A Preliminary Implementation of the F' Framework for Managing Machine Learning Applications

Bronco Space

California State Polytechnic University, Pomona

For Public Release

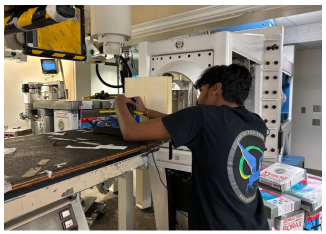
April 23, 2024

About Bronco Space

- Bronco Space is a mostly undergraduate student organization at Cal Poly Pomona.
- Founded in 2019, Bronco Space has run the entire gamut from starting at zero to becoming the leading space technology group at Cal Poly Pomona.
- Bronco Space has engaged in multiple NASA funded instrument development projects. Average time for TRL 3 to TRL 6 is 10 months.

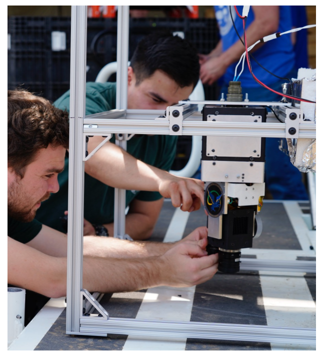
Providing a Path to Flight

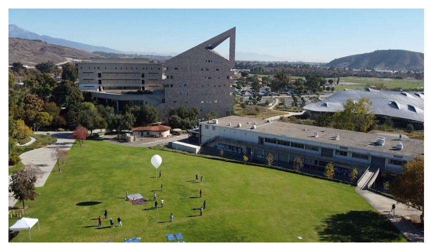
Laboratory Research and Development



- Technologies are created by students in the laboratories and tested in simulation environments.
- These technologies are then made into fully developed prototypes.

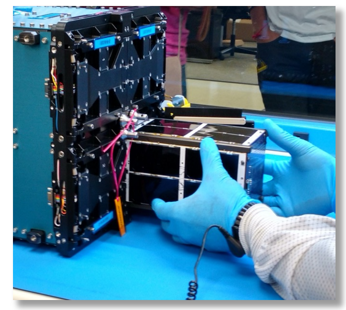
Sub-Orbital Testing

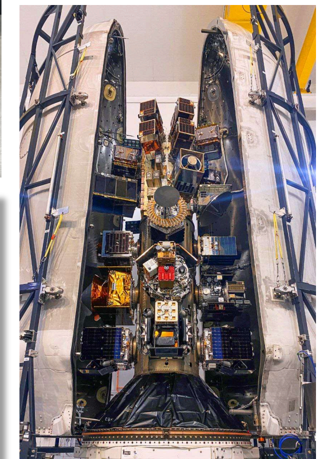




- Bronco Space has extensive experience conducting sub-orbital test campaigns.
- High altitude balloons and rocket power landers for TRL6 tech validation.

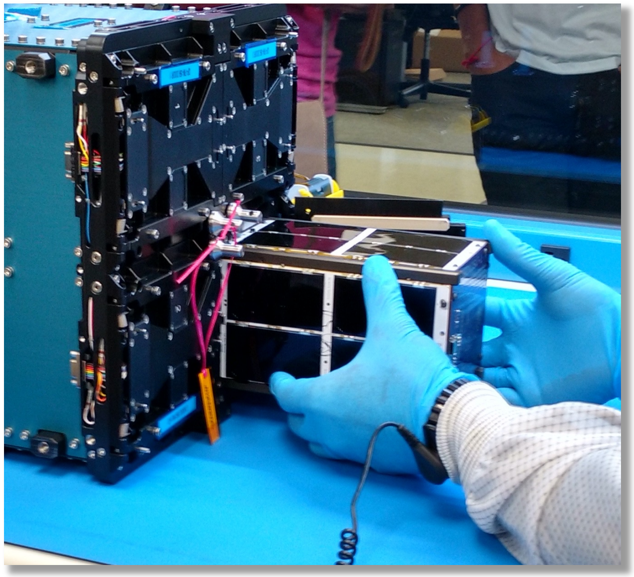
Orbital Launches

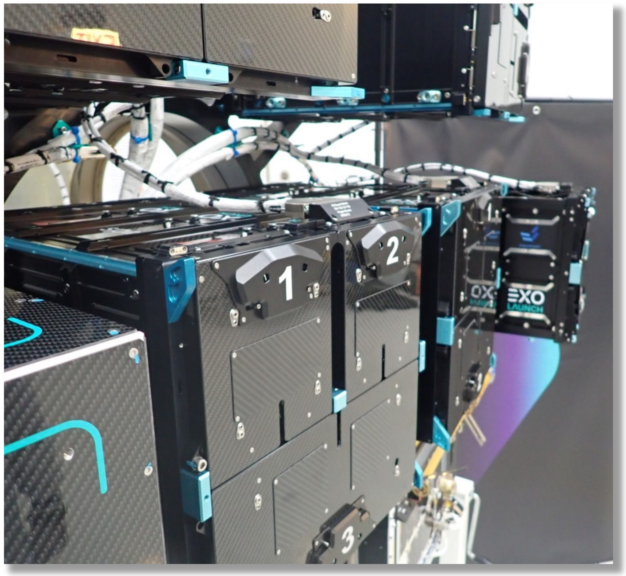




- Four CubeSat launches conducted in the last two years.
- Qualification testing, launch services, and mission operations through commercial and university partners.

