NASA/TM–20240011454

SupportU

Smart UAS Program for the Population by Offering Resources and Tools to the Unhoused

** All authors contributed equally to this memo. The names are listed alphabetically by last name.*

Samuel Beard
Leigh High School, California

Anjali Sinha
Washington High School, California

Beckett Song
Woodside Priory School, California

Anishka Vora
Dublin High School, California

Kevin Zhou
New England Innovation Academy, Massachusetts

September 2024

NASA STI Program Report Series

The NASA STI Program collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- Help desk contact information:

https://www.sti.nasa.gov/sti-contact-form/
and select the "General" help request type.

NASA/TM–20240011454

SupportU

Smart UAS Program for the Population by Offering Resources and Tools to the Unhoused

** All authors contributed equally to this memo. The names are listed alphabetically by last name.*

Samuel Beard
Leigh High School, California

Anjali Sinha
Washington High School, California

Beckett Song
Woodside Priory School, California

Anishka Vora
Dublin High School, California

Kevin Zhou
New England Innovation Academy, Massachusetts

National Aeronautics and
Space Administration

Ames Research Center
Mountain View, CA

September 2024

Acknowledgments

The team would like to acknowledge and thank David Thipphavong and Swati Saxena for their guidance throughout the entirety of the project. Another special thanks to Dr. Marcus Johnson and Pranav Kolluri for their help in understanding the complexities of UA design and assisting with the software implementations. Many thanks to Shannon Zelinski for organizing this experience.

This report is available in electronic form at

http://sti.nasa.gov

Table of Contents

Abstract

Global warming, challenging economic conditions, the opioid epidemic, and other widespread problems, have impacted people globally, particularly over the past several years. Preventable diseases like the common cold and the effects of heat stroke have become increasingly prevalent due to these issues.

It is estimated that 150 million people of the world's population are unhoused globally [1], with many dwelling in unsafe and unsanitary conditions while lacking access to basic hygienic items and other essentials. Unsanitary conditions coupled with this lack of access exacerbates and prolongs health problems and harms quality of life. Traditional methods of aid, such as homeless shelters and meal programs, face numerous challenges such as having limited reach and resources.

To address these problems, the Smart UAS Program for the Population by Offering Resources and Tools to the Unhoused (SUPPORT U) utilizes Uncrewed Aircraft Systems (UAS) to deliver resources to the unhoused and those in need of basic aid, prior to and during extreme temperature conditions, and after natural disasters in rural, suburban, and urban areas.

The UA is envisioned to be an autonomous aircraft capable of efficiently distributing essential supplies, including blankets, water, food, and medicine. The UAS fleet relies on advanced navigation and communication technologies to accurately identify unhoused people and efficiently and safely distribute materials to them. This will be done through a machine learning algorithm. By focusing on identified "homeless clusters", places where unhoused individuals are concentrated, the UAS network increases access to critical resources, thereby helping to mitigate some of the external risks to the health of unhoused individuals. The UA can also be used during crises, such as by transporting supplies to medical tents that are stationed in difficult-to-reach areas suffering from natural disasters.

Introduction

Background

Homelessness has been a pervasive issue in the United States, and has only worsened in the past few years. The unhoused population in the United States has been steadily increasing since 2017 and reached record highs in 2022, with the U.S. Census Bureau measuring approximately 528,000 unhoused individuals on one night in 2022 [2]. California especially struggles with homelessness, due to rising costs of living. California has 28% of the unhoused population in the United States [3], despite only having 12% of the nation's total population [4]. In fact, San Jose, a city in California, has nearly 6,266 unhoused individuals, and over 68% have been unhoused for over a year. On top of that, of those individuals, 23% do not have access or do not use any services the city provides [5].

In the United States, it is estimated that in 2018, only 65% of unhoused individuals lived in homeless shelters, while the remaining people lived on the streets or in other unsheltered areas [4]. These individuals often do not have access to proper sanitation or protection from harsh weather. This problem is exacerbated by climate change that has caused more extreme weather conditions such as severe cold or heat waves. Unhoused individuals not living in shelters are at higher risk during these conditions, with an estimated 700 unhoused deaths from hypothermia annually [6] and an additional 750 deaths from the heat each year [7]. Furthermore, unhoused individuals who reside in crowded areas such as encampments have greater exposure to harmful pathogens, which increases the risk of contracting tuberculosis, dysentery, and other illnesses. A lack of hygiene facilities can turn minor wounds into severe ones and amplify the spread of pathogens. People with disabilities are more likely to contract infections, such as sacral pressure injuries, which frequently happen to wheelchair users when they do not have a place to lay down. Diabetes and other associated illnesses that are not well controlled can potentially lead to infections [4]. Furthermore, as climate change continues to increase the frequency and severity of extreme weather events, the death toll will likely also grow. Exacerbating the dire situation even more, despite the rising unhoused population, the number of volunteers at homeless shelters continues to fall (see Figure 1), and shelters are struggling to keep up with the increasing demand for aid [8].

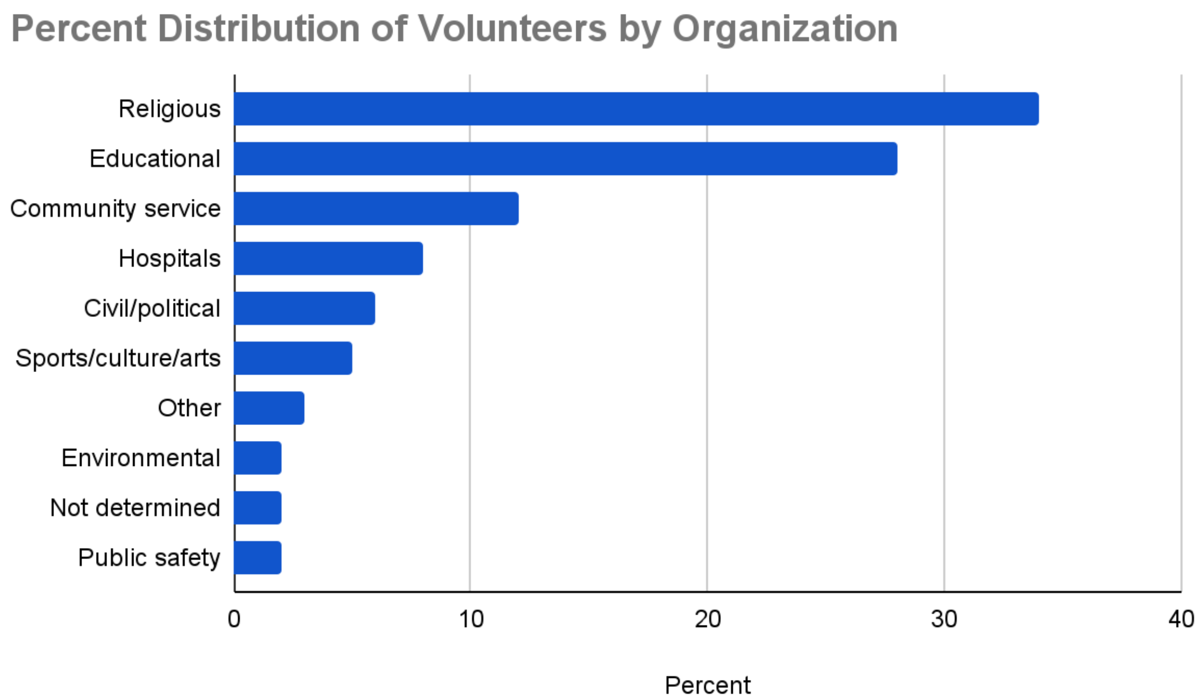


Figure 1. *Percent Distribution of Volunteers by Organization [8]*

Solution

The Smart UAS Program for the Population by Offering Resources and Tools to the Unhoused (SUPPORT U) concept is proposed to deliver hygiene products and weather relief items (water, blankets, ice packs, hand warmers, and more) to unhoused individuals or groups in rural, suburban, and urban environments. The UA operates autonomously using onboard artificial intelligence (AI), sensors, and pathfinding algorithms to traverse its environment, identify unhoused individuals, and deliver relief packages as efficiently as possible.

SUPPORT U Overview

In the United States, prior to sending out the UA, an operator must develop and submit a conflict-free operational plan to a UAS Service Supplier (USS) to ensure that all operations abide by UAS requirements [11]. By having contact with a USS, the UA will also have access to an air-traffic system for low-altitude UAS operations [12]. This ensures that all operations are safe and can be easily integrated into an ecosystem of existing low-flying UAS operations.

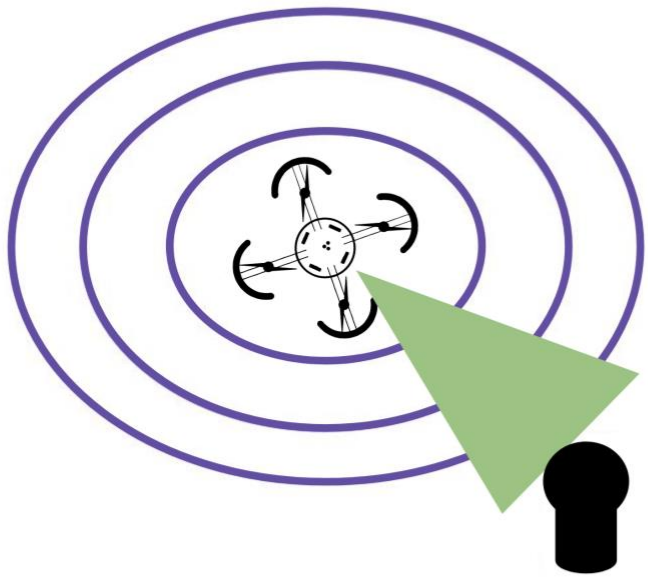


Figure 2. *UA identification utilizing AI model.*

The UA will use Dijkstra’s algorithm to traverse around its intended area of operations to locate the unhoused or others in need of the aid it is distributing. These UAs will fly about 40 feet above ground level, which is low enough for the system to identify unhoused individuals, while also being high enough to avoid traffic lights, power lines, and trees. In the initial operations stage, the UA will operate in cities like San Jose, where there are enough unhoused individuals for the UA to collect enough data to effectively train an AI model to identify unhoused individuals and refine the UA’s capabilities (see Figure 2). Later, once training of the object identification algorithm is complete and the UAS’ functions are refined, the UA can be deployed to larger cities with a larger unhoused population such as Los Angeles.

