Autonomous UAV Swarms: Distributed Microservices, Heterogeneous Swarms, and Zoom Maneuvers

Undergraduate Research Thesis

Presented in Partial Fulfillment of the Requirements for graduation "with Honors Research Distinction in Computer Science and Engineering" in the College of Engineering of The Ohio State University

by
Kevyn Angueira Irizarry

The Ohio State University

May 2024

Project Advisor: Dr. Christopher Stewart, Department of Computer Science

1. Abstract

Over the last few years, precision agriculture has greatly benefited from advancements in Unmanned Aerial Vehicles (UAVs). UAVs used in crop mapping allow farmers and researchers to tailor farming practices to the specific needs of individual management zones. Yet despite their benefits, traditional exhaustive approaches to UAV remote sensing are constrained by the low battery capacities of UAVs. To optimize battery usage, we present our work across 3 papers on alternative reinforcement learning (RL) and multi-agent reinforcement learning (MARL) approaches to UAV remote sensing, namely: SoftwarePilot 2.0, heterogeneous swarms, and the Zoom maneuver. Starting with SoftwarePilto 2.0., we introduce a software package that supports scalable autonomous UAV swarms through microservice model design, container deployment technologies, and specialized MARL policies. These improvements to SoftwarePilot 2.0. reduced energy costs by 50% and improved swarm decision-making by 2.1 times from base SoftwarePilot. Moreover, we further explored multi-agent strategies through the extrapolation of multiple health metrics with heterogeneous UAV swarms. Heterogeneous swarms can capture data from multiple types of sensors, e.g. RGB, thermal, multi-spectral, and hyper-spectral cameras, and then extrapolate for various distinct health metrics across the whole field. Our preliminary results showed 90% accuracy from extrapolation from sampling only 40% of the field. Lastly, we proposed a study on the battery and accuracy tradeoffs of Zoom maneuvers. Moreover, Zoom maneuvers or changes in altitude trade battery for increased local accuracy. Our study considers the computational battery cost and flight battery cost tradeoffs of autonomous vs. RL implementations of Zoom maneuvers. Ultimately, this paper provides new insights into the execution and performance of various autonomous and multi-agent UAV remote search strategies

Keywords

unmanned aerial vehicles, edge computing, swarm, agriculture

2. Scalable Distributed Microservices for Autonomous UAV Swarms

2.1 Introduction

Unmanned Aerial Vehicles (UAVs) are inexpensive and maneuverable IoT devices which are quickly changing key industries. UAV can sense over wide areas quickly in 3D space, access areas too dangerous for humans, and can act in groups as swarms to distribute tasks and learn from one another. Recent work has shown that UAV contribute broadly to agriculture, construction, infrastructure inspection, search and rescue, and more [65, 74].

While UAV are conventionally piloted by humans, improvements in onboard hardware and control software have led to automated and autonomous UAV flights. Open source and manufacturer-provided software development kits (SDKs) [30, 62], support libraries [53], and research middlewares [13, 33, 75, 98] provide features for UAV flight control via waypoint missions, data capture, computer vision, and even complex decision-making. Autonomous UAV flight, where UAV make complex decisions in-mission, has been shown to speed up some mission types and allow for new more complex UAV missions [7].

Despite their benefits, UAV suffer from critical bottlenecks. UAVs have short flight times due to small batteries, and limited onboard compute capacity. To cover wide areas, distribute intelligence, and elongate missions, UAV are often flown in swarms. Swarms of UAV can be dispatched across wide areas, learn from each other's observations, and work together to solve complex problems. Swarms of autonomous UAV can accomplish complex tasks quickly and benefit from each other's observations.

Few software packages provide even baseline UAV swarm capabilities [31]. Similarly, few packages and SDKs provide capabilities for autonomy [13, 75]. Some provide simple computer vision routines and automated flight, but none provide support for custom autonomy policies, computer vision, inter-UAV communication, or swarm control. Even complex research middlewares for UAV rarely provide native swarm support or combine it with autonomy. This is because autonomous swarms are hard to manage and scale. Single autonomous UAV already necessitate complex edge hardware and resource management practices [7]. Scaling up from a single UAV to a swarm includes not only considering resource impacts of additional swarm members, but how those impacts compound as members share information and learn.

In this paper, we present our adaptations to an autonomous UAV middleware, SoftwarePilot, which provides mechanisms for creating autonomous UAV. We modified SoftwarePilot to work with cloud-native and edge-appropriate scalable deployment technologies. SoftwarePilot 2.0 leverages these capabilities to deploy, distribute, and manage autonomous UAV swarms across edge clusters.

2.2 Design

SoftwarePilot [13] is a UAV middleware that allows users to implement autonomous missions. As shown in Figure 1, SoftwarePilot decouples mission code into loadable microservices called routines and drivers linked by the SoftwarePilot API. Drivers are application specific APIs that control UAV from different manufacturers, supply pathfinding and AI algorithms, and manage data. Routines are user-code that access drivers via the SoftwarePilot COaP API.

This design already includes scalable elements. By decoupling APIs from user code, users can load and unload drivers dynamically based on their needs. SoftwarePilot's original design did not, however, consider a number of important management concerns relating to UAV swarms. First, SoftwarePilot 1.0 UAVs are controlled by a single central virtual machine on which all microservices run, making it portable but difficult to distribute. Second, while SoftwarePilot was made to build autonomous UAV, it was not made to operate swarms. It includes no deployment mechanisms for swarms across edge clusters, network overlay features, or distributed systems management software. Third, it does not include any services for leveraging swarm intelligence.