Bronco Space's Satellites

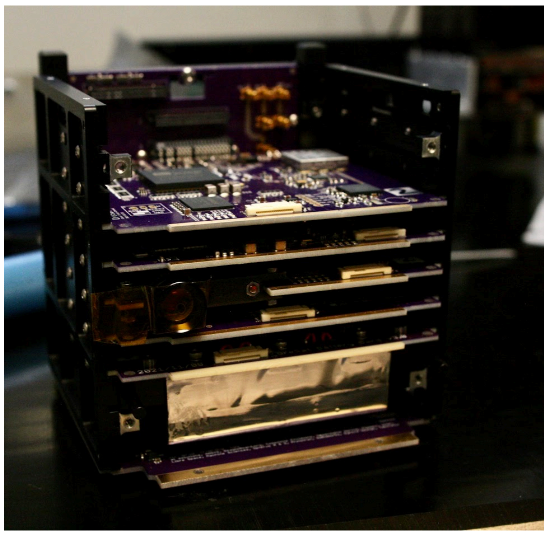




- In the last calendar year our organization has delivered three unique CubeSats for launch to LEO, all on commercial launch services. Our first CubeSat was launched in Summer 2022.
- The satellites have trended to be significantly cheaper and faster with each iteration.
- Our current efforts are focused on applying machine learning algorithms to space technologies.

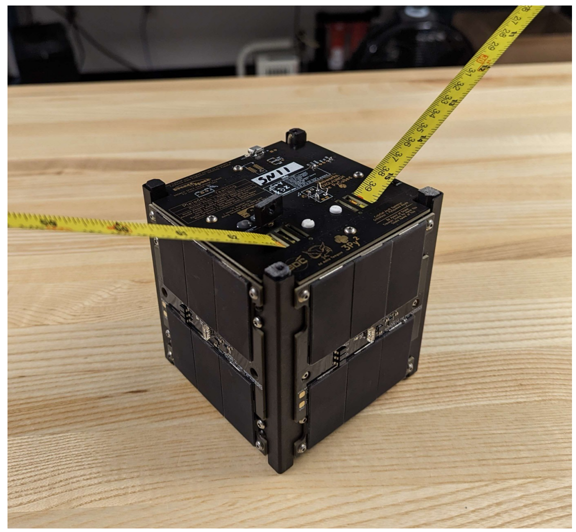
Software on the Three US Open-Source CubeSat Platforms

OreSat



- Modular Card Cage System.
- Custom flight software for each mission running on ChibiOS.
- Resilient but high cost.

PyCubed / PROVES

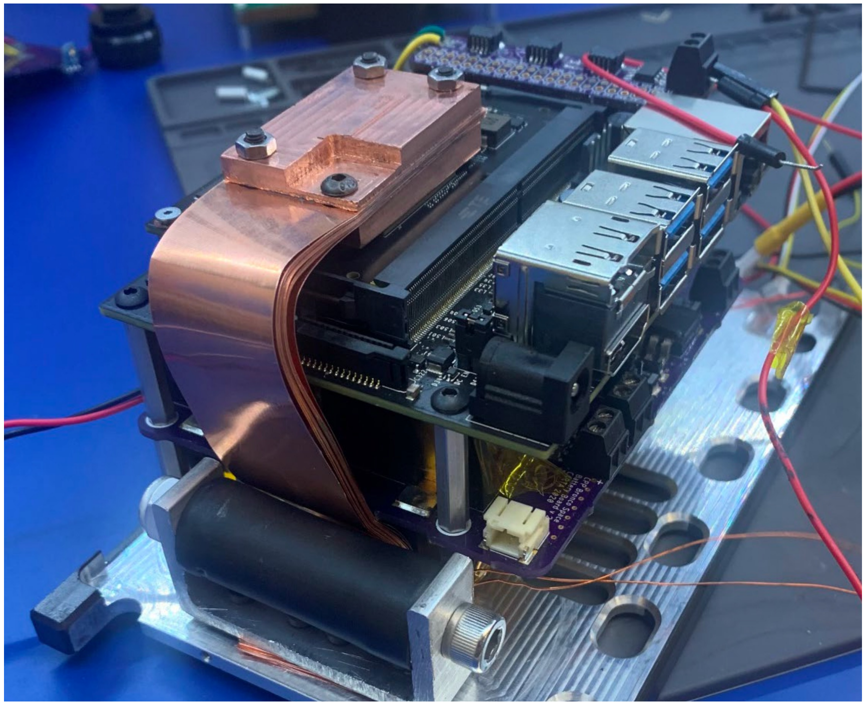


- Single Board Computer architecture.
- Simple state machine in Circuit Python.
- Less resilient but very low cost.