The UA will identify an unhoused individual using a modified and trained You Only Look Once, Version 3 (YOLO V3) AI model. This specific machine learning model is fast and accurate due to its usage of a simple neural network structure. The neural network structure consists of multiple layers of interconnected nodes, with positive weights exciting them. To train the model, an aerial image dataset of unhoused populations is required. Volunteer operators will assist in detecting unhoused individuals, and when a volunteer approves package delivery, the UA will collect the image data for training. Once the UA is trained, a neural network detects a member of the unhoused population, causing hardware components to be alerted. The UA will then descend, hovering about 10 feet above ground level.

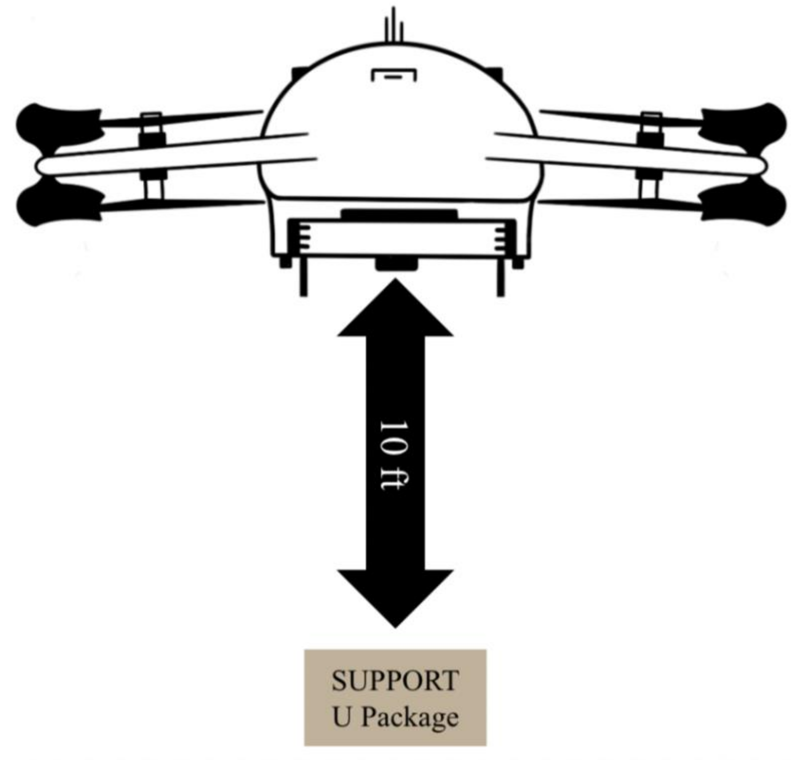


Figure 3. *UA distribution.*

Prior to distribution of aid, the UA will make a warning sound to inform people that a package is about to be delivered. It will state a message along the lines of “WARNING: FALLING PACKAGE.” It will then distribute the package that is filled with the items of need, which include, but are not limited to, food, water, and hygienic materials (see Figure 3).

After delivery, the UA will return to scouting the geography for other unhoused individuals. If necessary, it will return to the center of operations to be charged. The general rule is that the UA will return to the center of operations once the battery has discharged to within a stated threshold, adjusted with an extra five percent buffer.

Methods

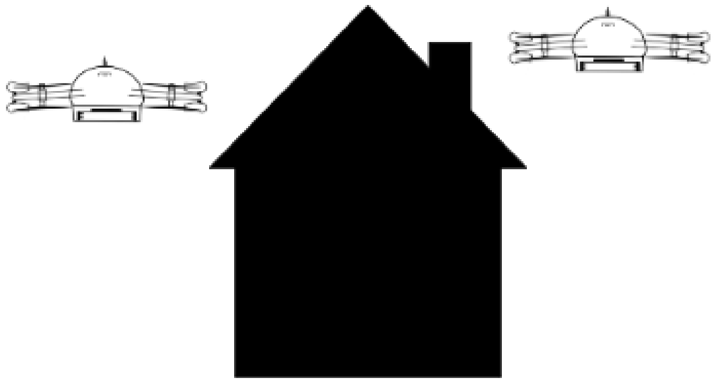
Implementation Roadmap

The implementation will occur in three phases described below and summarized in Table 1.

Phase One: Initial Operations

Initial operations will be a pilot program in a less dense area with low-tempo operations to collect data and learn. In this initial stage, the UA must map out the flight path that it plans to operate on. This includes paths around buildings and potential obstacles such as power lines and cranes that should be avoided. Although each UA is equipped with obstacle detection software, it is necessary to identify obstacles that may obstruct the UA’s path.

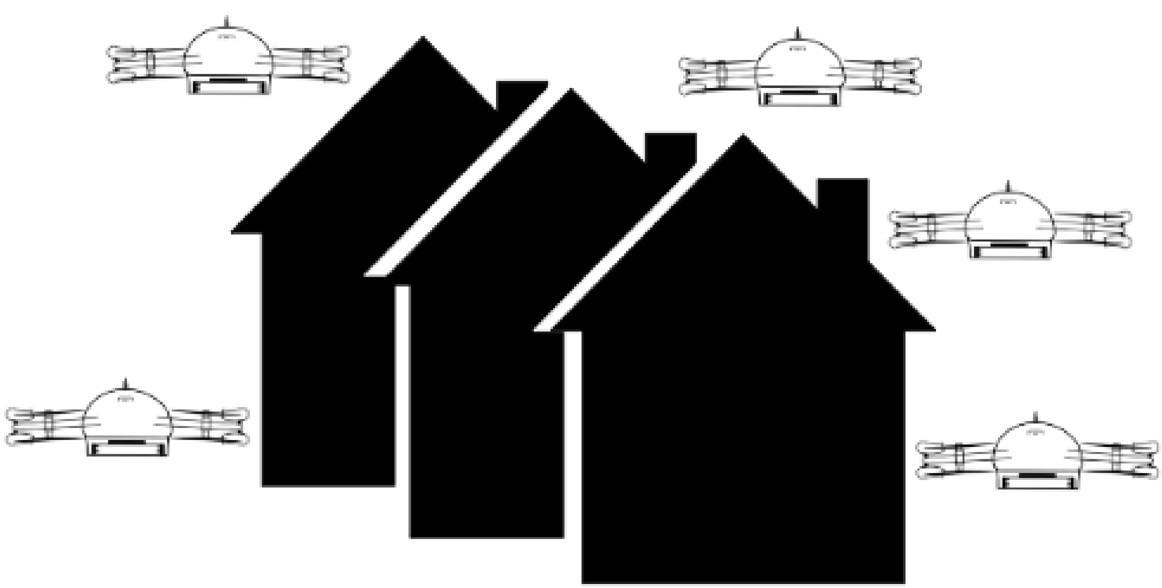
The UA fleet will be relatively small, consisting of fewer than five UAs. The geography in which the UAs will be deployed must be a suburban area with a low population density, such as the suburbs of San Jose. During this initial stage, the UAs will always be supervised and operated by volunteers at the center of operations. Volunteers will assist with UA navigation and confirm the UA's identification of unhoused individuals. Once the UA learns where clusters of the unhoused are, it will be able to navigate autonomously.



In order to train the machine learning model to identify members of the unhoused population, a dataset of overhead images of them is necessary. Since such a dataset does not yet exist, a custom dataset will be constructed over the first two phases of implementation as volunteers confirm identifications of members of the unhoused population. The image dataset will then be used to improve the machine learning model until the UA can correctly and reliably identify a member of the unhoused population on its own without a volunteer.

Phase Two: Intermediate Operations

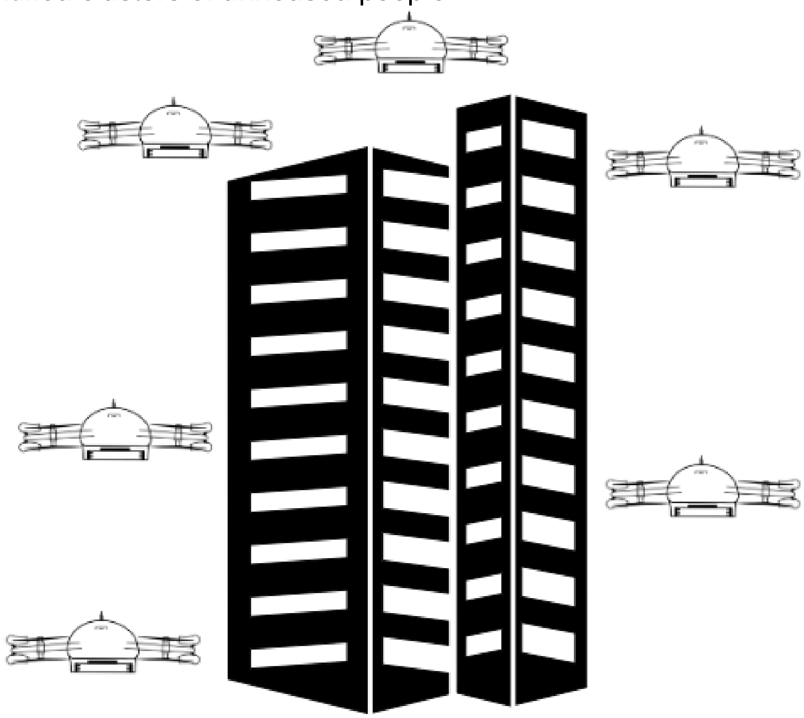
In the second stage of operations, the UA will have identified the area of operations and clusters of unhoused people as it distributes packages to these locations more frequently. As a result, UAs will autonomously fly to these locations without the guidance of volunteers. At this stage, it is expected that the UA will still require human approval to confirm the identification of an unhoused person and deliver a package. The operations in phase two will continue to grow the image database and the machine learning model to identify members of the unhoused population will be re-trained with the additional images included.



During intermediate operations, the fleet size will be scaled up to roughly 15 UAs. The UAs will be used in urban areas with medium population density, such as the outskirts of downtown San Jose.

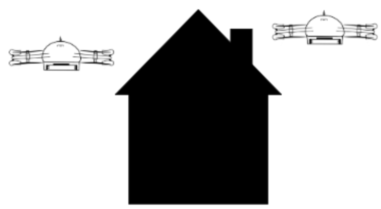
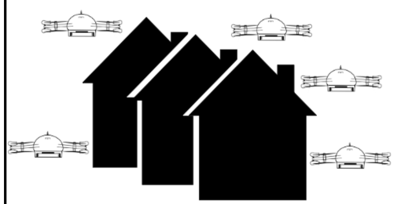
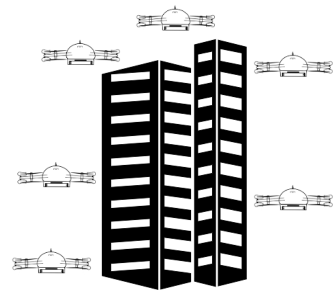
Phase Three: Advanced Operations

In the third phase of implementation, the image dataset will be of sufficient size for the machine learning algorithm to autonomously detect unhoused individuals accurately and reliably. In this stage, the UA will not only directly fly to the identified locations where there is a high concentration of unhoused people, it will also patrol the city to identify unhoused individuals in need throughout the city. This will allow the UA to reach and assist more people outside of the already identified clusters of unhoused people.



