PERFORMANCE ANALYSIS OF EMBEDDED AI FOR PHYTOPLANKTON CLASSIFICATION[1]

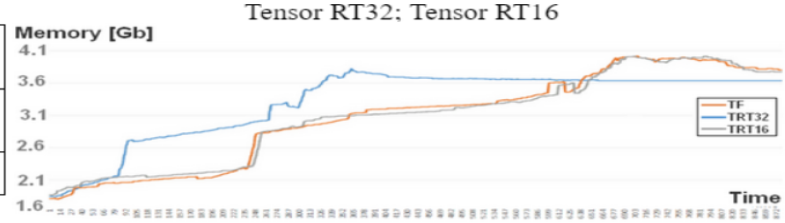
L. Caetano, B. Guterres, K. Sbrissa, A. Mendes, F. Vermeulen, L. Lain, M. Smith, J. Martinez, S. Botelho, P. Drews, N. Duarte, V. Menezes and M. Pias*
Federal University of Rio Grande, Av. Itália – KM 8, Rio Grande (Brazil)
E-mail: lucascaetano.mp@hotmail.com

Phytoplankton species play a critical role in marine ecosystems. Various environmental parameters may lead certain species to proliferate, resulting in harmful algal blooms (HAB). HAB can potentially cause adverse effects on aquatic life, human health and associated economic activities. The diversity in species makes communities highly heterogeneous in size, shape, and morphology. Early detection of phytoplankton species has significant importance to continuous HAB monitoring. However, it is a time-consuming and challenging activity even for experts. Embedded system solutions coupled with advanced Artificial Intelligence (AI) models are an early yet effective solution to support a HAB monitoring strategy. Based on the MobileNetV2 and NASNet architectures, these AI models take phytoplankton images as input and process them to provide helpful classification feedback to end-users.
Benchmark deep learning models [1-2] tailored for HAB monitoring within multitrophic aquaculture applications [3] have been embedded into an NVIDIA Jetson Nano platform. Overall system performance assessment aimed to check system resource usage and the impacts on FPS (image frames per second), which supports HAB early detection. The models were built using the TensorFlow library and Tensor-RT framework. Results indicate that the chosen hardware and software platform (Jetson Nano with Tensor-RT) can run the Deep Learning Phytoplankton classification at an acceptable image frame rate. Additionally, the results suggest additional bottlenecks to the already resource-constrained embedded boards. The use of CPU and GPU is acceptable, but memory usage becomes a crucial problem (Fig. 1). Provided all the benefits of the Tensor-RT framework, it still introduces an issue causing the system memory limit to be exceeded.

Table 1 - FPS results for the two target models.

Model	Tensor Flow	Tensor RT16	Tensor RT32
MobileNet V2	41	82	84
NASNet	34	73	80

Figure 1 - RAM use for MobileNetV2: Only with TensorFlow; Tensor RT32; Tensor RT16



[1] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
[2] Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8697-8710).
[3] Guterres, B., Khalid, S., Pias, M., & Botelho, S. (2022, April). A data integration pipeline towards reliable monitoring of phytoplankton and early detection of harmful algal blooms. In *NeurIPS 2021 Workshop Tackling Climate Change with Machine Learning* (Vol. 2021). NeurIPS.

[1] This work is part of the ASTRAL (All Atlantic Ocean Sustainable, Profitable and Resilient Aquaculture) project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 863034.