Artemis CubeSat Kit

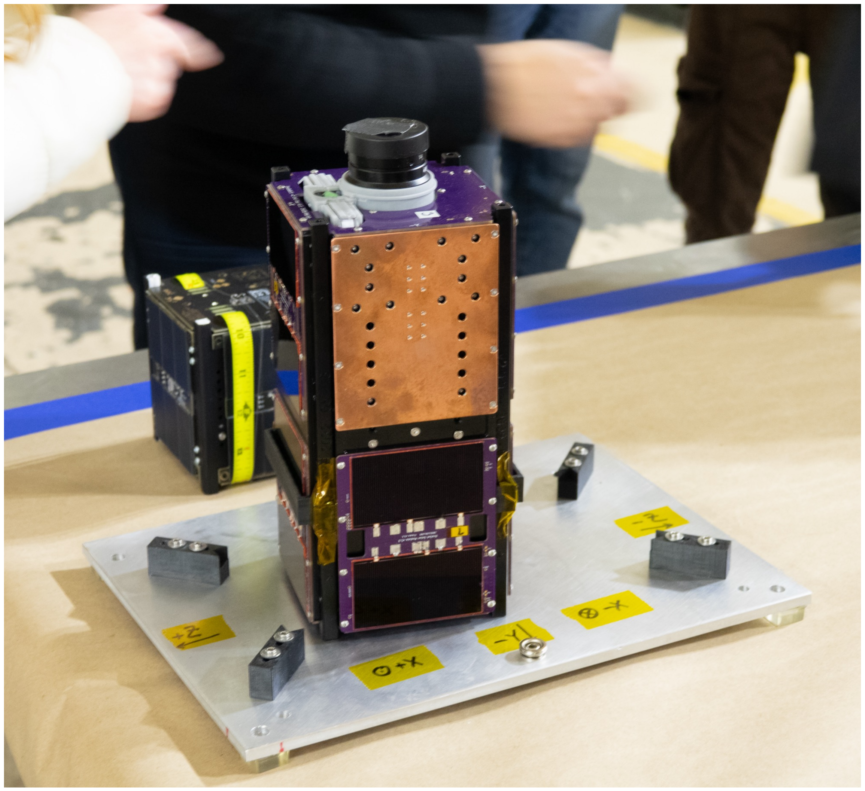
- Most traditional CubeSat architecture.
- Formerly COSMOS but now F’.
- Heavily supported by NASA.

What is SCALES?



- Spacecraft Compartmentalized Autonomous Learning and Edge Computing System (SCALES)
- Robust, adaptable, and rapidly deployable flight computer framework for implementing AI/ML decision-making systems on spacecraft.
- An integrated hardware / software system that creates a reusable environment for the execution and training of Machine Learning (ML) algorithms on-orbit.
- Path towards creating a safe, robust, and adaptable software environment to implement various ML/AI algorithms in orbit, enabling greater autonomy for existing space systems.

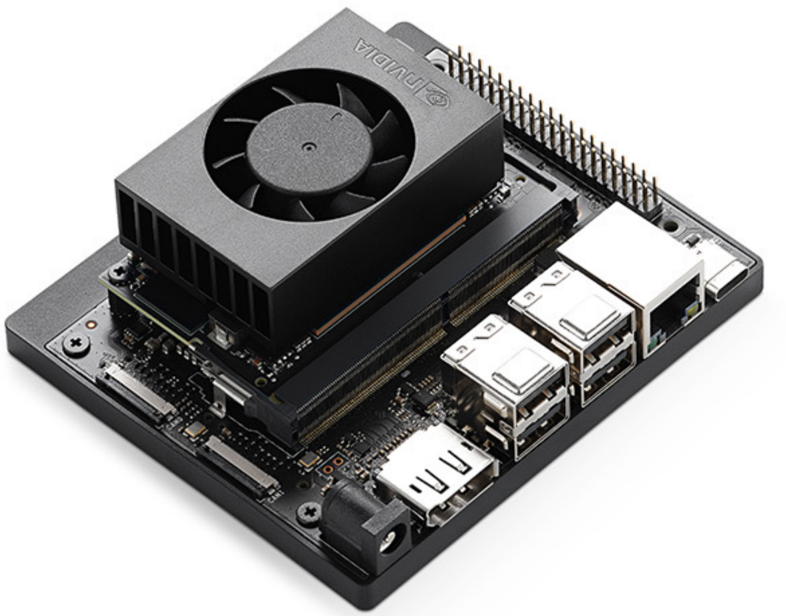
Why Are We Making SCALES?



- More than 2,300 small satellites have launched worldwide.
- Compact, cost efficient, and experimental.