The fleet size during this stage will be large, consisting of over 45 UAs. They will also be used in the largest cities, where homelessness is prevalent. This includes cities such as San Francisco, Los Angeles, and New York.

Table 1. *Summary of Phased Operations.*

Phase One: Initial Operations	Phase 2: Intermediate Operations	Phase 3: Advanced Operations
		
Suburban Areas	Small Cities	Large Cities
< 5 UA fleet	~ 15 UA fleet	> 45 UA fleet
Semi-autonomous. Human operated and supported.	Semi-autonomous functioning, autonomous flight. Human approved package drop.	Completely autonomous UA fleet.

Hardware and Design

UA Frame: Overview

The UA's body follows a multirotor X8 configuration, consisting of four protruding arms in an X shape with one motor below and above each arm. The UA frame consists of a thick carbon fiber base plate topped by an aerodynamic yet durable and weatherproof curved canopy that houses the UA's electronics. From one end of the propeller to the opposite end, the UA is exactly 4.24 feet long (see Figure 4). Each propeller is one foot in diameter.

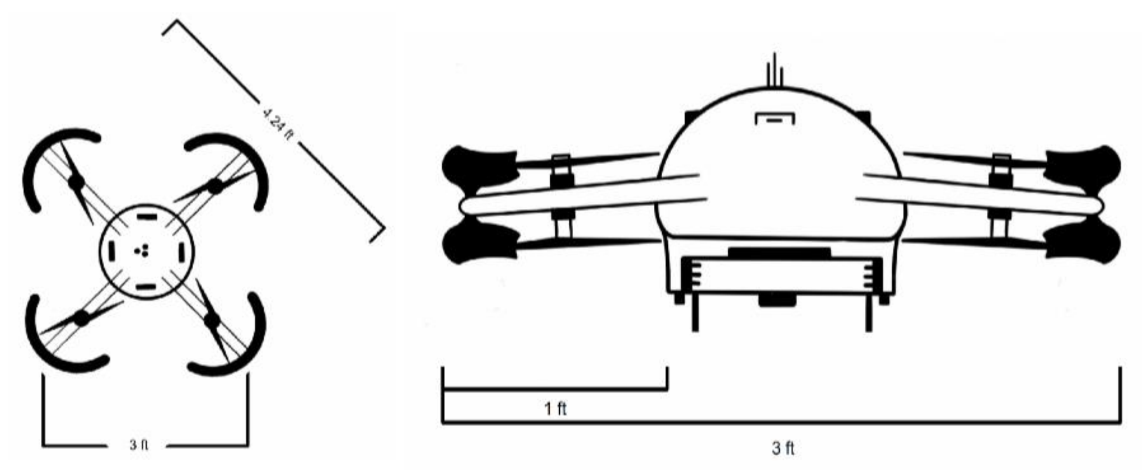


Figure 4. *Annotated Aerial and Front Facing View of UAS.*

The UA’s X8 configuration has greater mobility and maneuverability when compared to fixed wing aircraft. For example, the UA’s configuration allows it to take off and land vertically as well as hover, making it capable of traversing and navigating through dense cities. Additionally, an X8 configuration is more efficient than typical octocopters as having fewer arms results in less drag during flight. The UA frame should cost between $150 and $250.

UA Frame: Safety Features

The UA’s X8 configuration also serves as a safety feature. Extra motors means that if one motor were to fail, the UA would still be able to fly with its remaining seven motors, only failing in the unlikely case that two motors on the same arm fail at the same time.

Another safety feature incorporated into the UA frame is the use of propeller guards. At the end of each arm is a half-circle duct around the propeller. In the event that the UA collides with an object during flight, the propeller guard will protect people, trees, and power lines from the rapidly spinning and sharp propellers. A half-circle duct is sufficient protection, and having it only be a half-circle would prevent the UA from gaining substantial and unnecessary weight to the UA. Additionally, each duct around every pair of propellers houses two cameras for obstacle detection and avoidance. Based on the images provided by these cameras, the UA’’s software will guide it to avoid obstacles not anticipated by Dijkstra’s algorithm, the UA’s primary pathfinding algorithm, that maps the initial flight path.

UA Electronics: Overview

The UA’s curved canopy sits above the carbon fiber frame and houses essential electronics used to operate the UA such as the UA’s flight controller, power distribution boards, telemetry receiver and transmitter, and battery.

UA Electronics: Flight Controller

The UA’s Pixhawk® 6X flight controller [13] is responsible for using data collected by its key sensors to determine the UA’s orientation and desired speed for each motor. It accomplishes this by constantly running a PID (Proportional, Integral, Derivative) loop, an algorithm that calculates the required speed of each motor needed to achieve a desired motion, allowing the UA’s motors to function cooperatively to achieve stable flight even while performing movements such as turning. The cost of the Pixhawk® 6X controller is $270.

The Pixhawk® 6X flight controller is used with Ardupilot, an open source UA flight controller software that supports a variety of vehicle configurations, including multirotors. Ardupilot requires the flight controller running it to have a flash size of at least one megabyte, meaning that it can run on our chosen flight controller, which has an H7 processor. Ardupilot was chosen to be the UA’s flight controller software because it has various features that are useful for our use case. Specifically, Ardupilot supports autonomous waypoint missions, supports a wide variety of sensors, assists with autonomous vehicle collision prevention using Dijkstra’s, can use a non-GPS based navigation system, and supports multi-vehicle control [14].

UA Electronics: Power Distribution Boards (PDBs) and Electronic Speed Controllers (ESCs)

The UA requires two power distribution boards, which are both responsible for distributing the battery’s power to the UA’s eight electronic speed controllers, to manage the speed of the UA’s eight brushless motors. PDBs are typically designed to supply power to four motors, as standard multirotors only use four motors. Due to the UA’s X8 configuration, two PDBs are required to support the UA’s eight motors [15].

Additionally, the UA requires eight ESCs, as each ESC manages the speed of a single motor. The individual ESCs all work together for desynchronisation protection. This ensures that all motors are rotating in-sync and in the correct direction. Each ESC's job also includes protecting the motors from high voltages, current spikes, and extreme temperatures [16]. The ESCs can change their power output to help ensure that the UA flies flawlessly in various conditions while preserving motor health. Each ESC can cost between $75 and $120 while each PDB can cost between $50 and $150.

UA Electronics: Telemetry Communication System

A telemetry communication system will also be housed near the top of UA's canopy away from the rest of the UA's components to minimize interference from the antenna. Having a telemetry communication system allows the UA to receive and transmit information to and from the center of operations, allowing multiple UAs to be monitored from a central location. The receiver of the telemetry radio system housed at the center of operations receives the information transmitted by the telemetry transmitter on the UA, while the transmitter housed in the center of operations transmits information to be received by the telemetry receiver on the UA. This exchange of information between the center of operations and the UA allows for it to be remotely monitored [17]. Communication would happen through a three step process. Telemetry radios would relay information to cellular networks, which would then deliver it to the center of operations. By using multiple cellular networks, UA communication would have a 99.9% reliability rate [18].

The specific Telemetry Radio used by the UA is the SiK Telemetry Radio V3, which is compatible with the Ardupilot software that the UAS utilizes. Its range for communication is over 300 meters, making it perfect for UASs working in the city [17]. The 300 meter range would ensure the drone could give information to cellular networks. The SiK Telemetry Radio V3 costs around $65.

UA Electronics: Battery

The UA uses a high-capacity, high-discharge lithium ion battery to store its power. Compared to lithium polymer batteries, high-discharge lithium ion batteries are more energy dense for their weight and are safer because of their cylindrical metal enclosure. The high amp draw capabilities of lithium polymer batteries are not required as the UA's battery will not be subjected to high amp draw as the UA will fly and maneuver slowly, making lithium ion batteries the best choice because of the long flight times and increased safety they provide. In times of emergencies, like during heat waves or when the UA is utilized to deliver materials to natural disaster sites, the battery of the UA will be charged up to 90-100% [19]. In contrast, for regular flight deliveries, however, the UA's battery is expected to be recharged up to 80%, preventing battery degradation from repeated over-charging.

In general, charging lithium batteries past their nominal voltage gradually results in the growth of sharp metal tree-like structures called dendrites. If dendrites accumulate, they could pierce the metal shell of the lithium ion battery, causing short-circuiting or explosions. However, this is not a significant concern for lithium ion batteries as they use a hard metal shell that is harder for dendrites to pierce, while lithium polymer batteries' "pillow cells" are much easier for dendrites to puncture. Regardless, in normal operating conditions, not fully charging batteries is a safety precaution that helps prevent fires, explosions, and other worst-case events [20]. UA's lithium-ion batteries are interchangeable, increasing delivery efficiency during missions as a discharged battery can be quickly replaced with a charged one, which enables the UA to be redeployed quickly. These batteries can cost between $100 and $300.

Distribution Compartment and Mechanism Overview

The UA's distribution process requires two parts: the distribution mechanism and the distribution compartment. The distribution mechanism, responsible for pushing packages out of the UA, consists of two chain and sprocket designs that run along the underside of the UA's main body. The distribution compartment (see Figure 5) houses all of the distribution mechanism's parts. There are four steps to how the distribution compartment works.

Top View Side View

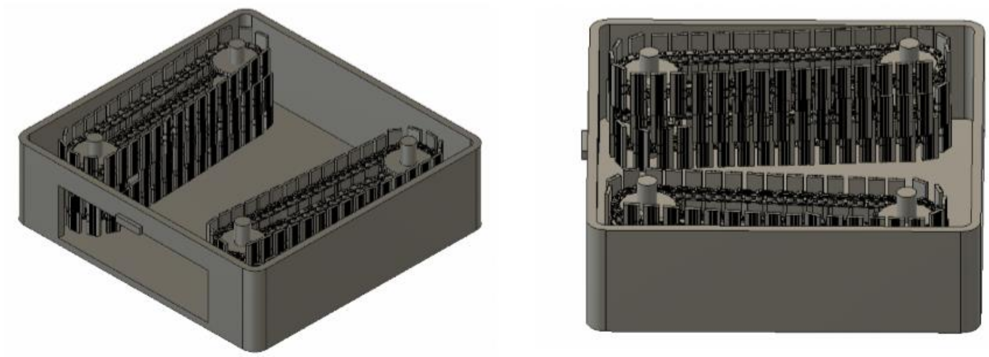


Figure 5. *CAD representations of the distribution compartment.*

First, the UA needs to correctly identify an unhoused individual. It will do so using the OpenCV video input and YOLO V3 object identification model (more details in later sections). As the UA lowers and positions itself in proximity to the unhoused individual, a signal will be sent out to the dispensing compartment to activate the dispensing mechanism to distribute a package.

