Introduction: Fully autonomous aerial systems (FAAS) execute dynamic missions based on data sensed at runtime, achieving high level goals with no human involvement. FAAS include unmanned aerial vehicles (UAV) for data sensing, and edge and cloud systems for data processing and decision making. FAAS are a considerable improvement upon unmanned aerial systems (UAS). By figure 1, UAS require human piloting and decision making, whereas FAAS execute autonomously to achieve user goals. Autonomous execution allows FAAS to operate continuously, intelligently, and in environments that may be unsafe for humans.

FAAS application domains are many, including crop scouting and disease detection to improve crop yield, search and rescue in environments too dangerous for humans to reach, and battlefield surveillance. Each domain and application require different configurations (e.g # of UAV, compute resources, search algorithms, classifiers, autonomy policies). Furthermore, FAAS operate in resource constrained environments like crop fields and disaster areas, creating power concerns for both the UAV and edge. FAAS end users are, therefore, concerned with maximizing **mission throughput**— *i.e., the number of missions the system can complete without recharging.*

FAAS end users would like to know which configurations will achieve high mission throughput. Semi and fully autonomous software packages exist for UAV, but leave architectural configuration up to the user[1]. Users must therefore test their FAAS with many configurations to maximize throughput. To the best of my knowledge, no existing FAAS platform has the ability to benchmark throughput for different hardware and software configurations. I intend to (1) create an open source middleware for FAAS, (2) provide precise FAAS benchmarks for common application domains, (3) introduce new models to improve FAAS execution and throughput testing, and (4) analyze and manage emerging hardware platforms and AI techniques for long-running FAAS.

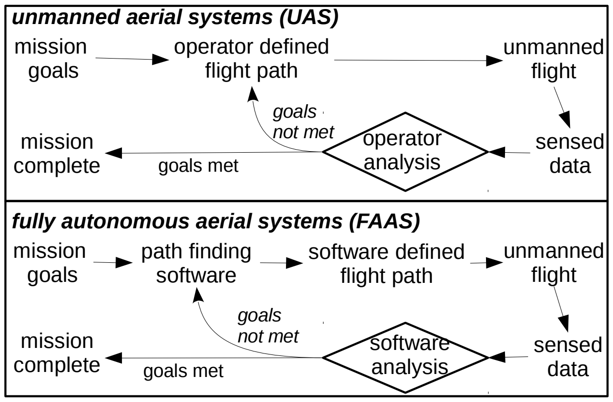

Fig. 1: Comparison between UAS and FAAS

Intellectual Merit: This research will make a number of contributions in benchmarking, modeling, and management.

I will create benchmarks (i.e., software that represents typical hardware usage patterns for a class of workloads) for three common FAAS use cases, (1) high quality target recognition, (2) aerial photography, and (3) aerial detection and recognition. Benchmark (1) will use UAV, facial recognition, and path finding algorithms to find the best images of human faces in the FAAS state space. Benchmark (2) will surveil a crop field, using computer vision and image analysis techniques to predict crop yield over the course of the growing season. Benchmark (3) will combine techniques from (1) and (2), to surveil a crop field, and detect and recognize a number of common crop diseases. Each of these benchmarks can be used to profile mission throughput for FAAS that perform tasks within their broad use case, and will perform highly as standalone applications.

To improve FAAS execution efficiency I will introduce a novel model, the autonomy cube: an offline data structures for storing previous FAAS execution data with spatial links to optimize FAAS movement. Autonomy cubes will allow the FAAS to use move-

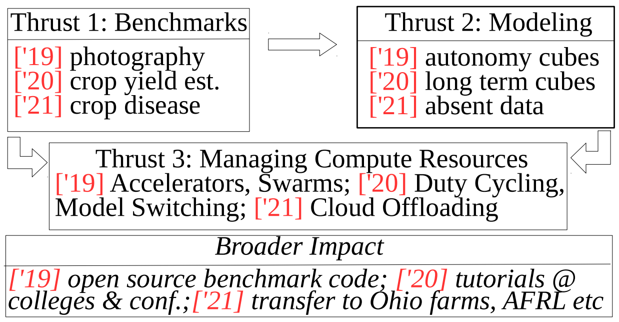
Thrust 1: Benchmarks
['19] photography
['20] crop yield est.
['21] crop disease
Thrust 2: Modeling
['19] autonomy cubes
['20] long term cubes
['21] absent data
Thrust 3: Managing Compute Resources
['19] Accelerators, Swarms; ['20] Duty Cycling,
Model Switching; ['21] Cloud Offloading
Broader Impact
['19] open source benchmark code; ['20] tutorials @ colleges & conf.;['21] transfer to Ohio farms, AFRL etc

Fig. 2: 3-year plan for my proposed research.

ment information from prior executions to find the highest utility images in their state space faster than competing approaches[1]. FAAS operate over long periods of time, and in environments where profile data is hard to collect. I will use online and offline learning techniques to improve the tolerance of autonomy cubes to long term and sparse datasets, allowing FAAS to properly navigate large, complex environments.

I will also create models that allow users to compute mission throughput without flying the FAAS. The key idea is to replay data sensed from real autonomous missions. End user can use these models to compare compute systems.

Using my benchmarks and models, I will study FAAS resource management techniques. I will explore the impacts of accelerators like FPGAs and GPUs, hardware duty cycling, and adaptive model switching on FAAS mission throughput similar to recent work on autonomous driving[2]. I will also experiment with scaling FAAS by the number of UAV, and type and power of compute resources. I will determine efficient mechanisms for scaling FAAS for long term autonomy using duty cycling, cloud offloading, hardware acceleration, and UAV swarms.

Broader Impact: Creating benchmarks for FAAS will help developers create efficient systems that help people like first responders and farmers. Discovering the effects of management approaches on FAAS and providing benchmarks will allow developers to make smart choices when creating FAAS. These smart choices will help improve FAAS mission throughput to the point where technologies like autonomous search and rescue and fully autonomous farming become realities.

To help facilitate the development of new FAAS, I will release and manage all of my software through Github. In order to build an active development community, I will engage students and collaborators at Miami University, Ohio State, and Denison, as well as at research conferences through FAAS talks, tutorials, and demonstrations. I will also share my technology with my US Air Force collaborators through internships.

Preliminary Work: I have designed and implemented a FAAS middleware to facilitate the creation of FAAS benchmarks. I have used this middleware to develop a FAAS benchmark for facial recognition similar to benchmark (1). This benchmark uses autonomy cubes to improved mission throughput by 2.25X compared to competing approaches.

I have developed a simulator for my FAAS middleware that takes aircraft and edge profile information and uses real FAAS execution data to predict mission throughput. This system is able to predict the throughput of benchmark (1) with at most 4% error for the 10 architectures we tested.

Our benchmark was run using a number of architectural configurations, goals, autonomy policies, and search algorithms. We were able to improve throughput by adding hardware resources (extra cores, GPUs), duty cycling GPUs and classification models, improving path finding using autonomy cubes, and providing appropriate autonomy goals. Compared to naive configurations, the combination of our approaches improves mission throughput by 10X.

References:

[1] L. Sanchez-Lopez et al., 'Aerostack', ICUAS, 2016

[2] S.-C. Lin et al., 'The architectural implications of autonomous driving', ASPLOS, 2018.