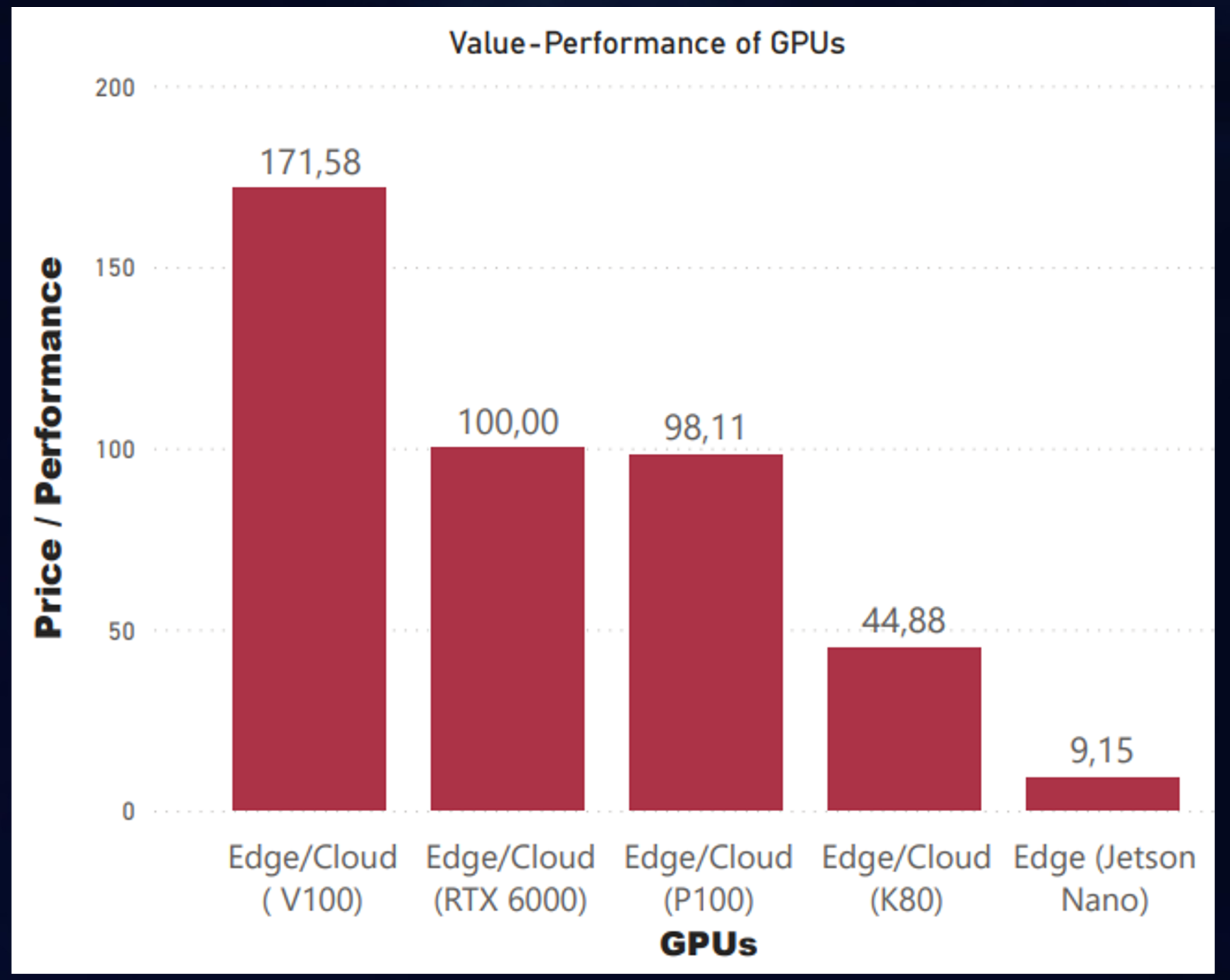
GPU	Location	Price
RTX 6000	CHI@UC	$4,000.00
Jetson Nano	CHI@Edge	$99.00
K80	CHI@TACC	$398.00
V100	CHI@TACC	$7,179.00
P100	CHI@TACC	$2,429.00

Besides measuring response time, we also measured the value/performance ratio by multiplying the GPU cost by the FPS (frames per second), which is the reciprocal of response tims. The prices of each GPU in dollars used to calculate the (price / performance) was collected in August 2021.

Results



The Jetson Nano does not have a transfer time since there is no need to transmit data. Though both the RTX 6000 and the edge device are located in Chicago, CHI@Edge adds latency due to network proxies to the device.



The Jetson Nano is almost 19 times cheaper per FPS than Edge/Cloud (V100). The cheaper RTX 6000 outperforms the more expensive V100 with this perspective.

Conclusion

- When you factor in the transfer time, the edge has a higher performance.
- The Jetson Nano also has a better price/performance ratio.
- The network condition has to significantly improve for the edge cloud to outperform the Jetson Nano.
- In terms of Response time and Value-Performance, we see that the RTX 6000 has a much better cost-benefit ratio than the V100.
- This experiment is reproducible through a Jupyter Notebook using the Python language in the Chameleon Cloud.