Second, the UA activates the dispensing compartment. It begins with the UA's onboard audio system that consists of a small audio speaker found on the underside of the dispensing compartment. It announces the pending delivery of a care package to prevent anyone from being hit by a falling package.

Third, the dispensing mechanism, inside of the dispensing compartment, activates on the underside of the UA. As the sprockets under the UA body begin to rotate, a package in the middle of the two chains will be pushed forward, until it is dropped from the front slit of the UA.

Fourth, a beam sensor positioned at the top and bottom of the slit in which the package is dispensed informs the dispensing mechanism to stop. When the sensor realizes that the package has been distributed, it will inform the dispensing mechanism to stop pushing packages forward and deactivate the dispensing compartment until its next use.

Distribution Compartment Design Overview

There will be multiple versions of the distribution compartments. The different distribution compartments' sizes are optimized with regard to an object's weight and size. For example, packages containing trail mix are far lighter than packages containing water bottles; however, packages containing trail mix take up more space than packages containing water bottles. So, to decrease the UA's weight and improve efficiency, the water bottle distribution compartments would be smaller than that of trail mix. Water bottles are smaller and do not require a large distribution compartment, reducing the weight of the compartment if a separate one for it is made. If enough weight can be removed from the compartment and there is still space remaining, more water bottles can be loaded into the compartment than initially expected.

The distribution compartments are mounted to the underside of the UA utilizing a sliding cover mechanism. After the component has been slid into place, bolts are used as reinforcement, allowing the distribution compartments to be easily interchangeable with one another. Having interchangeable distribution mechanisms also reduces the costs of having to design different UAs to deliver different types of relief packages. Having multiple distributing compartments can also be helpful when, in certain weather conditions, higher priority items need to be distributed. For example, during a heatwave, UAs can utilize compartments that efficiently deliver water, while during colder weather events, UAs can be outfitted with compartments that deliver more blankets than water.

Distribution Mechanism Design Analysis

Our team also considered the use of a claw-based distribution mechanism to lower packages to the ground to prevent harm to people and property that can be caused by falling packages. However, having objects dangling from the UA may negatively impact the UA's stability, as small movements become amplified by the swinging of the claw and package, causing the UA to be less precise in its deliveries. If an excess amount of pressure were to be put on the claw's string, the UA may fall. Additionally, the claw mechanism would weigh more than a dropping mechanism, limiting the amount of packages that can be held, making the former a less practical choice for the SUPPORT U's mission.

The next major issue that our team ran into was designing mechanisms that would minimize unintended harm from the distribution of packages. To minimize the risk of harm induced by falling packages while still using a dropping mechanism as opposed to a claw mechanism, the UA is equipped with a speaker that will play a warning message before a package is dropped. Each package delivered is also equipped with a parachute to slow its fall.

The UA's distribution mechanism, which consists of two chain and sprocket designs, each housing three chains and six sprockets, would cost approximately $1,280 (with each sprocket costing ~$34 and each chain costing ~$92).

Distribution Compartment: Through Beam Sensor

Once a package is dropped, it passes a through beam sensor. Once the sensor realizes the package has fallen, it will inform the sprockets to stop rotating. The through beam sensors are photoelectric sensors [21]. They work through two parts: a sensor and an emitter. When the emitter sends out a light signal, the receiver processes it. Then, when something gets in the way of the laser, the sensor knows that an object is passing by [22].

In addition to the through beam sensor, two other photoelectric sensors—retroreflective and diffused sensors—were also considered. Retroreflective sensors house both the emitter and receiver in one part of the sensor. A reflector is then used to reflect the beam from the emitter back to the receiver. This is used for reflective, clear, or large objects [23]. Diffused sensors are similar to retroreflective sensors, as one object houses the emitter and receiver. In diffused sensors, though, the object it is detecting reflects the laser back at it instead of needing another reflective object as part of the sensor [24]. This is used for large, clear, and fast objects [25].

The through beam sensor was selected as the optimal sensor for the UA because it is most helpful in precise position sensing and because they are commonly used for small, opaque objects. Retroreflective sensors are best for large packages, and diffused sensors may not be able to sense objects if the object is curvy and shiny (like the plastic packaging) [26]. All these sensors may have potential drawbacks when used on sunny days, as different forms of light may interfere with the laser's abilities. Overall though, through beam sensors seemed the most logical for this application. The price of through beam sensors range from $200 to $430.

Distribution Compartment: Audio System

The UA uses a small audio speaker to warn people that a package will be released. It will state a message along the lines of "WARNING: FALLING PACKAGE", and then will emit repeating beeps as the package falls. The specific audio speaker that is used by the UA is the SP-1605, a very small but loud speaker. This speaker costs only $2 and is lightweight [27], bringing the total cost of the entire compartment to between $1,482 and $1,712.

Software

Path-finding

In order for the UA to successfully navigate around the city, navigation/pathfinding software along with obstacle detecting cameras must be utilized. This section will explore various algorithms that can be applied to the UA and determine which one is the most suitable for our application. Pathfinding systems including Dijkstra's algorithm, A* (A-Star) Search Algorithm, and Depth First Search algorithm were assessed.

Pathfinding software is necessary because the UA is intended to be fully autonomous by the final stage of implementation (phase 3: advanced operations). In general, pathfinding algorithms utilize graphs containing nodes and discover the most favorable path between two specified points. In order for these algorithms to be usable in the real world, a navigation mesh must be created using maps. Commonly used in video games, a navigation mesh uses polygon nodes (cells) that are connected to each other using edges (see Figure 6). Cells contain information, which can be used to make decisions to avoid the cells. In our case, predetermined obstacles, such as buildings, will be cells that are labeled toxic. These toxic cells will be heavily weighted, so the algorithm considers them as very costly to traverse, effectively avoiding them [28].

The mesh may be based off maps created by UAs or by satellite systems. In the early stages of implementation, it will likely be more effective to use satellite maps as they are more easily available.

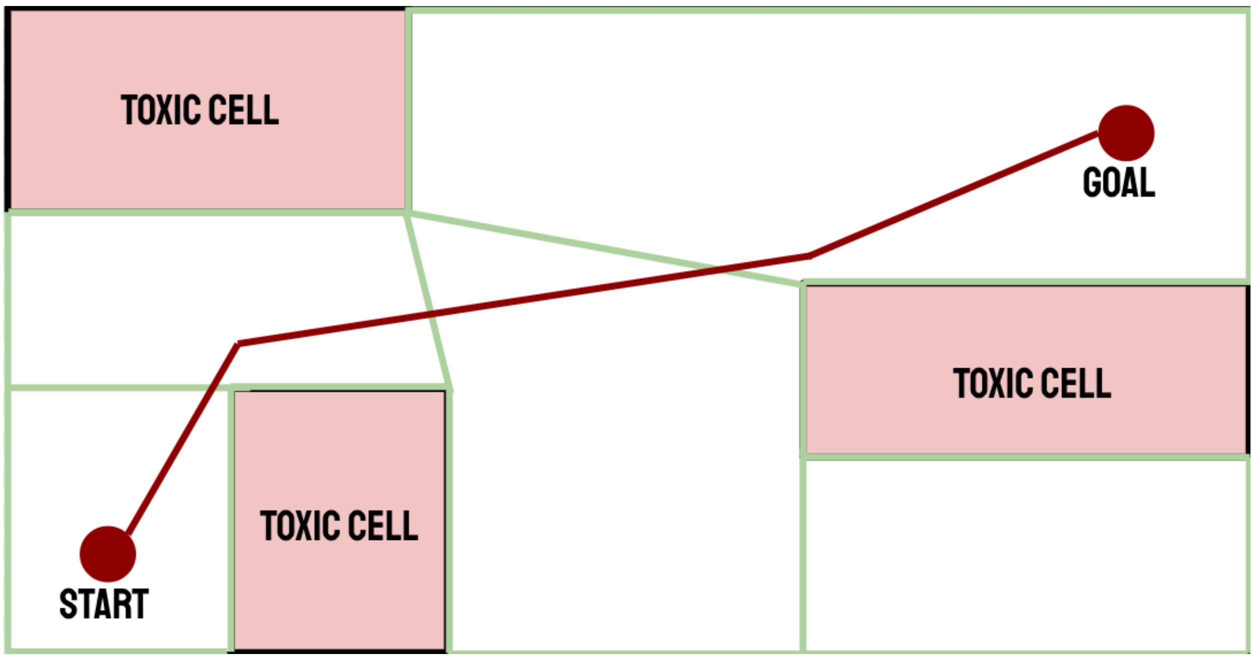


Figure 6. *Example Navigation Mesh with obstacles, cells, and edges.*

Dijkstra's Algorithm

Dijkstra’s algorithm functions by creating the shortest path tree from the source node to every other node in the graph. Beginning at the initial node, it analyzes the graph to find the shortest path between that node and all other nodes in the graph. The current shortest distance from each node to the source node is kept track of. The other node is designated as “visited” once the shortest path between it and the source node is discovered. Until each node in the graph has been added to the path, the process is repeated. Dijkstra's algorithm is especially suited for weighted graphs [29].

One important point about Dijkstra's algorithm is that there can be no negative weight edges within the graphs being analyzed since graphs can only be positively weighted. The shortest path to a node is indicated as the current path once it has been designated "visited." The overall weight should not be able to decrease following this step, which is why negative weights are not allowed [30].

The algorithm begins by creating a 'queue' of all nodes within the graph, an empty 'set' including all nodes the algorithm has visited, and an array of all tentative distances from the initial node to every other node. The only known distance in the beginning is from the initial node to the initial node (0), so all other distances are represented by infinity [31]. The queue begins full and the set begins empty, but when the algorithm ends, it's vice versa with the queue being empty and the set being full. The algorithm iterates through unvisited nodes with the smallest tentative distances from the initial node. Then, the adjacent nodes are visited, and the tentative distances of these nodes are updated if a shorter distance (or path) is found [32]. Figure 7 shows a visualization of Dijkstra's algorithm.