We addressed these three shortfalls in SoftwarePilot 2.0. First, we converted all SoftwarePilot microservices from independent applications to Docker containers. This change is significant in that it allows for increased portability without virtual machine overhead, it decentralizes microservices from inside the virtual machines where they previously ran, and allows SoftwarePilot to benefit from existing container deployment technologies. SoftwarePilot uses Kubernetes to deploy its containers across clusters of edge devices. Kubernetes features allow SoftwarePilot containers to communicate via overlay networks, set resource limitations, deploy to specific cluster nodes, and manage the experiment lifecycle in ways that our prior virtualization technique

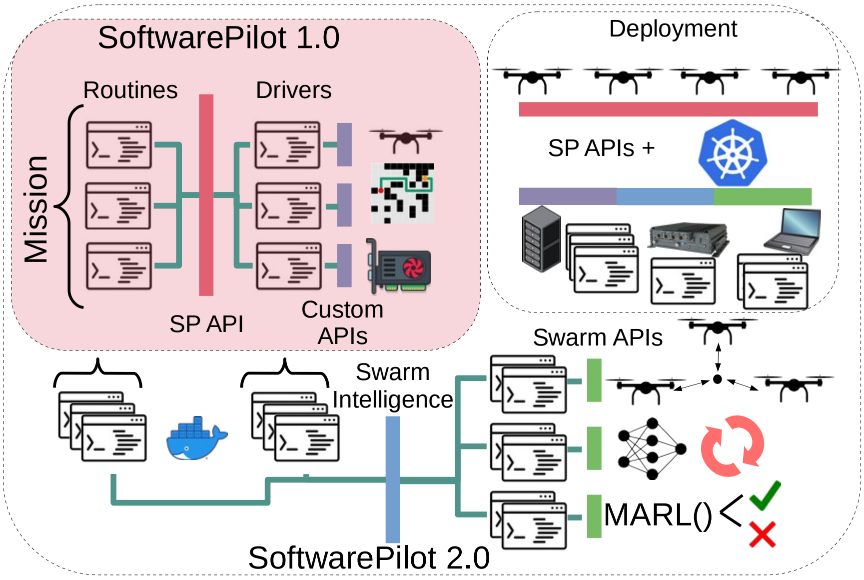

Fig. 2.1: SoftwarePilot 2.0 adds new swarm intelligence APIs and deployment management.

did not.

SoftwarePilot has also added support for UAV swarms and multi-agent reinforcement learning (MARL). SoftwarePilot 1.0 had no inherent way to instantiate multiple UAV and allow them to communicate. Using the new SoftwarePilot Swarm Intelligence API, users can instantiate and control multiple UAV, link them to microservices, and control them via centralized or distributed programming logic. This new API and Python package can leverage the kubernetes overlay network to communicate with UAV control and autonomy microservices across the cluster.

Additional swarm intelligence APIs can use this inter-UAV communication to implement global multi-agent reinforcement learning policies. SoftwarePilot 1.0 implemented autonomy policies on a drone by drone basis. SoftwarePilot 2.0 now allows users to implement MARL policies on top of single-UAV autonomy policies. MARL policies can track reward and mission progress at a global level. They combine observations from multiple agents to make better decisions and retrain models over time.

2.3 Early Experiences

SoftwarePilot 2.0 has been used to implement UAV swarm applications in agriculture. In Autumn 2021, we flew over 150 UAV Swarm missions using

SoftwarePilot 2.0's MARL and swarm features over crop fields in Ohio [9]. We used a swarm of autonomous UAVs to sample a crop field to assess soybean leaf defoliation, an important crop health indicator. We used SoftwarePilot drivers for the DJI Mavic UAV combined with custom drivers for defoliation detection using DefoNet [112]. To build an intelligent UAV swarm, we developed SoftwarePilot's swarm intelligence APIs. We implemented APIs for multi-agent Q-learning and online model updating which were containerized and distributed across our edge hardware using Docker and Kubernetes.

SoftwarePilot 2.0's new swarm mechanisms are efficient. For our deployment configuration, we found that SoftwarePilot 2.0 used 2X less energy than a swarm created using SoftwarePilot 1.0. This was due entirely to the use of our Kubernetes management platform which automatically duty-cycles cluster resources during resource troughs. We also found that the addition of MARL as sped up missions by up to 2.1X via better decision-making. In the near future, we plan to release SoftwarePilot 2.0 as an open source project to help facilitate the building and deployment of fully autonomous UAV swarms.

3. Multi-Agent Reinforcement Learning for Heterogeneous UAV Swarm Enabling Detailed Crop Health Assessment

3.1 Introduction

The rapid growth of the global population and the increasing demand for food necessitates swift advancements in agricultural practices to meet the pressing needs of humanity. Projections indicate that the combination of population expansion and heightened per capita food consumption will require a substantial 70% increase in agricultural yields by the year 2050 [34, 36]. However, the looming challenge of climate change is exacerbating the complexity of farming, as it contributes to stressors on crop health, such as drought, diseases, and pest infestations [50]. These adverse effects are projected to lead to a considerable reduction in crop yields by up to 11% by 2050 [58].

Precision agriculture occupies a pivotal role in meeting the escalating global food demand, as it centers upon the optimized utilization of resources and the maximization of yields through the strategic integration of technology. This, in turn, contributes significantly to the stabilization and reliability of the global food supply chain. Precision agriculture is a promising step toward improving efficiency and reducing adverse impacts of agriculture production [107]. It assesses the variation across the crop fields and divides the field into multiple management zones. So they can be treated efficiently and effectively [32,64]. The advent of digital agriculture, or data-driven precision agriculture, employs a suite of tools including remote sensors (e.g., satellites and UAVs), in-field sensors (such as embedded soil sensors), and data processing techniques (e.g., machine learning) [48]. This combination informs decisions related to planting, harvesting, and crop treatment, all aimed at maximizing yield and minimizing the environmental impact of agricultural activities. Frequent sensing using these technologies enables the detection of crop health stress due to factors like drought and heat, the identification of diseases, pests, and other harmful phenomena [20, 95, 109, 113]. A pivotal task in digital agriculture involves transforming the collected data into health maps, providing valuable geospatial insights into crop health, and guiding effective crop treatment strategies, commonly referred to as crop scouting.

Traditionally, data acquisition is approached through two main methods: human piloting of UAVs to capture high-resolution images or UAVs autonomously scouting the entire fields. Human pilots, while capable of capturing accurate data, tend to escalate operating costs due to the need for frequent field mapping. Conversely, the autonomous UAV scouting method is cost-effective, but it often leads to redundant data due to a 60-70% side overlap in captured images. Moreover, both approaches necessitate frequent battery replacements due to limited flight times [35], which subsequently extend execution times and have an impact on profit margins.