- Physical and budgetary constraints necessitate compromises.
  - Diminished High-bandwidth downlinks
  - Smaller operations teams
- Heightened need for autonomous functionality to address physical and fiscal constraints

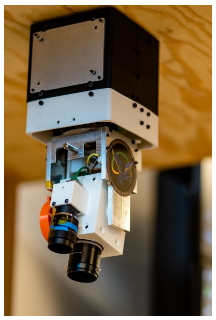
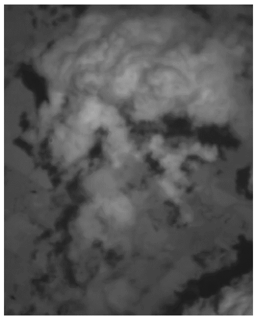
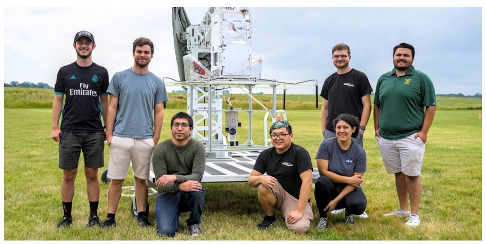
SCALES Mission



An integration between a Commercial Off the Shelf (COTS) edge computing hardware, that enhances on orbit computing, and a reusable software architecture, that provides reliability and mission assurance.

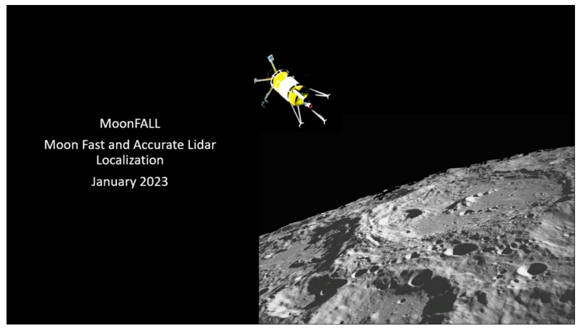
Pedigree for Using ML Models on Spacecraft

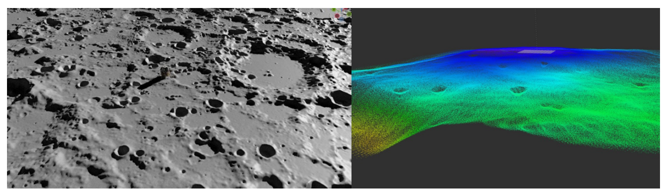
Bronco Ember

- Autonomous Wildfire Detection System
- Computer Vision and Edge Computing for Autonomous Remote Sensing

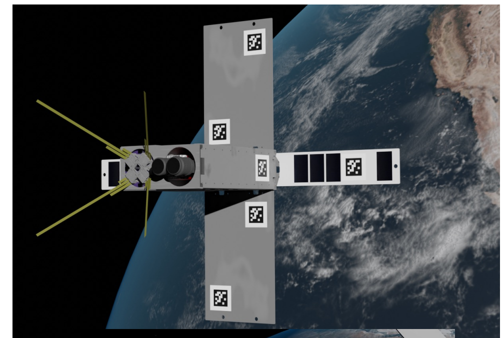
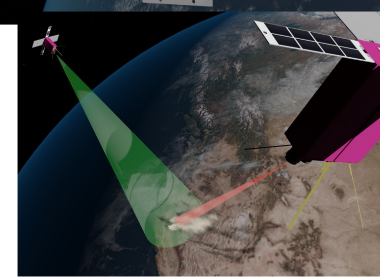
MoonFALL





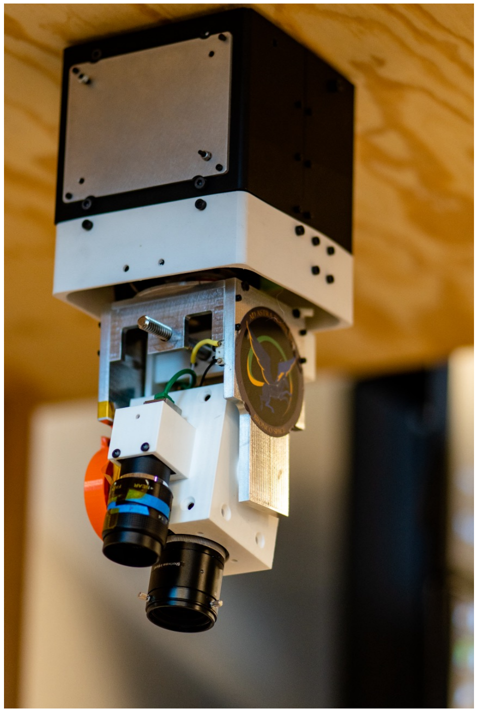
- Autonomous mapping, precision landing, and hazard avoidance system.
- ML model trained in a digital twin and tested sub orbitally

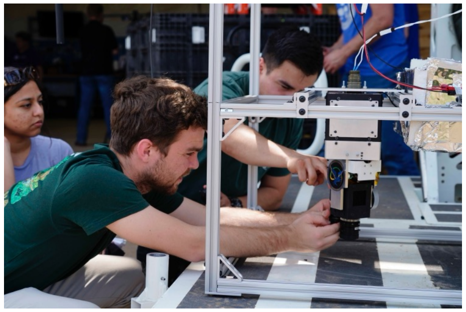
Autonomous Satellites

- Autonomous operations for Spacecraft conducting Rendezvous & Proximity Operations and Autonomous Observation.