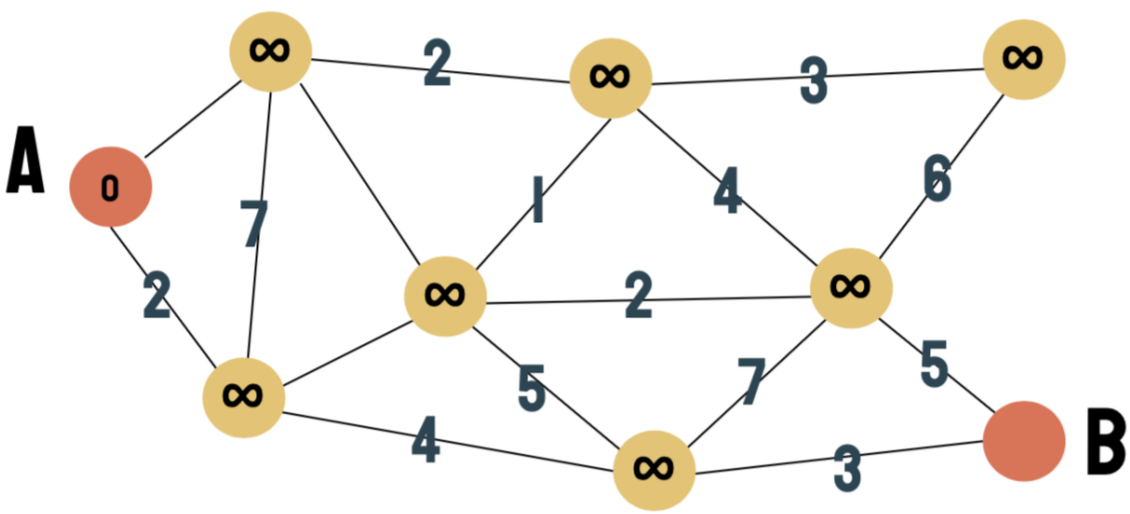


Figure 7. *Dijkstra's Algorithm Visualization, with numbers representing weights.*

A* (A-Star)

To operate, A* creates the lowest-cost path tree connecting the final node to the initial node. At every stage, A* algorithms can anticipate and make the optimal choice by using a heuristic. A* algorithms may sometimes not return the shortest path at all times. This is because it is heavily dependent on estimation and guessing via heuristics.

The heuristic function calculates the approximate total cost of a path from the given node to the final destination. In other words, it is an educated guess and is not always objectively accurate. This heuristic function called h(n), where n is the current node the algorithm is on, is one component of the function used to expand paths. The second part is g(n), which is the movement cost thus far to reach node n. Lastly, f(n) is the total approximate cost of a path to reach node n. Therefore, the entire function f(n) = g(n) + h(n) is utilized to pick the node with the lowest f(n) value. The function is repeated until the final node is reached [33].

There are two main ways the heuristic function is calculated. Whichever method's value of h(n) is closest to the real cost of getting to the end point is the most ideal heuristic method. A value of h(n) that is less than the real cost is more ideal than a value of which is greater. Greater h(n) values correlate to more speed but the optimal path will never be found because the heuristic believes the shortest path is longer than it actually is [34].

The first type of heuristic is known as the Manhattan Distance Heuristic. This heuristic only allows you to move in four directions: up, down, left, or right. Diagonal movements or any other type are not allowed. This can be effective in some cases as the heuristic will be exact if the real path only follows straight lines. This method may also be preferred when the user prefers a path consisting of several straight lines.

The second type of heuristic is Euclidean Distance Heuristic. Contrary to the Manhattan Distance, this heuristic can move in any direction, and cut through nodes in any manner. This heuristic may be more accurate, but it is much slower as it must inspect a larger area [35]. A diagram of an example A-Star algorithm is shown in Figure 8.

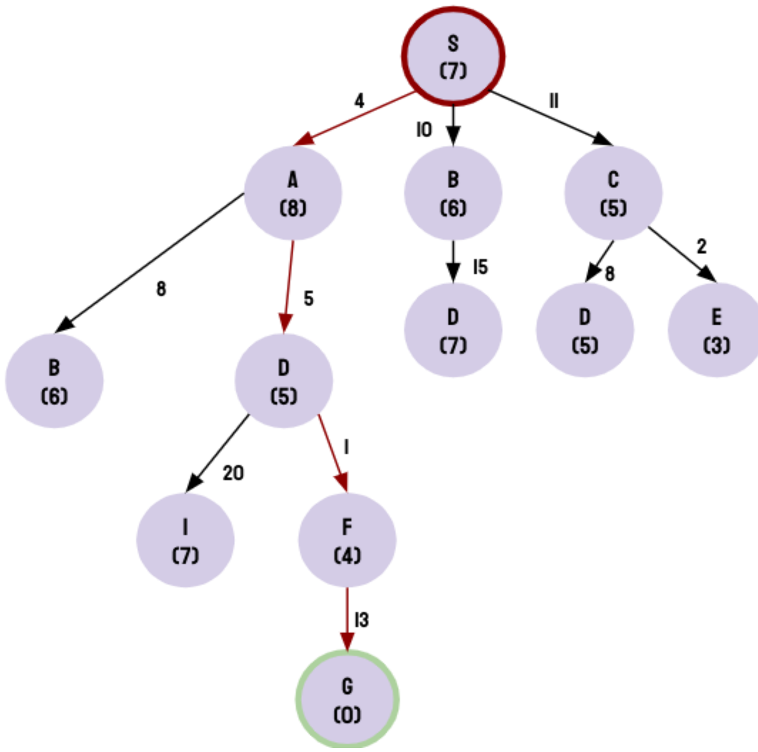


Figure 8. *A-Star algorithm diagram showing branches and trees.*

Depth First Search

Depth First Search (DFS) is a graph traversal algorithm which explores different nodes and edges within a graph. Beginning at the initial node, it explores each branch as far as it can get before backtracking to the original node. This strategy ensures that the algorithm goes over each node and edge exactly one time. Once the algorithm reaches a node with no unvisited adjacent nodes, the algorithm backtracks to the most recent node that has unvisited neighbors. Oftentimes, a stack is utilized to keep track of nodes to visit next as the algorithm iterates over all nodes [36].

DFS works the best within applications requiring all possible paths to be found within a graph. These real-world scenarios include mazes and puzzles where every solution must be found. In an urban landscape, however, DFS would be too exhaustive and not efficient [37].

Summary and Final decision

After analyzing all three pathfinding algorithms considered, Dijkstra's algorithm was chosen for this application due to its efficiency in terms of speed and accuracy. A* was not selected because the heuristic functions make it difficult to work with and it is not scalable in large applications. Further, since it relies heavily on estimations, it does not always yield the shortest path. DFS was not chosen because it may be too exhaustive in an urban landscape and, thus, not as efficient as Dijkstra's. On balance, despite that Dijkstra's algorithm is unable to work on graphs with negative weights, its relative accuracy and availability to quickly provide the shortest path makes it a good choice for the UA.

In order to ensure the efficiency and capability of Dijkstra's algorithm, a Python program was developed. To visualize the algorithm, the program outputs a map of the shortest path and a map of the shortest distances, respectively shown in Figure 9. For more information about the program, take a look at https://github.com/anjalis-onion/AIR_AID_2024. Currently, the program begins in the top left corner and attempts to find the shortest path to the bottom right corner, but this can be adjusted.

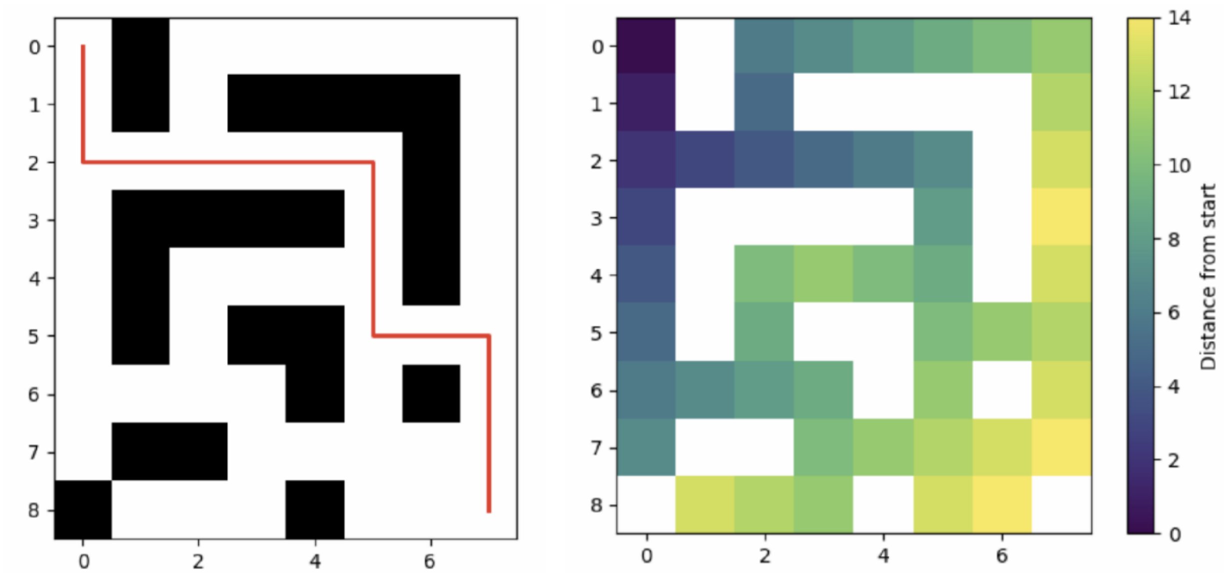


Figure 9. *Program visuals using matplotlib and numpy, left shows shortest path with black boxes resembling obstacles and right shows distance map from start.*

Table 2 summarizes the pros and cons of each pathfinding algorithm considered.

Table 2. Summarized Pros and Cons of the pathfinding algorithms considered.

	Dijkstra's	A*	DFS
Pros	• Very accurate and precise, will always find shortest path • Fast and good with time	• Using an admissible heuristic means that A* is guaranteed to find shortest path • Prioritizes specific nodes, so often times is faster	• Simple and easy to implement • Effective when all solutions must be known (maze, puzzle)
Cons	Unable to work on graphs with negative weights	• Heuristic function is hard to create • Algorithm can be ineffective without appropriate heuristic • Not easily scalable in large applications	• Does not guarantee shortest path • Lack of direction/purpose means long search time

Object Detection Overview

In order for the UA to accurately detect unhoused individuals within a specified geography, an accurate object detection system must be developed. Advancements in the field of AI have introduced several different prospective models and techniques, each having their own advantages and limitations. Three object classification models (You Only Look Once, Version 3 (YOLOV3), Faster Region-based Convolutional Neural Network which is an improved version of Region-based Convolutional Neural Network (R-CNN), and EfficientDet) were analyzed. The method of implementing the chosen model will also be covered.