The UAVs are equipped with various imaging sensors, including RGB, multi-spectral, thermal, and hyper-spectral. RGB cameras are well-suited for tasks such as growth prediction, biomass estimation, and canopy height measurement. On the other hand, multi-spectral cameras excel in early detection of drought stress, identification of pests, yield prediction, and their combination with thermal and hyper-spectral data enables estimation of nutrient status, pathogen presence, and weed detection [51]. Unlike RGB cameras, multi-spectral cameras capture both visible and invisible light spectra, enhancing the assessment of crop conditions and thereby enabling more informed agricultural decisions [69, 106]. Hyper-spectral and thermal cameras capture distinct bands of the invisible light spectrum. Hyper-spectral sensors are particularly effective in the early detection of pathogens and diseases [54], while thermal cameras are effective in identifying drought stress in crops [114]. RGB and multi-spectral cameras are commonly used whereas hyper-spectral and thermal are less common due to relatively higher costs [1].

Given the limited payload capacity of UAVs, they are constrained to carrying only one imaging sensor at a time [96]. Consequently, achieving a comprehensive analysis of a crop field necessitates the deployment of multiple drones. However, employing multiple drones for scouting an entire field introduces additional operational and maintenance expenses that may outweigh the potential gains. The increased costs associated with utilizing multiple drones can present a challenge in terms of maintaining profitability.

Contributions: In this paper, we present an efficient method for detailed analysis of whole-field without exhaustively scouting entire field. We employ a swarm of heterogeneous UAVs with distinct capabilities. We utilize multi-agent reinforcement learning to scout crucial areas through competing rewards, as a result battery replacements and payload requirements are minimized. The collected data is combined and extrapolated to provide deeper insights on crop health eliminating the need of exhaustive scouting of the whole field.

3.2 Methodology

This approach can be divided into two major alternating components: 1) RL algorithm for exploration and sensing, 2) extrapolation algorithm for creating a health map from sensed data. Each UAV will continually cycle between selecting their next location based on the estimated health map and updating the estimated health map by extrapolating from sensed data. Together all agents will pool their results into a combined extrapolated health map.

3.2.1 Reinforcement Learning Algorithm

We use a modified version of Q-learning and a MARbLE architecture to select the path to be sensed through multi-agent reinforcement learning [10]. As shown in figure 3.1 the UAVs explore during the initial phase and then try to maximize the utility of visiting a particular management zone. The states are the x and y coordinates of the management zones and the utility or reward of each action is the error between the predicted values from the CNN and the observed values. Ultimately, the Q-table is updated with the reward from the combined goal and budget preferences from the MARbLE algorithm. These rewards are updated using Bellman's Equation, as shown below.

$$Q(s_i, a_i) = (1 - \alpha) * Q(s_i, a_i) + \\ \alpha * [R(s_i, a_i, s_{i+1}) * \gamma \max(Q(s_{i+1}, a_{i+1}))] \tag{3.1}$$

Equation 3.1 calculates the maximum reward with learning rate α and a discount factor γ taking into account the immediate and long-term rewards.

To create a generalized and transferable model we use filters while populating the Q-table. The observed rewards are quantified based on their variance such that the observed pattern can be transferred to different fields as well.

3.2.2 Extrapolation

While the UAV explores the field, the health map is continuously extrapolated using the newly gathered data at each step. This extrapolation is crucial, as it provides an accurate foundation for decision-making within the RL algorithm. The RL algorithm aims to maximize the percentage error gain between predicted and ground truth values, thereby systematically reducing the error associated with the projected health map. To extrapolate

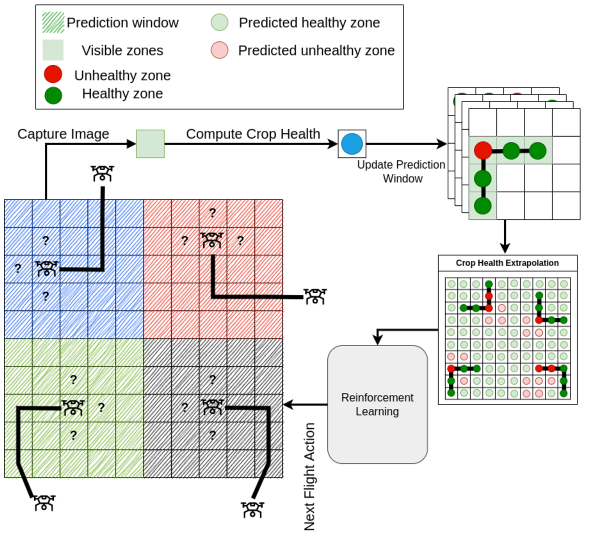

Fig. 3.1: An illustration of crop scouting using 4 UAVs with distinct capabilities through CNN extrapolation and Reinforcement learning

the health maps, we employ Convolutional Neural Networks (CNN). This extrapolation is built on the premise that distinct sensor readings correspond to various aspects of plant health. This concept is akin to human health diagnostics, where different tests reveal diverse health issues that may also be interconnected. For instance, in humans the identification of calcium deficiency through the Total Calcium Test (TCT) can potentially indicate the presence of osteoporosis [59].

We quantify comprehensive crop health by combining data from different health indicators such as NDVI, NDWI, and GLI. This combination allows us to present an overarching picture of the crop field's health. Moreover, the extrapolated individual health maps associated with these indicators offer more intricate insights, aiding in the identification of precise measures necessary to enhance the crop's well-being. The quantification of overall health based on different health indicators (NDVI, NDWI, GLI). Furthermore, the overall health map complements the decision making in multi-agent reinforcement learning.

The design of the CNN is based on U-net Architecture [73]. The input to the CNN is observed health maps, and the output is fully predicted health maps.

3.3 Preliminary Results

Our previous work using RL for crop scouting and CNN for extrapolation with single agent employing RGB camera provide promising results [110]. A health map is predicted with 90% accuracy by scouting only 40% field and hence reducing labor costs 4.8 times and boosting profit by 36%.

4. Profiling Edge Resource Demands of Zoom Maneuvers for Autonomous Unmanned Aerial Vehicles

4.1 Introduction

Autonomous unmanned aerial vehicles (AUAVs) conduct complex missions without human piloting [6, 7, 11, 60, 63, 75]. Services like Percepto employ AUAVs to inspect large industrial facilities for gas leaks, overheating equipment, and degraded structures, allowing companies to take early actions to address these problems [63].Similarly, AUAVs in digital agriculture capture high-resolution images of crop fields and convert them into maps that characterize crop health [7, 10, 21]. Additionally, AUAVs have further applications in smart cities, forestry, wildlife conservation, and military. Like unmanned aerial vehicles, AUAVs can conduct missions that are too risky for manned aircraft. As a type of unmanned aerial vehicle, AUAVs are not piloted remotely by human operators. Instead, they use data captured by onboard sensors (e.g., cameras and GPS) to decide where to fly next and when to land. AUAVs rely on platforms that allow software to issue commands during flight. Such platforms are provided aircraft manufacturers (e.g., DJI and Parrot), open-source flight-control systems (e.g., Pixhawk and Ardupilot), and AI-driven platforms for navigation (e.g., SoftwarePilot and Aerostack).