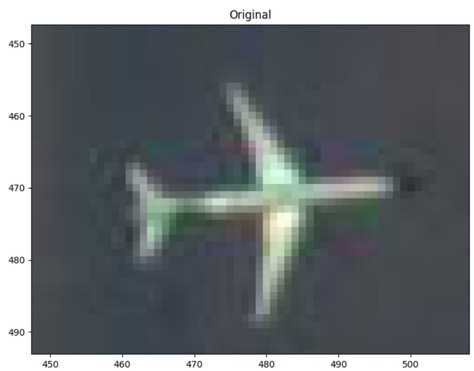
Use Case Study: Bronco Ember



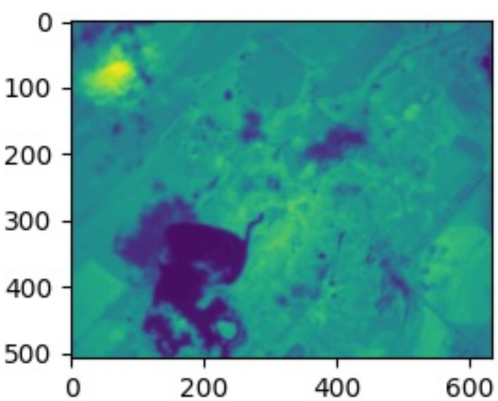


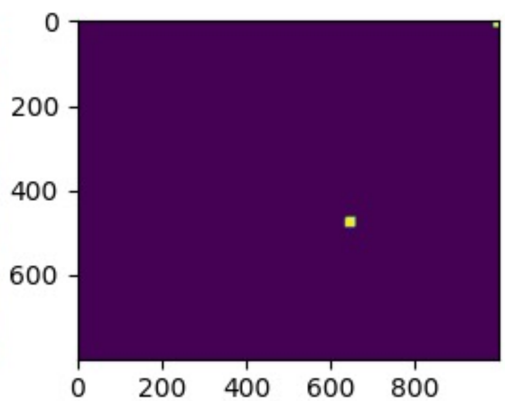
- Bronco Ember Autonomous Wildfire Detection Instrument
- Requires Human tagging of data for models
- Real-time fire prediction model and on-board analysis of data for autonomous search and track of nascent wildfires.

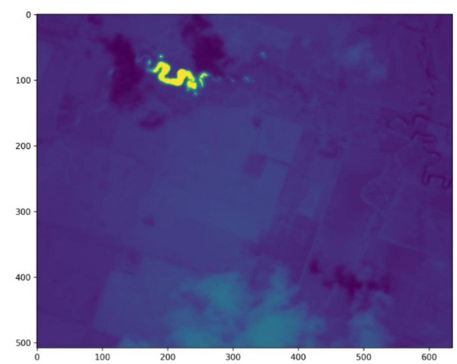
Use Case Study: Bronco Ember

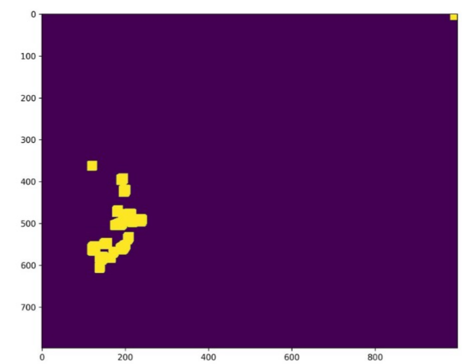


- Running this sort of detection software on remote platforms can be valuable for autonomous notification and re-tasking.
  - Downlink of images would require minutes to complete. A notification message is milliseconds.
- Current implementation is undynamic and requires close monitoring while using the model outputs as feedback into the control system.