Before these specific models are explained further, it is important to understand the general structure of object detection models. Each model utilizes a deep learning framework, called a neural network (see Figure 10), that extracts specific features and detects objects. Neural networks utilize nodes and layers, somewhat similar to a human brain. In general, the architecture of a neural network consists of an input layer (to process and categorize the data), one or multiple hidden layers (to perform further analysis of the data), and an output layer. Between each layer and the multiple nodes, many connections are created, represented by numerical weights. Positive weights excite connected nodes and demonstrate a connection to a specific class (dog, bicycle, truck). This causes nodes to pass along data to other nodes. If nodes do not get excited, they do not pass along any data [38].

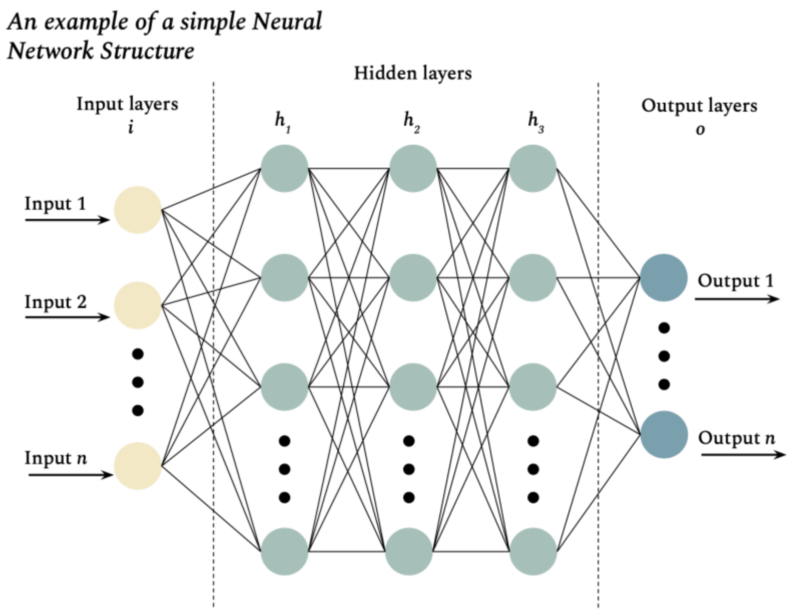


Figure 10. *Neural Network diagram showing layers, weights, and nodes.*

Object detection models can be trained using unsupervised learning, supervised learning, or a mix of both. In supervised learning, the model is trained on a dataset which consists of labeled data that has already been classified. These datasets are specially curated to get the machine learning models to classify objects and predict outcomes. By contrast, in unsupervised learning, the dataset used to train the model is not labeled. This prompts the algorithms to discover the underlying connections between the inputs and outputs. In unsupervised learning, the models are generally meant to perform functions including association, clustering, and dimensionality reduction [39].

YOLOv3

The YOLO family of models is known for their speed and accuracy. They are all single-stage deep learning algorithms. These types of object detectors have only one stage, unlike two-stage algorithms, which will be discussed later in the paper. These one-stage algorithms perform detection and image classification in one step, making them fast [40]. Single-stage deep learning algorithms take in an input image, extract specific features using a neural network, predict bounding boxes and class labels using regression analysis, and provide an output including a set of bounding boxes associated with certain class labels. In particular, YOLO will partition the image into a grid and anticipate the class probabilities and bounding boxes for every grid cell [41].

The process for single-stage algorithms includes a method known as feature extraction, best done by neural networks. It is especially important to understand how neural networks work, specifically the connections between nodes. Every new connection that a node receives will be assigned a number called a "weight." Over each of its connections, the node receives a unique data item, a different number, when the network is active, which it then multiplies by the corresponding weight. After that, it totals the output products to get a single figure. The node does not send any data to the following tier if that number is less than a certain threshold. The node activates if the value surpasses the threshold, which typically entails transmitting the value, the total of the weighted inputs, along all of its outgoing connections [42]. Nodes themselves can hold values, which can sometimes even be binary.

Single-stage algorithms also involve the prediction of bounding boxes and class labels (see Figure 11). Regression layers are used by the model to identify the bounding box coordinates and dimensions after relevant features have been extracted from the image. In order to do regression, the image is divided into a grid, with bounding boxes for items predicted in each grid cell. These boxes are formed by grouping the dimensions of actual items in the training data to detect common shapes and sizes. A confidence score is assigned to each bounding box, signifying the probability of its accurate representation of an object and its class. The model forecasts the class labels, which indicate the kind of object detected, for every bounding box. [43]. Single-stage algorithms combine object detection and image classification into a single step, allowing the input image to be fed into the network directly and producing the class probabilities and bounding box coordinates as the output [44].

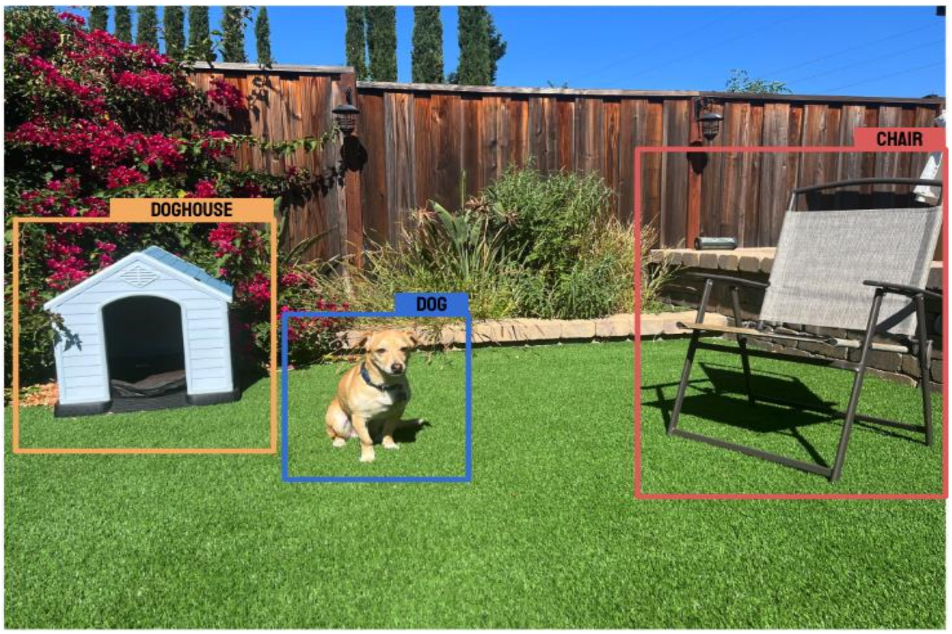


Figure 11. *Example of bounding boxes and YOLOv3 detection*

Faster R-CNN

Faster R-CNN (depicted in Figure 12) is the improved version of R-CNN. This model is similar to YOLOv3 in that it utilizes deep learning. Both of these models also share the use of anchor boxes and bounding boxes to handle different objects of unique shapes and sizes. Building off of this, both models predict bounding box coordinates utilizing regression layers. The key difference between the two models is the architecture, as faster R-CNN follows a two-stage approach. These two phases consist of generating region proposals using a Regional Proposal Network (RPN) and then classifying and detecting objects using the Convolutional Neural Network (CNN) 45].

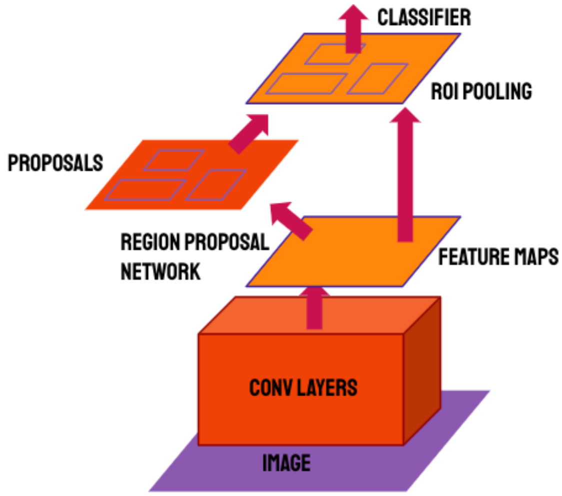


Figure 12. *Faster R-CNN structure*

RPN is a type of neural network that uses convolutional layers to anticipate the boundaries of objects. These potential boundaries are called the region proposals. The RPN will go over the image and create anchor boxes at each potential location. It will predict if the boxes contain objects and will then refine the coordinates of the bounding box using regression [46].

After proposed anchor boxes have been generated, Region of Interest (RoI) pooling occurs. This function standardizes the size of each of the proposed regions so that the CNN can take the regions in as an input. This standardization is done by dividing each region proposal into subregions and applying max pooling to each sub region. Max pooling will take the maximum value in a sub region. Once each maximum value is obtained, a fixed-size matrix of all maximum values will be produced. This allows all prominent features in the region to be retained, and the dimensions will now fit into the CNN [47].

Finally, the Fast R-CNN classifies the features within the bounding boxes created by the RPN and sized down by RoI pooling. Unlike YOLO's simple neural network, the CNN that it used is much more complex. This learning model also contains layers, but these layers include convolutional layers, pooling layers, and fully connected layers. The convolutional layers filter the input image and detect specific things like edges, textures, and patterns through performing convolutions. Next, although pooling has already been applied to the proposed region, more pooling must be done to the filtered image. The pooling functions generally downsample and select the maximum value in sub-regions, often using max pooling to do this. The fully connected layers pass this image through multiple different nodes, which learn to detect and classify [48]. The output may consist of the class scores of the image classification task. CNN's complexity means it takes longer to run and may not be effective in real world situations.

EfficientDet

Similar to how Faster R-CNN uses a convolutional neural network, EfficientDet, who's backbone is EfficientNet, also utilizes a CNN architecture. This CNN is specifically designed to balance the depth, width, and resolution of images that it classifies. Another key component of this model is the Bidirectional Feature Pyramid Network (BiFPN). This type of network allows for a very efficient multi-scale feature fusion, as it adds learnable weights to a variety of input features. By doing so, computation is somewhat reduced as the network structure is slightly simplified. After the weights are added, different features from different levels of the CNN are fused and combined together. This is done to enhance the ability to detect objects of various sizes and scales. A compound scaling method is also used to scale different components of the model, including the base CNN and the BiFPN. Anchor boxes are then created to predict the bounding boxes [49].