Most traditional flight-control platforms only support UAVs through automated waypoint missions or predetermined flight paths. This approach utilized by automated UAVs usually relies on exhaustively scouting all states (e.g. GPS locations) for an entire region. However, if adjacent or similar states convey the same or correlated information, exhaustive automated approaches waste limited battery resources without contributing compensatory benefits. By contrast, reinforcement learning (RL) AUAVs approaches exploit the correlation of adjacent states to maximize the data map accuracy while minimizing the number of states required to be visited. These RL approaches conserve battery by requiring a lower number of states. For this reason, it is worth researching the possible benefits of the RL approach, but its implementation would require novel resources to support it. To address this lack of resources, our group previously developed SoftwarePilot 2.0. This is a middleware designed for rapid implementation of both explicit UAV and UAV swarm autonomy [37].The utility of SoftwarePilot 2.0 is in providing a general RL solution to multiple competing goals and budgets in an autonomous UAV context. SoftwarePilot 2.0's crop health mapping balances crop stressors and health metrics that are competing for budgets, maximizing mapping accuracy within the given budget.

This paper proposes improvements to AUAV RL mapping approaches through an exploration into the zoom maneuver [105]. AUAVs can adjust their altitude during flight to sense their surroundings in greater detail. For example, using a 4K HD camera, an AUAV flying 5 meters will capture images where each pixel represents a 2-millimeter area and the picture spans 2-3 meters. Conversely, flying at 100 meters yields coarse images that can cover a hectare. Incorporating zoom maneuvers can significantly improve the execution of missions by (1) unveiling previously obscured data or (2) reducing the number of visited waypoints and saving crucial battery resources.

4.2 Proposed Study

The zoom maneuver includes any policies that adjust the altitude of UAVs or improve data accuracy while localizing data recollection. This paper aims to perform a study on the impacts of combinations of exploration and zoom policies on crop health map accuracy. We will outline a procedure to measure algorithm efficiency in terms of accuracy and resource management. Improvements in health map accuracy from zoom maneuvers will be transferable to other models for autonomous health mapping and will improve the efficiency of UAV use in precision agriculture. The results of the study will inform the utility and implementation of the zoom maneuver into SoftwarePilot 2.0 [37, 47]. In total we defined four key implementations across our study: Auto Exploration (1)Auto Zoom, (2)Auto Exploration RL Zoom, (3)RL Exploration Auto Zoom, and (4)RL Exploration RL Zoom. We define Auto as automated, i.e. a preset route, and RL as a reinforcement learning policy, making live informed decisions. These four implementations were chosen to compare and contrast the improvements of RL strategies for exploration and zoom independently over autonomy, and the improvements of RL strategies for joint exploration and zoom over other methods. Each of the methodologies above must compete for the highest accuracy within the same constraints for battery, time, and number of states. These arrangements allow us to compare different reinforcement learning strategies against a set

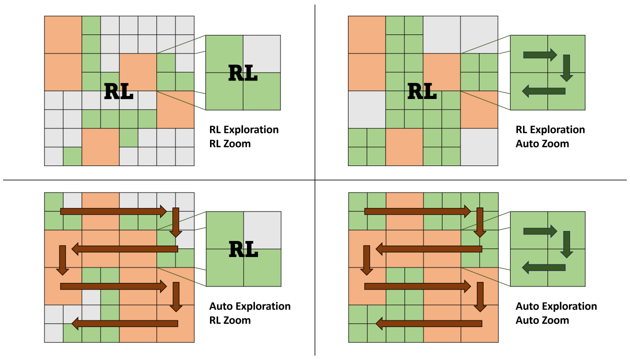

Fig. 4.1: The four methods being studied: Auto Exploration Auto Zoom, Auto Exploration RL Zoom, RL Exploration Auto Zoom, and RL Exploration RL Zoom

baseline. We now define the different strategies:

1. **Automated Exploration + Automated Zoom** uses an automated lawnmower pattern to represent exhaustive search for both exploration and zoom. It employs a random policy to decide when to employ the zoom maneuver. This method suffers from the highest flight costs as it must visit at least one state per management zone and all lower states on a zoom. However, as this model does not employ RL, it suffers no hovering costs waiting on the next action from the edge device.
2. **Automated Exploration + RL Zoom** uses an automated lawnmower pattern for exploration and an RL model for zoom. Its RL indicates when to zoom and what lower states to explore. This method must explore at least each state from every management zone; however, it may choose how many states to explore during zoom.
3. **RL Exploration + Automated Zoom** uses RL for exploration and an automated lawnmower pattern for zoom. Its RL strategically decides its next action for exploration while it employs a random policy to decide when to zoom. This method may reduce its total flight costs through strategic location, however it cannot control when to zoom and must explore all four lower states.
4. **RL Exploration + RL Zoom uses RL** for both exploration and zoom. It lumps lower and higher states into as possible waypoints for a single model to learn from, the model may freely choose actions to move to lower states as if they were adjacent states. This method may reduce its total flight cost through strategic locations, however it incurs the highest

decision hovering costs as it must wait for a response from the edge for each action.

Experimental Plan: These experiments will allow us to profile the performances of different combinations of exploration and zoom strategies. For each we method we will track the battery consumption and computational costs. Automated searches from method 1 serve as the baseline and represent the traditional exhaustive approaches. By comparing methods 2 and 3 to method 1 we can measure the improvements in mapping accuracy from RL Zoom and RL Exploration respectively. Additionally comparing method 4 to method 1 demonstrates the total improvements from the baseline. This methodology may be then repeated with multiple versions of RL algorithms, different automated strategies to measure the impacts of the Zoom maneuver across different conditions and develop the most optimal solution for the given task. The results of this study will then inform future implementation of the *Zoom maneuver* into SoftwarePilot 2.0 and other AUAV RL strategies.
Acknowledgments: This work was funded by NSF Grant OAC-2112606 and the Ohio Soybean Council.