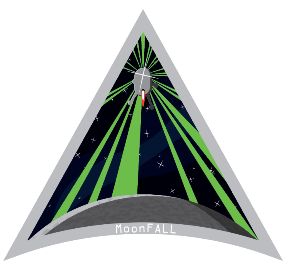






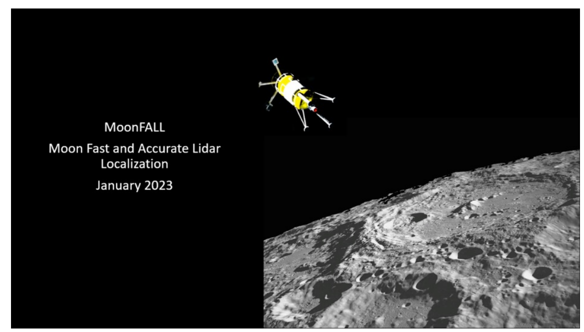


Use Case Study: MoonFALL

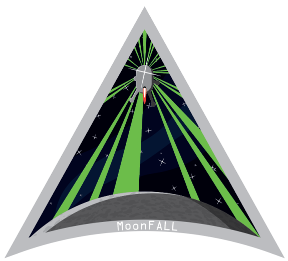


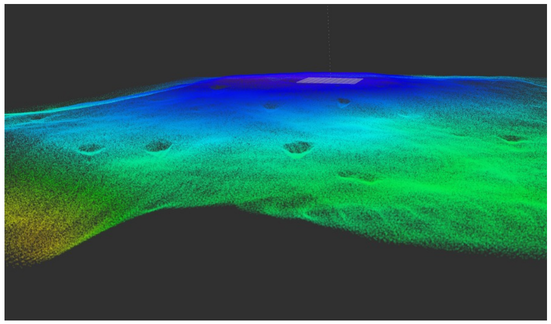
Picture Credit: Masten Space

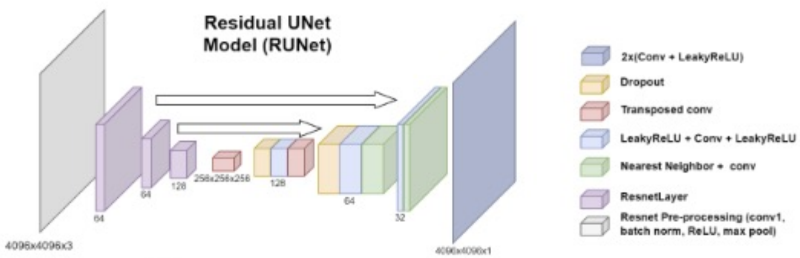
- MoonFALL: Moon Fast and Accurate LiDAR Localization
- Machine Learning model used to generate a lunar terrain map during a nighttime precision challenge.
- Rapid inference is required to ingest the LiDAR and visible camera data streams within the time constraints.

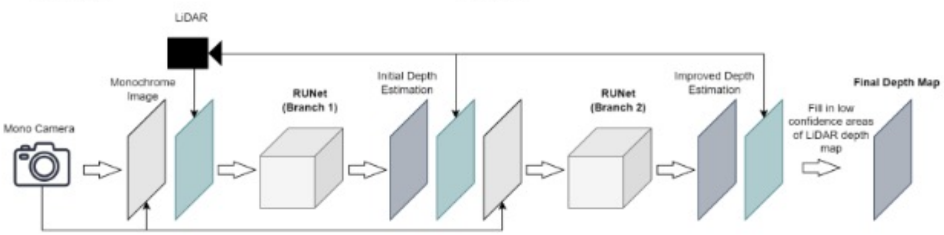


Use Case Study: MoonFALL

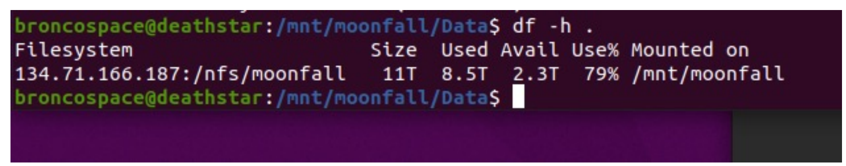








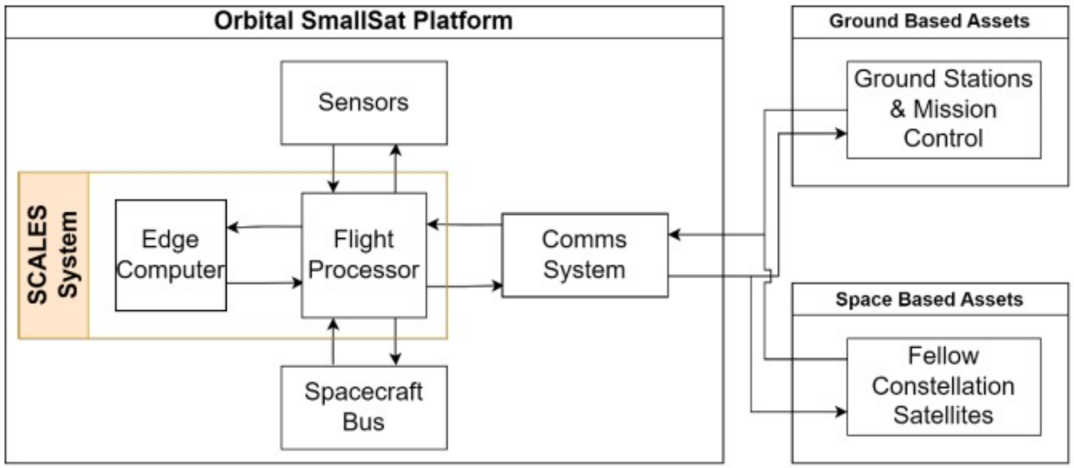
- MoonFALL Training models depend on millions of simulated terrain data.
    - 20MP Cameras, Lidar point clouds
    - 10 seconds landing simulation generates 5 GB of training data.
- 200,000 photos + lidar samples 8.5 Terabytes of data.
    - Requires weeks of training per batch.
- Training is better done in situ than saved, transferred, and processed.



