SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Date: June 10, 2021

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Connor Azzarello
Chris Gerbino
Ruchir Mehta

ENTITLED

Enhanced Sensing Methods for UAV-Based Disaster Recovery

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING

____
Thesis Advisor

____
Department Chair

Enhanced Sensing Methods for UAV-Based Disaster Recovery

by

Connor Azzarello
Chris Gerbino
Ruchir Mehta

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science & Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 10, 2021

Enhanced Sensing Methods for UAV-Based Disaster Recovery

Connor Azzarello
Chris Gerbino
Ruchir Mehta

Department of Computer Science & Engineering
Santa Clara University
June 10, 2021

ABSTRACT

A good abstract is a concise summary (1–2 paragraphs) of the entire project: introduction, problem statement, work accomplished, results, conclusions, and recommendations. When you write the abstract, imagine that the reader will not read anything else, but that you must get your major point across immediately. This requires efficiency of words and phrases. An abstract is written to stand alone, without jargon or reference to figures and tables in the report body.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

1.1.1 Motivation

Natural and human-caused disasters can cripple, displace, and diminish civilian populations. Disasters are not only physically damaging to the communities but also have a major economic impact. In 2017 alone, there were 301 disasters and, on average, 60,000 deaths per year attributed to natural disasters. In 2017, North America suffered an economic loss of $244 billion dollars due to natural disasters [Insert Reference]. Over the past century, the rate at which these catastrophes occur has skyrocketed. Since global climate change is a large contributor to the increasing rate of disasters, the upward trend in the number of natural disasters is likely to continue. At Santa Clara University, as Jesuits, we are expected to uphold considerable moral values. The most important being service to others. Jesuits are encouraged to consider actions that can better the lives of others less fortunate than themselves, as Jesus did in his time. For our project, we tried to create a service that would benefit victims and disaster-response teams alike.

1.1.2 UAV-Based Recovery

Unmanned aerial vehicles, or UAVs, are used as technological solutions to increase the effectiveness of first responders during the search and rescue phase of disaster response. Existing UAV disaster response technology has proven to be quite effective when applied to various emergency situations. Currently, UAVs are being used for Disaster relief situations such as Hazardous chemical spills, the need for mapping high risk areas, assessing structural damage, delivering emergency infrastructure and supplies, and extinguishing wildfires.

1.1.3 Problem

Commercial UAV-based disaster response technology is cost prohibitive and therefore it is not accessible by non-governmental organizations and developing countries. This is problematic as natural and human-caused disasters are often more devastating to developing countries and low-income areas.

A natural disaster that occurs in the US will have a more collaborative and expensive response turnout, compared to a lesser-developed country. In the US, relief teams, funds, and problem solving are all strategized by the Department of the Interior. Besides designating federal funds the department also organizes expenses through local, state, territorial, and stakeholders. On top of this is advanced technology like manned aircraft and strong communication infrastructure. If needed, other developed countries never fail to send funds, as they have in the past for the many hurricanes that have ravaged the south. The benefits listed diminish in scale the more underdeveloped the country is. It is understandable how an earthquake at the magnitude of Haiti's would result in way less losses.

1.1.4 Challenges

COVID-19

The COVID-19 pandemic has been the largest challenge for our group. Each group member had to be isolated for the duration of this project. This meant that we could not hold in-person meetings and therefore we were less productive when troubleshooting hardware or software issues. Furthermore, the lack of in person interaction meant we couldn't just walk into our project advisor's office and ask for help.

Testing

On the topic of testing, we quickly ran into problems flying our drone. Due to city regulations, a permit must be first acquired before users are allowed to fly a drone at any altitude. Moreover, the school was unable to accommodate any exceptions to the restrictions which created a major setback to our project. We originally planned on creating our own test data and turning to an imaging dataset to fill in the gaps to what our testing could not provide. Now, we had to completely rely on an aerial imaging dataset for our model.

Hardware

Before we were limited to only indoor spaces by city regulations, we performed tests in a local park. During one of our runs, we had a slight malfunction from one of our motors which stopped midair. Since this was during our preliminary testing stages, we hadn't coded error handling into our drone, so the drone fell almost 12 feet out of the air. While the drone did land relatively softly on the grass, the motor sustained considerable damage and needed to be sent back to the manufacturer for repairs. A more significant challenge we faced all throughout development was ensuring our battery would be sufficient in powering all our electronics. Throughout the project, we were constantly monitoring whether our power consumption was optimal and not surpassing the wattage required for the other team's hardware to work properly.