5. Conclusion

Throughout this paper, we discuss our advancements in autonomous and multi-agent UAV remote sensing strategies. Starting with SoftwarePilot 2.0, we introduced a framework for scalable autonomous UAV swarms. Moreover, SoftwarePilot 2.0 leverages Docker and Kubernetes to manage and optimize resources for scalable UAV control and autonomy microservices. Additionally, SoftwarePilot 2.0. improves on swarm decision-making through global MARL policies. Through these improvements, SoftwarePilot 2.0 reduced energy consumption by 50% and improved swarm decision-making by 2.1 times when compared to base SoftwarePilot.

Further enhancements were realized by introducing heterogeneous swarms. Heterogeneous UAV swarms allow users to capture and combine various distinct sources of data, e.g. RGB, thermal, multi-spectral, and hyper-spectral cameras, to develop more robust and comprehensive health metrics. We developed strategies for whole-field extrapolation of distinct health metrics from partial sampling using heterogeneous swarms. Preliminary testing showed a 90% accuracy by sampling only 40% of the field. Although promising, additional testing is required to further optimize the complex MARL policies of heterogeneous swarms.

Lastly, we proposed a study on the implementation of Zoom maneuvers. Zoom maneuvers or changes in altitude, trade battery for increased local accuracy. Moreover, many key features, like leaf defoliation, greatly benefit from Zoom maneuvers. Our proposed study maps the battery and accuracy tradeoffs from different implementations of the Zoom maneuver, namely autonomous vs. reinforcement learning approaches. Moreover, autonomous or predefined patterns minimize computational costs on battery meanwhile reinforcement learning strategies minimize flight costs on battery. The results of this study will greatly inform how the Zoom maneuver will be implemented and will further optimize resources.

Acknowledgments: This work was funded by NSF Grant OAC-2112606 and the Ohio Soybean Council. The research presented in this thesis was conducted in the ReRout Lab at The Ohio State University under the supervision of Professor Christopher Stewart. The ideas and concepts studied here build upon years of research by prior lab members. I have cited many of those papers in the bibliographic references to acknowledge their impact.

Bibliography

[1] N. Amarasingam, A. S. Ashan Salgadoe, K. Powell, L. F. Gonzalez, and S. Natarajan. A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops. *Remote Sensing Applications: Society and Environment*, 26:100712, 4 2022.

[2] N. Babu and C. Stewart. Energy, latency and staleness tradeoffs in ai-driven iot. In *ACM Workshop on Hot Topics in Web of Things*, 2019.

[3] N. Babu and C. Stewart. Revisiting online scheduling for ai-driveninternet of things. In *ACM Symposium on Edge Computing*, 2019.

[4] A. Baietto, J. Boubin, P. Farr, T. J. Bihl, A. M. Jones, and C. Stewart. Lean neural networks for autonomous radar waveform design. *Sensors*, 22(4):1317, 2022.

[5] C. M. Barnes, K. Bellman, J. Botev, A. Diaconescu, L. Esterle, C. Gruhl, C. Landauer, P. R. Lewis, P. R. Nelson, A. Stein, et al. Chariot-towards a continuous high-level adaptive runtime integration testbed. In *2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W)*, pages 52–55. IEEE, 2019.

[6] B. Boroujerdian, H. Genc, S. Krishnan, W. Cui, A. Faust, and V. Reddi. Mavbench: Micro aerial vehicle benchmarking. In *MICRO*, 2018.

[7] J. Boubin, N. Babu, C. Stewart, J. Chumley, and S. Zhang. Managing edge resources for fully autonomous aerial systems. In *ACM Symposium on Edge Computing*, 2019.

[8] J. Boubin, C. Burley, P. Han, B. Li, B. Porter, and C. Stewart. Programming and deployment of autonomous swarms using multi-agent reinforcement learning. *arXiv preprint arXiv:2105.10605*, 2021.

[9] J. Boubin, C. Burley, P. Han, B. Li, B. Porter, and C. Stewart. Marble: Multi-agent reinforcement learning at the edge for digital agriculture. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 68–81. IEEE, 2022.

[10] J. Boubin, C. Burley, P. Han, B. Li, B. Porter, and C. Stewart. MARbLE: Multi-Agent Reinforcement Learning at the Edge for Digital Agriculture. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 68–81. IEEE, 12 2022.

[11] J. Boubin, J. Chumley, C. Stewart, and S. Khanal. Autonomic computing challenges in fully autonomous precision agriculture. In *IEEE International Conference on Autonomic Computing*, 2019.

[12] J. Boubin, C. Stewart, S. Zhang, N. Babu, and Z. Zhang. Softwarepilot, 2019.

[13] J. Boubin, C. Stewart, S. Zhang, N. T. Babu, and Z. Zhang. Softwarepilot. `http://github.com/boubinjg/softwarepilot`, 2019.

[14] J. Boubin, S. Zhang, V. Mandadapu, and C. Stewart. Characterizing computational workloads in uav applications. In *IEEE Internet-of-Things Design and Implementation*, 2018.

[15] J. Boubin, Z. Zhang, J. Chumley, and C. Stewart. Adaptive deployment for autonomous agricultural uav swarms. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 1089–1095, 2022.

[16] A. Chakrabarti, S. Parthasarathy, and C. Stewart. A pareto framework for data analytics on heterogeneous systems: Implications for green energy usage and performance. In *2017 46th International Conference on Parallel Processing (ICPP)*, pages 533–542. IEEE, 2017.

[17] A. Chakrabarti, S. Parthasarathy, and C. Stewart. A pareto framework for data analytics on heterogeneous systems: Implications for green energy usage and performance. In *IEEE International Conference on Parallel Processing*, 2017.

[18] A. Chakrabarti, C. Stewart, and S. Parthasarathy. Green- and heterogeneity-aware partitioning for data analytics. In *IEEE International Workshop on Green and Sustainable Networking and Computing*, 2016.

[19] D. Chiu, C. Stewart, and B. McManus. Electric grid balancing through low-cost workload migration. In *ACM GreenMetrics Workshop*, 2012.

[20] S. Das, S. Chapman, J. Christopher, M. R. Choudhury, N. W. Menzies, A. Apan, and Y. P. Dang. UAV-thermal imaging: A technological breakthrough for monitoring and quantifying crop abiotic stress to help sustain productivity on sodic soils – A case review on wheat. *Remote Sensing Applications: Society and Environment*, 23:100583, 8 2021.