It is important to understand EfficientDet's main distinguishing factor from other CNN models, the BiFPN, which enables its scalability. BiFPN works by taking feature maps produced at different layers of a CNN and combining them. This means that information flows from higher resolution, more detailed layers to lower-resolution layers. BiFPN uses learnable weights to make this fusion as effective as possible, so the network is able to decide the importance of each feature map during the fusion process. The network can prioritize the most useful information, which makes it computationally efficient 50]. Figure 13 illustrates how different structures of EfficientDet can be scaled in different ways, such as width, depth, and resolution.

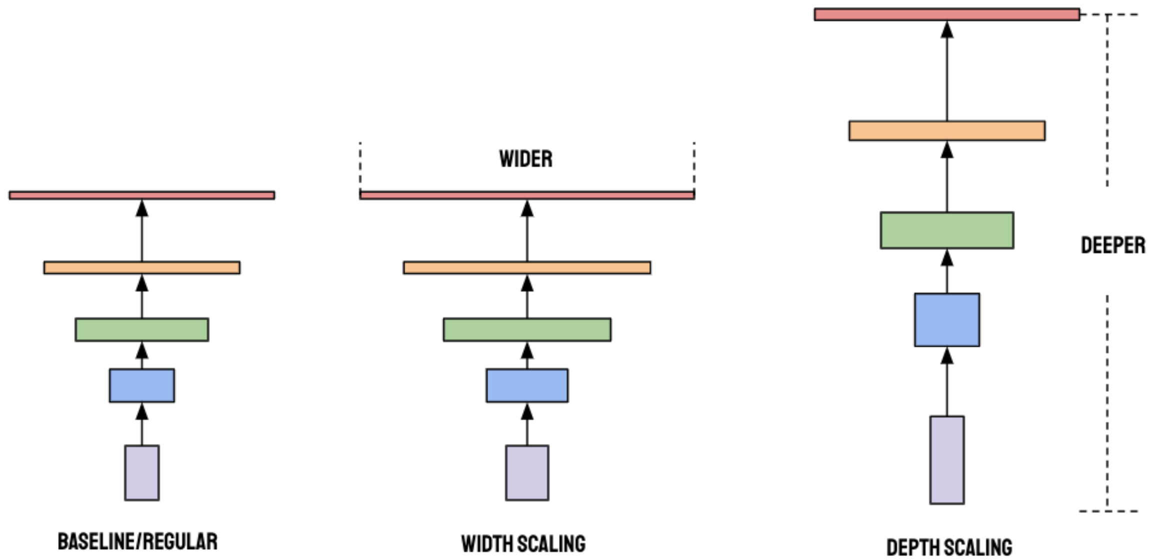


Figure 13. *Different structures of EfficientDet can be scaled in different ways*

Final Decision and Implementation

After analyzing all three models, it is evident that they each have their own specific use cases for which they would be the most effective. However, for the SUPPORT U system, the speed of the model is highly prioritized. The model needs to effectively work in a real time situation. Therefore, YOLOv3 was deemed to be the most suitable object detection software. It is extremely fast and simple due to its single neural network structure and use of a single-stage algorithm for both the detection and classification of objects. It is also very accurate. Table 3 summarizes the pros and cons of each Machine Learning model considered.

Table 3. *Summarized pros and cons of Machine Learning models*

	YOLOv3	EfficientDeT	Faster R-CNN
Pros	● Accurate and fast due to simple, single neural network structure Accurate and fast due to simple, single neural network structure ● Its simplicity makes it easy to train and implement	● Very accurate due to BiFPN (Bi-directional Feature Pyramid Network) ● Good application with large datasets	● Very flexible and customizable ● Uses two-stage approach for accuracy
Cons	● May not be as effective in complex cases	● Extremely complex and relatively long runtime	● Slower, requires lots of computing power ● Difficult to implement because of complexity

With regard to how the model is trained, a large image dataset of the unhoused population is required. It is important to have these images taken from an aerial perspective, as this is how the UA will be analyzing image data. Because such a dataset does not currently exist, these images need to be collected during the first two stages of implementation. The images in the dataset will be automatically labeled using software known as labelbox, which will also draw anchor boxes around the pictures. The model will be trained, with weights determined through a back propagation algorithm. This function will adjust the weights to minimize the error between the actual and predicted outputs [51]. On top of this, hyperparameters, which are parameters, such as batch size or learning rate, that are adjustable, will automatically be adjusted using tools like Keras for maximum efficiency. OpenCV will be utilized for the video input and analysis, and Raspberry Pi will be utilized for computing.

Analysis

Professional Feedback

During the design process, we met with and received feedback from two UAS professionals, one experienced in software and the other in hardware.

One of the creators of Unmanned aircraft system Traffic Management (UTM) gave us feedback and advice about UAS hardware. They noted that weather would be a major challenge for UA efficiency, as they would struggle in both hot and cold temperatures. They explained that cold temperatures would decrease battery life, as the chemicals inside batteries move slower and undergo less energy-releasing reactions. Hot temperatures would cause the air around the UAS to become less dense, requiring the motors to work harder to produce sufficient thrust. They also noted that wind speeds could also be an issue, as they can affect the flight and stability of the UAS. They advised us to implement a warning system before dropping the package, so that those nearby are aware of the falling object. Finally, they recommended that we gather a large and varied dataset of images to train the object identification AI, as there are no existing datasets of overhead images of unhoused individuals. If the AI is not provided with sufficient data, it would not be able to correctly identify unhoused individuals from other people or even other objects. Similarly, if our dataset is not diverse enough, the AI may not work in varied geographies or situations.

Additionally, an expert in image classification provided us feedback and advice based on their experience in UAS software. They gave us recommendations on processors to use to run the onboard AI, such as the NVIDIA Jetson or a Raspberry Pi, as these processors are small enough to fit on the UA while possessing the processing power to run the AI. They also warned us of how it may be difficult to prevent the AI from serving the same individual multiple times within the same time frame. This problem could occur if someone who had already received a package moves to a different location where the UA may distribute another package to them.

Potential Challenges

In the United States, one major issue inhibiting the potential for autonomous UAs are Federal Aviation Administration (FAA) laws and limitations. Other countries may also have similar rules and regulations governing UAS. In the United States, the most significant of these are the operational limitations preventing small, unmanned aircraft from flying beyond visual line of sight (BVLOS). Currently, FAA guidelines require “the unmanned aircraft must remain within VLOS of the remote pilot in command and the person manipulating the flight controls of the small UAS. Alternatively, the unmanned aircraft must remain within VLOS of the visual observer.” These guidelines also require “at all times the small, unmanned aircraft must remain close enough to the remote pilot in command and the person manipulating the flight controls of the small UAS for those people to be capable of seeing the aircraft with vision unaided by any device other than corrective lenses.” These rules prevent a truly autonomous UA system from being possible, as having a visual observer following around the UA nullifies any benefit of the UAs being autonomous in the first place. They also prevent these systems from being implemented in large cities or over rural areas, as the UAs can travel to areas further than the pilot would be able to observe them by eyesight. In order for the SUPPORT U concept to be implemented, these regulations would need to be changed to facilitate the use of fully autonomous UASs.

The largest potential challenge this operation may face is a lack of sufficient data for detecting unhoused people. Currently, there are no databases containing images of unhoused people that can be used to train an AI, so there is a need to create a large database of these images during initial operations. Acquiring sufficient amounts of high quality image data to develop an accurate model may require a substantial amount of time and disk storage, so it is important to make sure that the UAS can process that much information onboard. Also, the UA needs a way to determine if someone has already received a package to prevent the same person from receiving the same package more than once if they move.

On top of this, there are several privacy concerns regarding collecting images of people and monitoring people in general. It would not be feasible to get approval to watch all unhoused individuals from above, so there is a need to find a way to deliver these packages without people feeling privacy violations.

Future Implementations

The UA must be able to efficiently traverse the geography in off-optimal conditions where the UA can be most helpful, such as emergency situations. For example, in hot conditions, air is less dense, which requires the motors to work harder; in cold climates, the battery efficiency decreases. There is a need to develop solutions to ensure that the UA can work in the most critical of situations.

Conclusions

As the homelessness crisis continues to grow in the United States, more and more unhoused individuals are forced to live unsheltered [52] in unsafe and unsanitary conditions. Homeless shelters, transitional housing, and other facilities are often unable to provide adequate dwelling to this growing population, leaving many exposed to the elements and without access to basic necessities such as food, water, and hygiene products. In recent years, due to the negative impacts of climate change, the effects of these conditions on the unhoused population are expected to become more severe, leading to additional hardship from exposure and increased potential loss of life.

An autonomous UAS utilizing AI, sensors, and pathfinding algorithms to traverse 3D environments, can deliver relief items to unhoused individuals in need of but lacking access to such relief. Additionally, the UA is designed to deliver a variety of objects bearing different weight and size in an efficient and safe manner. Its eight powerful motors allow it to carry heavier payloads and navigate through dense urban areas effectively, while minimizing the risk and consequences of a failure. The UA's high capacity batteries provide a long flight time before needing to recharge. Further, the drop dispensing mechanism consisting of two chains and sprockets is lighter than a claw-based delivery mechanism and is simpler to implement. Prior to dispensing the aid, the UA will issue a warning to help minimize the harm from unanticipated delivery. Using advanced AI and pathfinding algorithms, the UA can be deployed in areas known to have a high concentration of unhoused individuals. In its initial phases of implementation, the UA will require operator control to gather aerial data for training software to detect unhoused individuals. Ultimately, the goal is to have the UA operate safely and efficiently in diverse geographies without any human operators.

Current limitations of the SUPPORT U concept include operation in inclement weather conditions, compliance with UAS laws, and the dearth of AI training data. Extreme weather conditions such as high winds or unusually severe temperatures can greatly impact the performance of UAs. Further, current FAA laws requiring UAS to be in the operator's visual line of sight at all times prevent the UA from operating as desired in this concept. Finally, training the AI to effectively identify unhoused individuals will require sufficient aerial images of unhoused individuals in various environments. If these hurdles can be overcome, UAs can positively impact the lives of many thousands of unhoused individuals by providing essential support and aid, especially in times of crisis such as severe temperatures and natural disasters.

References

[1] S. Fleming. “How to solve homelessness – lessons from around the world.” World Economic Forum. https://www.weforum.org/agenda/2019/12/how-to-solve-homelessness-poverty-cities-urbanization/ (accessed August 1, 2024).

[2] B. Glassman. “New survey data provides demographic profile of population experiencing homelessness who lived in emergency and transitional shelters.” United States Census Bureau. https://www.census.gov/library/stories/2024/02/living-in-shelters.html (accessed June 10, 2024).

