HALO: High-efficiency Automated Low-SWaP Operations

Team Members:

- Sloan Hatter (shatter2022@my.fit.edu)
- Blake Gisclair (bgisclair2022@my.fit.edu)

Faculty Advisor: Dr. Ryan T. White (rwhite@fit.edu)

Client: Dr. Ryan T. White (rwhite@fit.edu), NEural TransmissionS (NETs) Lab

Date(s) to Meet with Client:

- January 14th, 2026

Goal and Motivation:

Currently, orbital object detection, done through the Vision Transformer (ViT) neural network architecture, is carried out on large computers; however, this is not sustainable as larger computers take up too much space and resources within satellite systems. It is ideal for orbital object detection functionalities to be carried out on smaller computers, such as a Raspberry Pi, as they take up much less space and resources. The current method to make neural networks compatible with running on smaller computers is to take out layers of the model; however, this reduces accuracy. The technique I aim to use is to condense the neural network's weights to a smaller representation. Currently, weights are stored as floats in C++, which occupy 32 bits. I aim to reduce the weights to a 1-bit representation.

There has been some research and development in 1-bit representations for the ViT architecture; however, the smallest representation is currently only 1.58 bits, and its use is not directed for large-scale object detection and segmentation, as would be used in on-orbit satellite characterization. Therefore, I aim to expand upon these successes and develop a 1-bit quantization for a ViT deployable on on-orbit satellites to be used in autonomous systems.

Approach:

Accurate Object Detection

The user will be able to detect and identify orbital objects, more specifically, satellite objects, within a given image space.

Low-SWaP Hardware Compatibility and Deployability

The user will be able to send to space smaller object detection-capable computers that take up less space and resources.

Autonomous Operations Abilities

The user will gain the ability to run these computer vision models on satellite hardware and execute autonomous operations, such as docking and repairs. Automating these operations will reduce the need to send up manned missions and increase the longevity of active missions.

Novel features/functionalities:

One-bit representations of weights have been restricted mainly to LLM transformers and have not been explored extensively for vision transformers and vision tasks, such as classification and segmentation.

Algorithms and Tools:

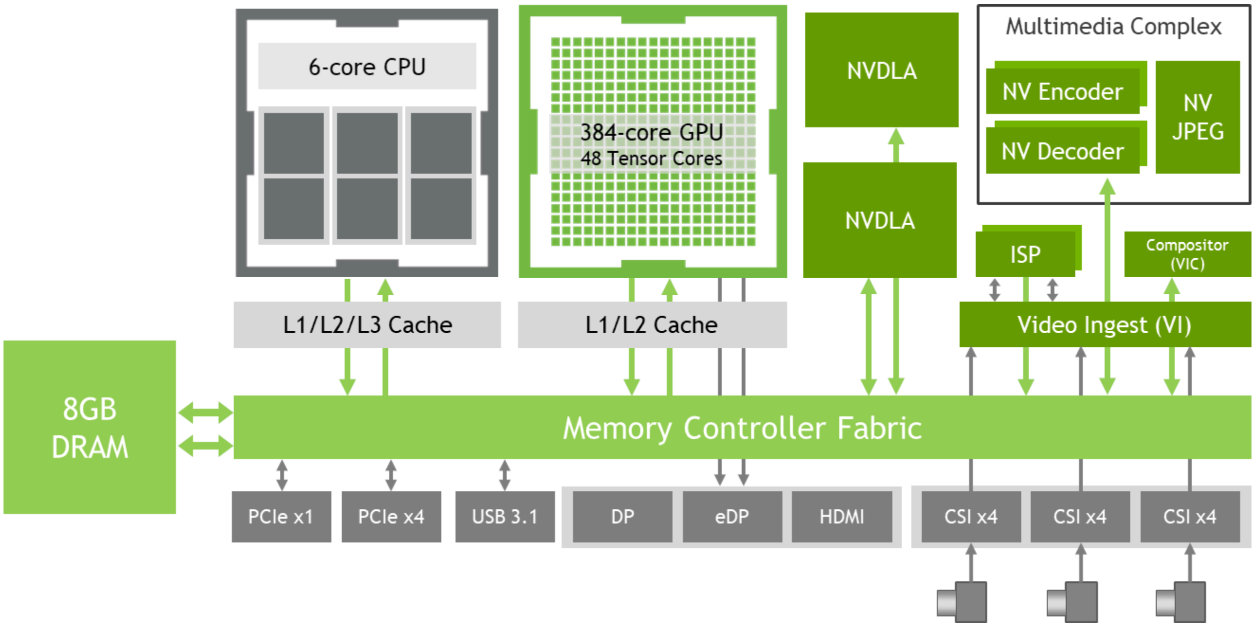
- Python
- Jupyter Lab
- Multiple datasets for training and testing: hardware in the loop (HIL), web satellite data, and digital twin on-demand data.
- Neural Networks and Vision Transformers (ViTs)
- Hailo-8 NPU/ HailoRT (for model inference)
- Jetson AGX Orin Developer Kit (64GB)

Technical Challenges:

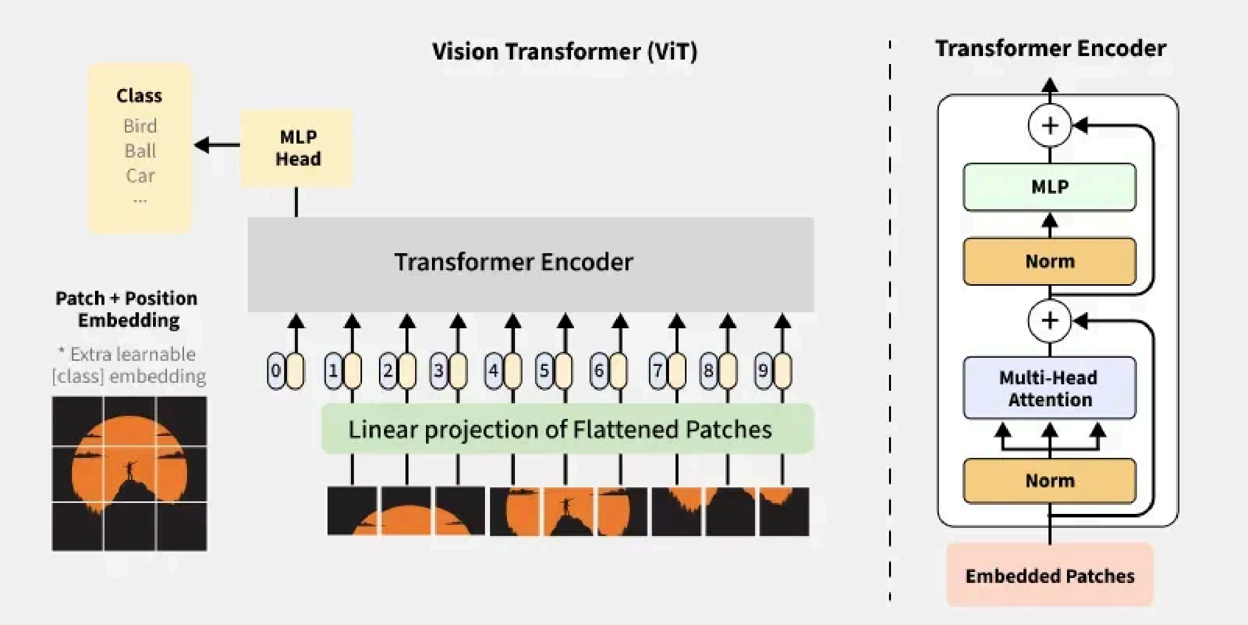
- Maintaining accuracy as lower bit models are implemented
- Balancing power draw while still adhering to Low-SWaP parameters
- Ensuring the Jetson AGX Orin has the proper infrastructure to effectively run lower bit models

Design:

- System Architecture Design of Jetson AGX Orin



- Neural Network Architecture: Transformer Diagram



Evaluation:

- **Overall Confidence**: Across 898 detections on 100 images, the model assigns high confidence on average (mean = 0.91), indicating strong certainty in its predicted bounding boxes.
- **Per-Image Stability**: Average confidence per image is consistently high (mean/median = 0.94, IQR = 0.91–0.97), showing low variability across images rather than confidence being driven by a small subset.
- **Image-Level Coverage**: 79% of images have mean confidence above 0.90, and 43% exceed 0.95, suggesting that high confidence is the norm across the evaluation set.
- **Worst-Case Behavior**: Even the least confident images maintain per-image averages ≥ 0.81, with all detections above 0.70, indicating no low-confidence failure cases.
- **Within-Image Consistency**: In 50% of images, all detections score above 0.80, suggesting coherent and stable predictions within individual scenes.
    - The 16-bit model shows consistently high and stable confidence across detections and images, with minimal variance and no low-confidence failure cases, indicating strong internal certainty.

Progress Summary:

Task	Completion %	Sloan	To Do
Hardware Swap	100%	Switch out the Raspberry Pi 5 AI HAT+ for the Jetson AGX Orin	None
16-bit Representation	100%	Implement 16-bit model	None

Milestone 4 (Feb 23):

- Implement and get metrics for 8-bit model
- Implement and get metrics for 4-bit model
- Record demo video for each model

Milestone 5 (March 30):

- Manually implement and get metrics for 2-bit model
- Work towards shrinking the model down to 1-bit through quantization methods such as Quantization Aware Training (QAT)
- Record metrics throughout implementation and testing

Milestone 6 (April 20):

- Complete 1-bit model
- Record final metrics for 1-bit model
- Create demo video
- Create showcase poster

Task Vector for Milestone 4:

Task	Sloan
Implement 8-bit model	100%
Record metrics for 8-bit model	100%
Demo 8-bit model	100%
Implement 4-bit model	100%
Record metrics for 4-bit model	100%
Demo 4-bit model	100%

Description of each planned task for Milestone 4:

- **Implement 8-bit model:** Start with the current 16-bit model and utilize NVIDIA tools from their JetPack SDK software to convert the model into INT8 precision; this will be done by implementing either Post-Training Quantization (PTQ) or Quantization-Aware Training (QAT). The quantized model is then optimized by TensorRT into an efficient file that is specifically designed for the Jetson AGX Orin hardware. The optimized TensorRT file will then be integrated into our application using the TensorRT runtime.
- **Record metrics for 8-bit model & Demo 8-bit model:** I will record metrics for the 8-bit model, including but not limited to run time, accuracy, and power draw. There will be a video accompanying the metrics showing that the 8-bit model is correctly identifying objects.
- **Implement 4-bit model:** Start with the current 16-bit model and utilize NVIDIA tools from their JetPack SDK software to convert the model into INT4 precision; this will be done by implementing either Post-Training Quantization (PTQ) or Quantization-Aware Training (QAT). The quantized model is then optimized by TensorRT into an efficient file that is specifically designed for the Jetson AGX Orin hardware. The optimized TensorRT file will then be integrated into our application using the TensorRT runtime.
- **Record metrics for 4-bit model & Demo 4-bit model:** I will record metrics for the 4-bit model, including but not limited to run time, accuracy, and power draw. There will be a video accompanying the metrics showing that the 4-bit model is correctly identifying objects.