[21] J. del Cerro, C. Cruz Ulloa, A. Barrientos, and J. de León Rivas. Unmanned aerial vehicles in agriculture: A survey. *Agronomy*, 11(2):203, 2021.

[22] N. Deng. Carbon-aware cloud application. In *IEEE Network Operations and Management Symposium*, 2012.

[23] N. Deng, C. Stewart, D. Gmach, and M. Arlitt. Policy and mechanism for carbon-aware cloud applications. In *IEEE Network Operations and Management Symposium*, 2012.

[24] N. Deng, C. Stewart, J. Kelley, D. Gmach, and M. Arlitt. Adaptive green hosting. In *IEEE International Conference on Autonomic Computing*, 2012.

[25] N. Deng, C. Stewart, and J. Li. Concentrating renewable energy in grid-tied datacenters. In *IEEE International Symposium on Sustainable Systems Technology*, 2011.

[26] N. Deng, Z. Xu, C. Stewart, and X. Wang. From the outside looking in: Probing web apis to build detailed workload profiles. In *USENIX Workshop on Feedback Computing*, 2014.

[27] N. Deng, Z. Xu, C. Stewart, and X. Wang. Tell-tale tails: Decomposing response times for live internet services. In *IEEE International Green and Sustainable Computing Conference*, 2015.

[28] P. Derosa, K. Shen, C. Stewart, and J. Pearson. Realism and simplicity: Disk simulation for instructional os performance evaluation. In *ACM SIGCSE Technical Symposium*, 2006.

[29] J. Ding, R. Cao, I. Saravanan, N. Morris, and C. Stewart. Characterizing service level objectives for cloud services: Realities and myths. In *2019 IEEE International Conference on Autonomic Computing (ICAC)*, pages 200–206. IEEE, 2019.

[30] DJI. Prerequisites-dji mobile sdk documentation. https://developer.dji.com/, 2018.

[31] EpiSci. Swarmsense: Intelligent uav swarms. https://www.episci.com/product/swarmsense/.

[32] R. Gebbers and V. I. Adamchuk. Precision Agriculture and Food Security. *Science*, 327(5967):828–831, 2 2010.

[33] H. Genc, Y. Zu, T. Chin, M. Halpern, and V. J. Reddi. Flying iot: Toward low-power vision in the sky. *IEEE Micro*, 37(6):40–51, November 2017.

[34] H. C. J. Godfray, J. R. Beddington, I. R. Crute, L. Haddad, D. Lawrence, J. F. Muir, J. Pretty, S. Robinson, S. M. Thomas, and C. Toulmin. Food security: The challenge of feeding 9 billion people. *Science*, 327(5967):812–818, 2 2010.

[35] P. J. Hardin and R. R. Jensen. Small-Scale Unmanned Aerial Vehicles in Environmental Remote Sensing: Challenges and Opportunities. *GIScience & Remote Sensing*, 48(1):99–111, 1 2011.

[36] L. T. Hickey, A. N. Hafeez, H. Robinson, S. A. Jackson, S. C. M. Leal-Bertioli, M. Tester, C. Gao, I. D. Godwin, B. J. Hayes, and B. B. H. Wulff. Breeding crops to feed 10 billion. *Nature Biotechnology*, 37(7):744–754, 7 2019.

[37] K. A. Irizarry, Z. Zhang, C. Stewart, and J. Boubin. Scalable distributed microservices for autonomous uav swarms. In *Proceedings of the 23rd International Middleware Conference Demos and Posters*, pages 1–2, 2022.

[38] M. Jordan and C. Stewart. Adaptive middleware for component-level deployment. In *Workshop on Adaptive and Reflective Middleware*, 2005.

[39] S. Kambhampati, J. Kelley, W. C. Stewart, C. Stewart, and R. Ramnath. Managing tiny tasks for data-parallel, subsampling workloads. In *IEEE International Conference on Cloud Engineering*, 2014.

[40] J. Kelley and N. Morris. Rapid in-situ profiling of colocated workloads. In *IEEE Workshop on Data Center Performance*, 2019.

[41] J. Kelley and C. Stewart. Balanced and predictable networked storage. In *IEEE International Workshop on Data Center Performance*, 2013.

[42] J. Kelley, C. Stewart, S. Elnikety, and Y. He. Cache provisioning for interactive nlp services. In *ACM Large and Distributed Systems and Middleware*, 2013.

[43] J. Kelley, C. Stewart, N. Morris, D. Tiwari, Y. He, and S. Elnikety. Obtaining and managing answer quality for online data-intensive services. In *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, 2017.

[44] J. Kelley, C. Stewart, D. Tiwari, and S. Gupta. Adaptive power profiling for many-core hpc architectures. In *IEEE International Conference on Autonomic Computing*, 2016.

[45] J. Kelley, C. Stewart, D. Tiwari, Y. He, S. Elnikey, and N. Morris. Measuring and managing answer quality for online data-intensive services. In *IEEE International Conference on Autonomic Computing*, 2015.

[46] T. Kelly, K. Shen, A. Zhang, and C. Stewart. Operational analysis of parallel servers. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2008.

[47] T. P. Kelly, A. X. Zhang, and C. C. Stewart. Determining performance of an application based on transactions, May 18 2010. US Patent 7,720,955.

[48] L. Klerkx, E. Jakku, and P. Labarthe. A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. *NJAS: Wageningen Journal of Life Sciences*, 90-91(1):1–16, 12 2019.

[49] J. Kline, C. Stewart, T. Berger-Wolf, M. Ramirez, S. Stevens, R. R. Babu, N. Banerji, A. Sheets, S. Balasubramaniam, E. Campolongo, et al. A framework for autonomic computing for in situ imageomics. In *2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 11–16. IEEE, 2023.

[50] M. S. Kukal and S. Irmak. Climate-Driven Crop Yield and Yield Variability and Climate Change Impacts on the U.S. Great Plains Agricultural Production. *Scientific Reports*, 8(1):3450, 2 2018.

[51] W. H. Maes and K. Steppe. Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture. *MPDI*, 2018.

[52] V. Mandadapu and C. Stewart. Using game theory to manage self-aware unmanned aerial systems. In *IEEE International Workshops on Foundations and Applications of Self* Systems*, 2018.

[53] L. Meier, J. Camacho, B. Godbolt, J. Goppert, L. Heng, M. Lizarraga, et al. Mavlink: Micro air vehicle communication protocol. *Online]. Tillgänglig: http://qgroundcontrol. org/mavlink/start.[Hämtad 2014-05-22]*, 2013.