[3] M. Kendall. “How big is California’s homelessness crisis? inside the massive, statewide effort to find out.” CalMatters. https://calmatters.org/housing/homelessness/2024/01/california-homeless-point-in-time-count-2024/ (accessed July 22, 2024).

[4] C. Y. Liu, S. J. Chai, and J. P. Watt, “Communicable disease among people experiencing homelessness in California: Epidemiology & infection,” Cambridge Core, https://www.cambridge.org/core/journals/epidemiology-and-infection/article/communicable-disease-among-people-experiencing-homelessness-in-california/01D82460F7E8092791D0C5B1B94C8343 (accessed July 24, 2024).

[5] "Homeless reports." City of San Jose. https://www.sanjoseca.gov/your-government/departments-offices/housing/resource-library/homeless-reports (accessed August 9, 2024).

[6] "National Coalition for the Homeless Calls for Warming Centers to be Opened in US Cities to Meet Demand!" National Coalition for the Homeless. https://nationalhomeless.org/tag/hypothermia/ (accessed July 22, 2024).

[7] K. Good. “The Disproportionate Impact of Climate Change on People Experiencing Homelessness.” Texas Homeless Network. https://www.thn.org/2024/04/03/the-disproportionate-impact-of-climate-change-on-people-experiencing-homelessness/ (accessed July 22, 2024).

[8] “Organizations for which volunteers work.” TED: The Economics Daily. https://www.bls.gov/opub/ted/2002/dec/wk5/art03.html (accessed July 24, 2024).

[9] [“Welcome to the best delivery experience not on earth.” Zipline. https://www.flyzipline.com (accessed July 22, 2024).

[10] M. Faithful. “Walmart Drone Dream Has Been Cleared For Take Off In Dallas.” Forbes. https://www.forbes.com/sites/markfaithfull/2024/05/13/walmart-drone-dream-has-been-cleared-for-take-off-in-dallas/ (accessed June 10, 2024).

[11] I. S. Smith, J. L. Rios, D. Mulfinger, V. Baskaran, and P. Verma, “UAS Service Supplier Checkout,”National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California, NASA/TM–2019–220456, 2019.

[12] J. L. Rios, I. S. Smith, P. Venkatesen, J. R. Homola, M. A. Johnson, J. Jung. “UAS Service Supplier Specification,” National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California, NASA/TM-2019-220376, 2019.

[13] “Pixhawk 6X (ICM-45686).” Holybro Store. https://holybro.com/products/pixhawk-6x?variant=44390763397309 (accessed July 23, 2024).

[14] “Object Avoidance - Copter documentation.” Ardupilot. https://ardupilot.org/copter/docs/common-object-avoidance-landing-page.html (accessed July 27, 2024).

[15] “Sky-Drones SmartAP Power Distribution Board - Copter documentation.” Ardupilot. https://ardupilot.org/copter/docs/common-smartap-pdb.html (accessed July 29, 2024).

[16] “What is an ESC?” Advanced Power Drives. https://powerdrives.net/blog/what-is-an-esc (accessed July 24, 2024).

[17] “SiK Telemetry Radio V3.” Holybro.com. https://docs.holybro.com/radio/sik-telemetry-radio-v3 (accessed July 26, 2024).

[18] J. Wigard. "Now it is time for drones to connect to mobile networks." Nokia. https://www.nokia.com/blog/now-it-is-time-for-drones-to-connect-to-mobile-networks/ (accessed August 9, 2024).

[19] dxiang. “Is It Bad to Charge An Electric Vehicle to 100%.” Midtronics. https://www.midtronics.com/blog/is-it-bad-to-charge-an-electric-vehicle-to-100/ (accessed July 28, 2024).

[20] J. Deng, X. Yang, and G. Zhang. “Simulation study on internal short circuit of lithium ion battery caused by lithium dendrite.” Materials Today Communications. https://doi.org/10.1016/j.mtcomm.2022.103570 (accessed July 28, 2024).

[21] OMRON. “Overview of Photoelectric Sensors | OMRON Industrial Automation.” www.ia.omron.com. https://www.ia.omron.com/support/guide/43/introduction.html (accessed July 27, 2024).

[22] “Photoelectric Sensor Explained (with Practical Examples) - RealPars.” www.realpars.com. https://www.realpars.com/blog/photoelectric-sensor (accessed July 29, 2024).

[23] “Polarized Retroreflective.” Tri-Tonics. https://www.ttco.com/sensors/products/retroreflective.html (accessed July 29, 2024).

[24] “Diffuse Sensors.” www.ttco.com. https://www.ttco.com/sensors/products/diffuse.html (accessed July 28, 2024).

[25] “What is a photoelectric sensor?.” www.ttco.com. https://www.ttco.com/what-is-a-photoelectric-sensor (accessed July 25, 2024).

[26] “3 Photoelectric Sensing Modes and How to Choose.” Banner Engineering. https://www.bannerengineering.com/my/en/company/expert-insights/3-photoelectric-sensing-modes-how-to-choose.html (accessed July 28, 2024).

[27] “Soberton Inc. SP-1605.” DigiKey. https://www.digikey.com/en/products/detail/soberton-inc./SP-1605/3973691?utm_adgroup=&utm_source=google&utm_medium=cpc&utm_campaign=PMax%20Shopping_Product_Low%20ROAS%20Categories&utm_term=&utm_content=&utm_id=go_cmp-20243063506_adg-_ad-__dev-c_ext-_prd-3973691_sig-CjwKCAjwnqK1BhBvEiwAi7o0X-oevfv6yndLqwf-CLvaOIrREh5IDqpQaTkI8prVRTNvh0joUqIJ7BoCIwkQAvD_BwE&gad_source=1&gclid=CjwKCAjwnqK1BhBvEiwAi7o0X-oevfv6yndLqwf-CLvaOIrREh5IDqpQaTkI8prVRTNvh0joUqIJ7BoCIwkQAvD_BwE (accessed July 22, 2024).

[28] Jb-Dev. “Navigation meshes and pathfinding.” GameDev.net. https://www.gamedev.net/tutorials/programming/artificial-intelligence/navigation-meshes-and-pathfinding-r4880/ (accessed July 20, 2024).

[29] “DSA Dijkstra's Algorithm.” W3Schools. https://www.w3schools.com/dsa/dsa_algo_graphs_dijkstra.php (accessed July 25, 2024).

[30] E. C. Navone. “Dijkstra's Shortest Path Algorithm - A Detailed and Visual Introduction.” freeCodeCamp. https://www.freecodecamp.org/news/dijkstras-shortest-path-algorithm-visual-introduction/ (accessed July 25, 2024).

[31] GeeksforGeeks. “What is Dijkstra’s Algorithm? Introduction to Dijkstra s Shortest Path Algorithm.” GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-dijkstras-shortest-path-algorithm/#dijkstras-algorithm (accessed July 25, 2024).

[32] T. Abiy, H. Pang, C. Williams, J. Khim, and E. Ross. “Dijkstra’s Shortest path algorithm.” Brilliant. https://brilliant.org/wiki/dijkstras-short-path-finder/ (accessed July 23, 2024).

[33] T. Abiy, H. Pang, B. Tiliksew, K. Moore, and J. Khim. “A* Search” Brilliant. https://brilliant.org/wiki/a-star-search (accessed July 25, 2024).

[34] GeeksforGeeks. “A search algorithm.” GeeksforGeeks. https://www.geeksforgeeks.org/a-search-algorithm (accessed August 1, 2024).

[35] R. A. S. “A* Algorithm concepts and implementation.” Simplilearn. https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/a-star-algorithm (accessed July 26, 2024).

[36] K. Moore, K. Jennison, and J. Khim“Depth-First Search (DFS) | Brilliant Math & Science Wiki.” Brilliant. https://brilliant.org/wiki/depth-first-search-dfs (accessed July 21, 2024).

[37] GeeksforGeeks. “Depth first search or DFS for a graph.” GeeksforGeeks. https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph (accessed August 1, 2024).

[38] “What is a Neural Network? - Artificial Neural Network Explained - AWS.” Amazon Web Services, Inc. https://aws.amazon.com/what-is/neural-network\ (accessed August 1, 2024).

[39] “What is Supervised Learning?” Google Cloud. https://cloud.google.com/discover/what-is-supervised-learning (accessed July 23, 2024).

[40] A. Sharma. "Introduction to the YOLO family" PyImageSearch. https://pyimagesearch.com/2022/04/04/introduction-to-the-yolo-family/ (accessed July 26, 2024).

[41] J. Jordan. "An overview of object detection: one-stage methods." Jeremy Jordan. https://www.jeremyjordan.me/object-detection-one-stage/ (accessed July 26, 2024)

[42] "Explained: Neural networks." MIT News. Apr. 14, 2017. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414 (accessed July 24, 2024).

[43] A. Rosebrock. "Object detection: Bounding box regression with Keras, TensorFlow, and Deep Learning." PyImageSearch. https://pyimagesearch.com/2020/10/05/object-detection-bounding-box-regression-with-keras-tensorflow-and-deep-learning/ (accessed July 23, 2024).

[44] "How do one-stage and two-stage detectors compare? | 5 Answers from Research papers." SciSpace. https://typeset.io/questions/how-do-one-stage-and-two-stage-detectors-compare-163ds3vkrc# (accessed July 28, 2024).

[45] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." ArXiv. https://arxiv.org/abs/1506.01497 (accessed July 28, 2024).

[46] GeeksforGeeks. "Region Proposal Network (RPN) in object detection." GeeksforGeeks. May 6, 2024. https://www.geeksforgeeks.org/region-proposal-network-rpn-in-object-detection/ (accessed July 29, 2024).

[47] Sambasivarao. K, "Region of interest pooling - towards data science." Medium, Oct. 30, 2023. https://towardsdatascience.com/region-of-interest-pooling-f7c637f409af (accessed July 27, 2024).

[48] "What are Convolutional Neural Networks? | IBM." https://www.ibm.com/topics/convolutional-neural-networks (accessed July 30, 2024).

[49] M. Tan, R. Pang, and Q. Le V. "EFfIcientDET: Scalable and efficient object Detection." ArXiv. https://arxiv.org/abs/1911.09070 (accessed July 29, 2024).

[50] "Papers with Code - BiFPN Explained." https://paperswithcode.com/method/bifpn (accessed July 24, 2024).

[51] GeeksforGeeks. "Backpropagation in Neural Network." GeeksforGeeks, July 09, 2024. https://www.geeksforgeeks.org/backpropagation-in-neural-network/' (accessed July 31, 2024).

[52] "State of Homelessness: 2023 Edition." National Alliance to End Homelessness. https://endhomelessness.org/homelessness-in-america/homelessness-statistics/state-of-homelessness/ (accessed July 28, 2024).