Machine Learning in a Space Environment with F’

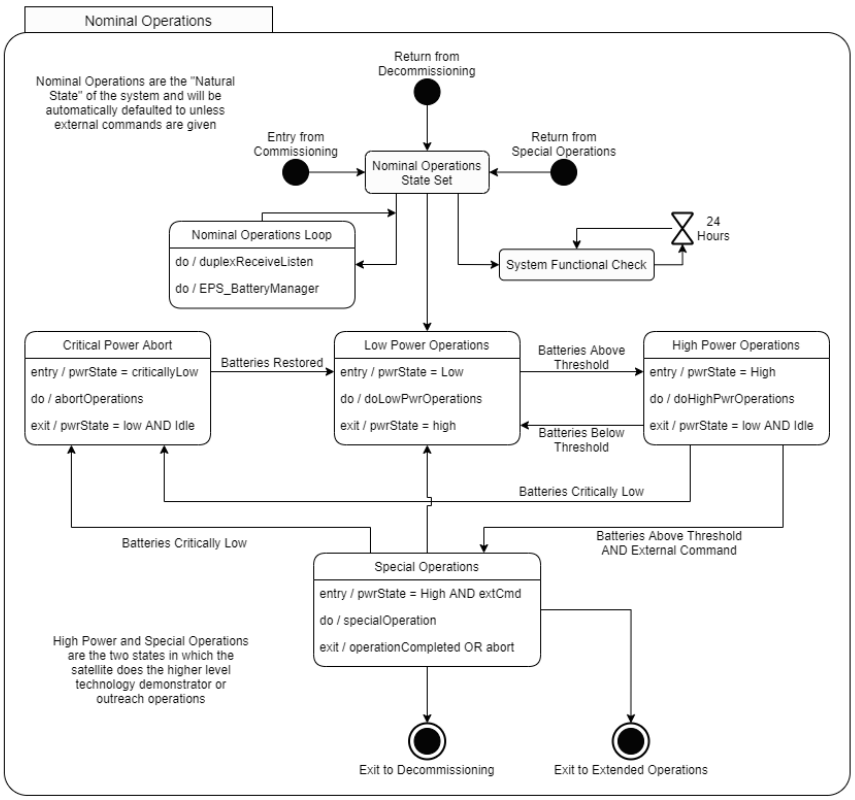
- Survival of the Space Environment
- Emphasis on Safety and reliability
- Allows for easy integration of new functionalities without compromising the performance of existing features.

- Quick reuse of core functionalities, saving engineering time to generate Source Lines of Code (SLoC) and perform testing and verification.



- Open-source satellite development and investigations into low cost rapidly deployed systems.