[54] P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka, and E. Hernandez. Plant Disease Detection Using Hyperspectral Imaging. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 11 2017.

[55] N. Morris, S. Renganathan, C. Stewart, R. Birke, and L. Chen. Sprint ability: How well does your software exploit bursts in processing capacity? In *IEEE International Conference on Autonomic Computing*, 2016.

[56] N. Morris, C. Stewart, R. Birke, and L. Chen. Early work on modeling computational sprinting. In *ACM Symposium on Cloud Computing*, 2017.

[57] N. Morris, C. Stewart, L. Chen, R. Birke, and J. Kelley:. Model-driven computational sprinting. In *ACM European Conference on Computer Systems*, 2018.

[58] G. C. Nelson, H. Valin, R. D. Sands, P. Havlík, H. Ahammad, D. Deryng, J. Elliott, S. Fujimori, T. Hasegawa, E. Heyhoe, P. Kyle, M. Von Lampe, H. Lotze-Campen, D. Mason d'Croz, H. van Meijl, D. van der Mensbrugghe, C. Müller, A. Popp, R. Robertson, S. Robinson, E. Schmid, C. Schmitz, A. Tabeau, and D. Willenbockel. Climate change effects on agriculture: Economic responses to biophysical shocks. *Proceedings of the National Academy of Sciences*, 111(9):3274–3279, 3 2014.

[59] B. C. Nordin and H. A. Morris. The Calcium Deficiency Model for Osteoporosis. *Nutrition Reviews*, 47(3):65–72, 4 2009.

[60] Nothrop-Grumman. Autonomous Systems. `https://www.northropgrumman.com/what-we-do/air/autonomous-systems`, 2023.

[61] S. Ockerman, J. Wu, C. Stewart, and Z. Zhang. A reflection on ai model selection for digital agriculture image datasets. In *2nd AAAI Workshop on AI for Agriculture and Food Systems*, 2023.

[62] Parrot. Parrot olympe sdk. https://developer.parrot.com/docs/olympe/index.html.

[63] Percepto Inc. The Differences Between UAV, UAS, and Autonomous Drones. `https://percepto.co/what-are-the-differences-between-uav-uas-and-autonomous-drones/`, 2023.

[64] F. J. Pierce and P. Nowak. Aspects of precision agriculture. *Advances in agronomy*, 67:1–85, 1999.

[65] C. Qu, J. Boubin, D. Gafurov, J. Zhou, N. Aloysius, h. Nguyen, and P. Calyam. Uav swarms in smart agriculture: Experiences and opportunities. In *2022 IEEE International Conference on eScience*. IEEE, 2022.

[66] R. Rao, C. Stewart, A. Perez, and S. Renganathan. Assessing learning behavior and cognitive bias from web logs. In *IEEE Frontiers in Education*, 2018.

[67] S. Renganathan, C. Stewart, A. Perez, R. Rao, and B. Braaten. Preliminary results on an interactive learning tool for early algebra education. In *IEEE Frontiers in Education*, 2017.

[68] M. E. Rodgers, E. Franz, A. Chalker, D. E. Hudak, C. Stewart, R. Machiraju, S. Chang, and C. Smith. Data commons to support university-wide cross discipline research. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*, pages 1–2, 2019.

[69] J. Rodríguez, I. Lizarazo, F. Prieto, and V. Angulo-Morales. Assessment of potato late blight from UAV-based multispectral imagery. *Computers and Electronics in Agriculture*, 184:106061, 2021.

[70] E. Romero, C. Stewart, A. Li, K. Hale, and N. Morris. Bolt: Fast inference for random forests. In *Proceedings of the 23rd ACM/IFIP International Middleware Conference*, pages 94–106, 2022.

[71] E. Romero, C. Stewart, and N. Morris. Fast inference services for alternative deep learning structures. In *ACM Symposium on Edge Computing*, 2019.

[72] E. Romero-Gainza and C. Stewart. Ai-driven validation of digital agriculture models. *Sensors*, 23(3):1187, 2023.

[73] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[74] L. Ruetten, P. A. Regis, D. Feil-Seifer, and S. Sengupta. Area-optimized uav swarm network for search and rescue operations. In *2020 10th annual computing and communication workshop and conference*, pages 0613–0618. IEEE, 2020.

[75] J. L. Sanchez-Lopez, R. A. S. Fernández, H. Bavle, C. Sampedro, M. Molina, J. Pestana, and P. Campoy. Aerostack: An architecture and open-source software framework for aerial robotics. In *International Conference on Unmanned Aircraft Systems*, 2016.

[76] K. Shen, C. Stewart, C. Li, and X. Li. Reference-driven performance anomaly identification. In *ACM International Conference on Measurement and Modeling of Computer Systems*, 2009.

[77] K. Shen, A. Zhang, T. Kelly, and C. Stewart. Operational analysis of processor speed scaling. In *Brief Announcement at the Symposium on Parallelism in Algorithms and Architectures*, 2008.

[78] K. Shen, M. Zhong, S. Dwarkadas, C. Li, C. Stewart, and X. Zhang. Hardware counter driven on-the-fly request signatures. In *Conference on Architectural Support for Programming Languages and Operating Systems*, 2008.

[79] C. Stewart. Continuous adaptive runtime integration testbed for complex and autonomous systems. In *IEEE National Aerospace and Electronics Conference*, 2019.

[80] C. Stewart, A. Chakrabarti, and R. Griffith. Zoolander: Efficiently meeting very strict, low-latency slos. In *IEEE International Conference on Autonomic Computing*, 2013.

[81] C. Stewart, T. Kelly, and A. Zhang. Exploiting nonstationarity for performance prediction. In *European Conference on Computer Systems*, 2007.

[82] C. Stewart, T. Kelly, A. Zhang, and K. Shen. A dollar from 15 cents: Cross-platform management for internet services. In *USENIX Annual Technical Conference*, 2008.

[83] C. Stewart, M. Leventi, and K. Shen. Empirical examination of a collaborative web application. In *2008 IEEE International Symposium on Workload Characterization*, pages 90–96. IEEE, 2008.

[84] C. Stewart and J. Li. Power provisioning for diverse datacenter workloads. In *ACM Workshop on Energy-Efficient Design*, 2011.

[85] C. Stewart and J. Li. Power-utilization provisioning for data centers. In *IEEE International Green and Sustainable Computing Conference*, 2015.

[86] C. Stewart, N. Morris, L. Chen, and R. Birke. Performance modeling for short-term cache allocation. In *Proceedings of the 51st International Conference on Parallel Processing*, pages 1–11, 2022.

[87] C. Stewart and K. Shen. Performance modeling and system management for multi-component online services. In *Symposium on Networked Systems Design and Implementation*, 2005.

[88] C. Stewart and K. Shen. Configuration-space performance anomaly depiction. In *Workshop on Large Scale Distributed Systems and Middleware*, 2008.

[89] C. Stewart and K. Shen. Some joules are more precious than others: Managing renewable energy in the datacenter. In *ACM Workshop on Power Aware Computing and Systems*, 2009.

[90] C. Stewart, K. Shen, S. Dwarkadas, M. L. Scott, and J. Yin. Profile-driven component placement for cluster-based online services. In *International Middleware Conference*, 2004.

[91] C. Stewart, K. Shen, A. Iyengar, and J. Yin. Entomomodel: Understanding and avoiding performance anomaly manifestations. In *IEEE Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2010.

[92] C. Stewart, M. Zhong, K. Shen, and T. O'Neill. Comprehensive depiction of configuration-dependent performance anomalies in distributed server systems. In *USENIX Hot Topics in System Dependability*, 2006.

[93] M. Taylor, J. Boubin, H. Chen, C. Stewart, and F. Qin. A study on software bugs in unmanned aircraft systems. In *IEEE International Conference on Unmanned Aircraft Systems*, 2021.

[94] M. Taylor, H. Chen, F. Qin, and C. Stewart. Avis: In-situ model checking for unmanned aerial vehicles. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 471–483. IEEE, 2021.

[95] E. C. Tetila, B. B. Machado, G. Astolfi, N. A. d. S. Belete, W. P. Amorim, A. R. Roel, and H. Pistori. Detection and classification of soybean pests using deep learning with UAV images. *Computers and Electronics in Agriculture*, 179:105836, 12 2020.

[96] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis. A Review on UAV-Based Applications for Precision Agriculture. *Information*, 10(11):349, 11 2019.

[97] R. van Renesse, R. Rodrigues, M. Spreitzer, C. Stewart, D. Terry, and F. Travostino. Challenges facing tomorrow's datacenter: Summary of the ladis workshop. In *Workshop on Large Scale Distributed Systems and Middleware*, 2008.

[98] A. Van't Hof and J. Nieh. Androne: Virtual drone computing in the cloud. In *Proceedings of the Fourteenth EuroSys Conference 2019*, pages 1–16, 2019.

[99] B. Varghese, N. Wang, D. Bermbach, C.-H. Hong, E. D. Lara, W. Shi, and C. Stewart. A survey on edge performance benchmarking. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.

[100] Z. Wang, X. Liu, A. Zhang, C. Stewart, X. Zhu, and T. Kelly. Autoparam: Automated control of application-level performance in virtualized server environments. In *Feedback Control Implementation and Design in Computing Systems and Networks*, 2007.

[101] Z. Xu, N. Deng, C. Stewart, and X. Wang. Cadre: Carbon-aware data replication for geo-diverse services. In *IEEE International Conference on Autonomic Computing*, 2015.

[102] Z. Xu, C. Stewart, N. Deng, and X. Wang. Blending on-demand and spot instances to lower costs for in-memory storage. In *IEEE International Conference on Computer Communications*, 2016.

[103] Z. Xu, C. Stewart, and J. Huang. Elastic, geo-distributed raft. In *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, pages 1–9. IEEE, 2019.

[104] Z. Xu, C. Stewart, and J. Huang. Rafting over on geo-diverse spot markets. In *IEEE Workshop on Data Center Performance*, 2019.

[105] M.-D. Yang, J. G. Boubin, H. P. Tsai, H.-H. Tseng, Y.-C. Hsu, and C. C. Stewart. Adaptive autonomous uav scouting for rice lodging assessment using edge computing with deep learning edanet. *Computers and Electronics in Agriculture*, 179:105817, 2020.

[106] W. Yang, W. Xu, C. Wu, B. Zhu, P. Chen, L. Zhang, and Y. Lan. Cotton hail disaster classification based on drone multispectral images at the flowering and boll stage. *Computers and Electronics in Agriculture*, 180:105866, 1 2021.

[107] N. Zhang, M. Wang, and N. Wang. Precision agriculture—a worldwide overview. *Computers and Electronics in Agriculture*, 36(2-3):113–132, 11 2002.

[108] S. Zhang and C. Stewart. Computational thinking curriculum for unmanned aerial systems. In *IEEE National Aerospace and Electronics Conference*, 2019.

[109] X. Zhang, L. Han, Y. Dong, Y. Shi, W. Huang, L. Han, P. González-Moreno, H. Ma, H. Ye, and T. Sobeih. A Deep Learning-Based Approach for Automated Yellow Rust Disease Detection from High-Resolution Hyperspectral UAV Images. *Remote Sensing*, 11(13):1554, 6 2019.

[110] Z. Zhang, J. Boubin, C. Stewart, and S. Khanal. Whole-Field Reinforcement Learning: A Fully Autonomous Aerial Scouting Method for Precision Agriculture. *Sensors*, 20(22):6585, 11 2020.

[111] Z. Zhang, J. Boubin, C. Stewart, and S. Khanal. Whole-field reinforcement learning: A fully autonomous aerial scouting method for precision agriculture. *Sensors*, 20(22):6585, 2020.

[112] Z. Zhang, S. Khanal, A. Raudenbush, K. Tilmon, and C. Stewart. Assessing the efficacy of machine learning techniques to characterize soybean defoliation from unmanned aerial vehicles. *Computers and Electronics in Agriculture*, 193:106682, 2022.

[113] Z. Zhang, S. Khanal, A. Raudenbush, K. Tilmon, and C. Stewart. Assessing the efficacy of machine learning techniques to characterize soybean defoliation from unmanned aerial vehicles. *Computers and Electronics in Agriculture*, 193:106682, 2 2022.

[114] Z. Zhou, Y. Majeed, G. Diverres Naranjo, and E. M. Gambacorta. Assessment for crop water stress with infrared thermal imagery in precision agriculture: A review and future prospects for deep learning applications. *Computers and Electronics in Agriculture*, 182:106019, 3 2021.