Example of Simplified CubeSat Software

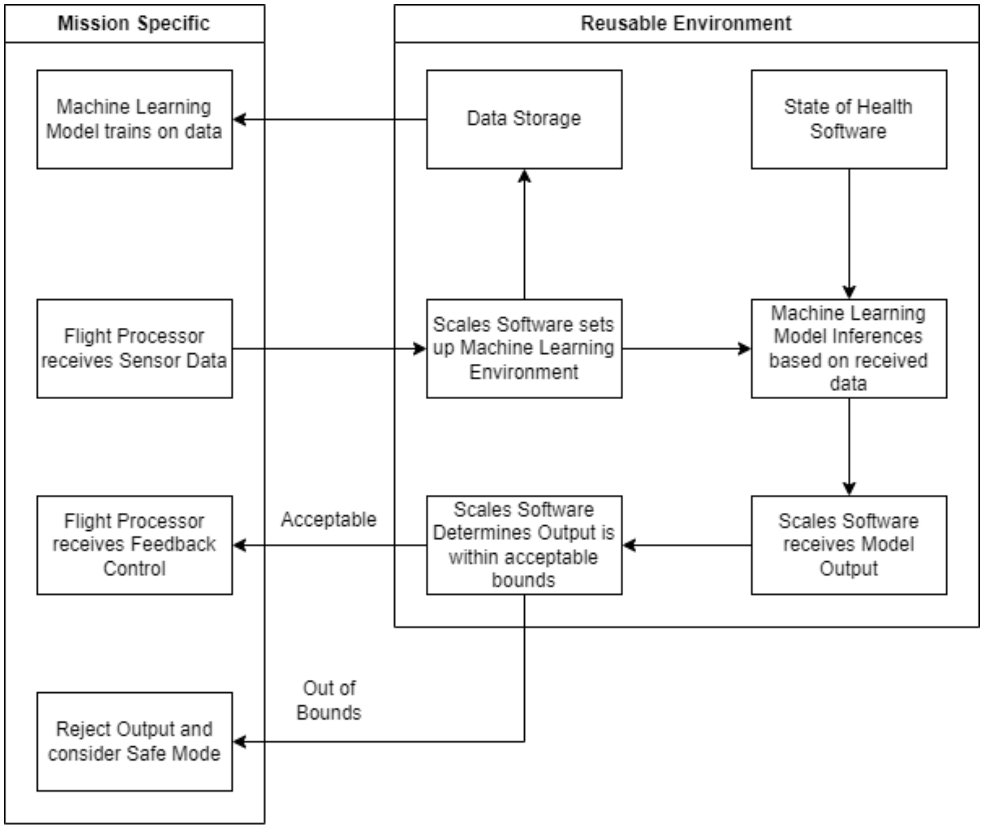


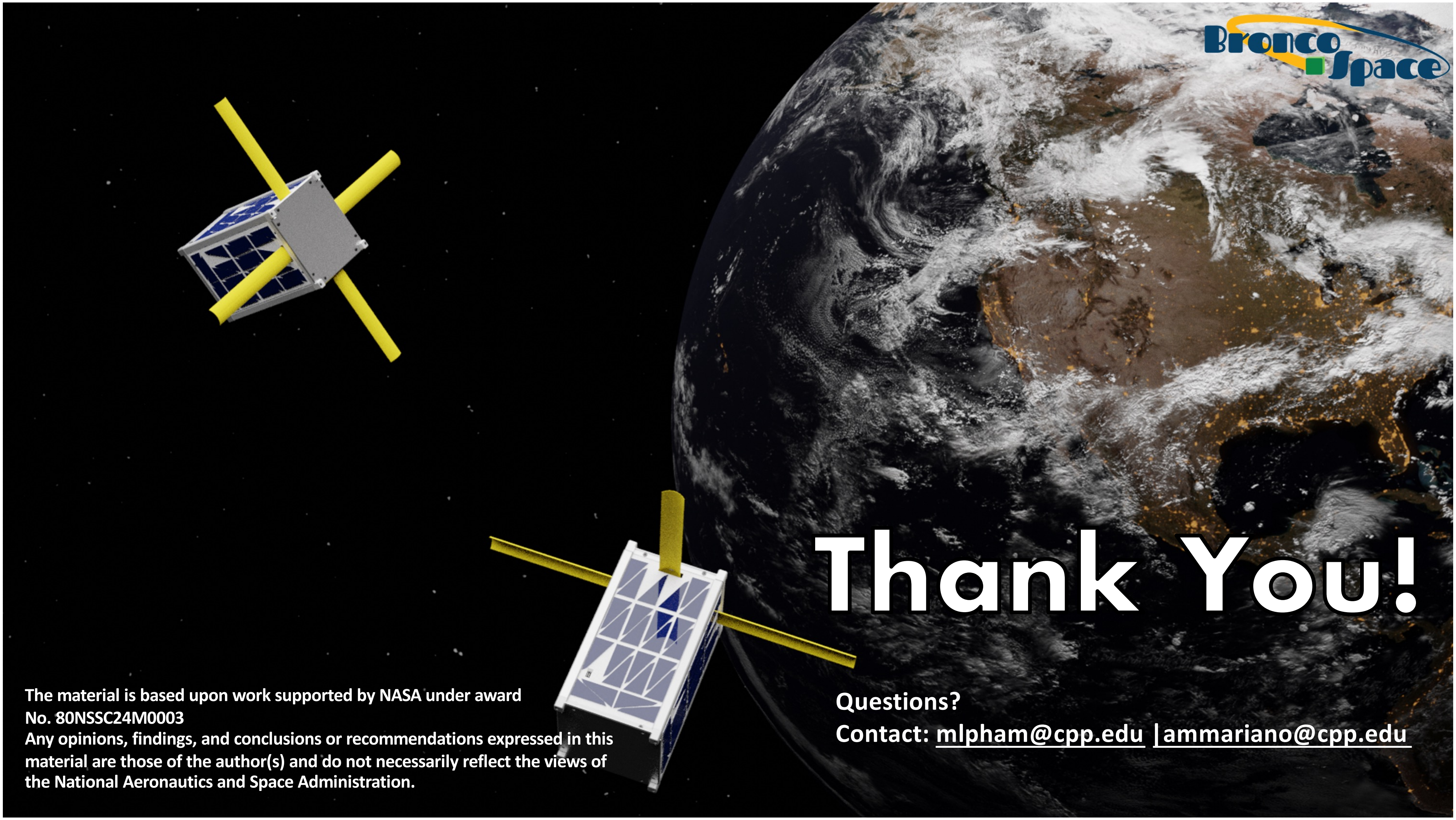
- Usually, our software manifests as a set of state machines.
- Software becomes mission-specific and unscalable.

- Need for a reusable and scalable software framework to save time on redeployment of core functionalities between spacecraft.
  - Downlink, telemetry, health monitoring, commands, etc.
  - Mission-specific components should be able to comfortably connect to existing trusted, robust, and generalized software.

Enhancements / Impacts

- Inference Time (YOLO Algorithm)
  - SoA: 4 Images/Sec
    - Using the Mars Ingenuity Helicopter Processor
  - Target: ~1,400 Images/Sec
    - Using the proposed SCALES dedicated co-processor
- Ability to directly train new ML models on board, reducing the need for ground based high performance computers and costly downlink time.
- Significantly enhance satellite productivity by enabling autonomous decision-making.
  - Cislunar and Deep space
- Enhance adaptability to dynamic environments.




Bronco Space
Thank You!
Questions?
Contact: mlpham@cpp.edu | ammariano@cpp.edu
The material is based upon work supported by NASA under award No. 80NSSC24M0003
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.