Holybro S500

1.1.5 Solution

We will create a low-cost UAV-based solution to assist in disaster response. The novelty of our solution is the emphasis on frugal innovation such that our relatively low-cost product will be a realistic option for developing countries and humanitarian organizations. Something about design thinking ?

Throughout this project we will also be working in parallel with another group consisting of two members Mark Rizko and Cameron Burdsall. Mark and Cameron are working to create a drone mesh WiFi system for disaster scenarios. When both projects have been completed, we will combine the technologies to create a more complete solution for disaster recovery. Our combined cost-effective solution will have the capabilities to assist in safely identifying victim locations, and resurrecting destroyed communication infrastructure.

1.1.6 Evaluation Results

¡Insert Final evaluation results¿

Chapter 2

Related Works

2.1 Related Work

In creating our solution, we decided to draw inspiration from other studies already produced in the field of object detection and computer vision. By doing extensive research on other projects, we were able to narrow down on the best dataset to use as well. In the first project we looked at, we discovered the company Nanonets that specializes in making convolutional neural networks for aerial drones. As it turns out, drones are used for a variety of industry applications such as surveillance over solar panel farms, monitoring progress in construction projects, and various other cases. We were able to gather some information about the algorithms used but could not apply it heavily to our project as they didn't have any written documentation on human detection. A larger scale project we found was one from a Microsoft team, which created a drone system of manned aerial vehicles (MAVs) to persistently track one individual. It could be for the purposes of tracking or other security-based applications. Like our project, the Microsoft team implemented an approach that avoided performing detection on every frame, and instead, focused on creating a performance-effective pattern detection hypothesis on frames which, if deemed to be important, would then have detection performed on the relevant frames. We followed this approach since applying detection per frame for our system would be very power intensive, and disallow for longer flight times or the redirection of power to our embedded systems. An important related work in our process to finding a reliable imaging dataset was a paper by TODO which outlined different available training datasets for projects that could not afford the time or means to create their own. In the article are many notable examples like Haar Cascade and MobileNet. It dives into the differences and similarities between the two, the situations where one is better than the other, and implementation required.

Chapter 3

Requirements

3.1 Requirements

3.1.1 Functional Requirements

Critical

Efficacy Precise & Accurate Identification

A system based mainly on object-detection capabilities is only as useful as the accuracy of the resulting data. Most of the popular object-detection frameworks that currently exist display results based on a certain confidence threshold generated by the model itself. Therefore, it is imperative that this threshold be set to a very high value, in order to limit the number of false positive results. On the other hand, false negatives are similarly problematic and contingent on the training data used to train the model.

Speed Fast Inference Speed

The term *inference* refers to the speed at which a computing device can process a set of data and output a result using a mathematical model. In this project, the "sets of data" that will be processed are digital images representing each frame of the video. The model that we will employ is likely to be one of the many industry standard deep learning models that are commonly used for computer vision. Inference speed in such a scenario is more accurately represented as the number of video frames that can be processed per second. Due to the state of low-power computing and wireless communication at this time, it is likely not feasible to display this video feed in real time to users, but every extra frame that can be captured and processed each second makes the resulting data more accurate and provides more data points to first responders.

Notification Real-Time Notification of Detection

As mentioned above, the power usage and computing overhead necessary to transmit an annotated video stream to users is likely to high to be technically feasible in this application. However, the data that will be generated by the imaging array on each UAV is incredibly useful to disaster response teams. As a result, it was deemed a critical requirement that the system must provide instantaneous notification to users when a human (or other object class depending on circumstances) is detected by the system. This notification allows the disaster response teams to modify their flight path or, where necessary, take manual control of the flight in order to procure more data that may save additional lives.

Power Lowest Possible Power Footprint

The maximum flight time of the UAV is completely contingent on the battery capacity available to its DC motors while in flight. Therefore, it is imperative that any auxiliary functions require as little power as possible. Preliminary calculations showed us that the maximum power draw of most single board computers that exist today are significantly lower than that of drone flight, but every optimization is useful, and incredibly important, when such a system is deployed into a disaster scenario.

Recommended

Metrics Count the number of victims that have been identified along a flight path

UAV Cooperation Allow for multiple UAV's to be used simultaneously during a disaster response scenario

Suggested

Chapter 4

Use Cases

4.1 Use Cases

4.2 Activity Diagram

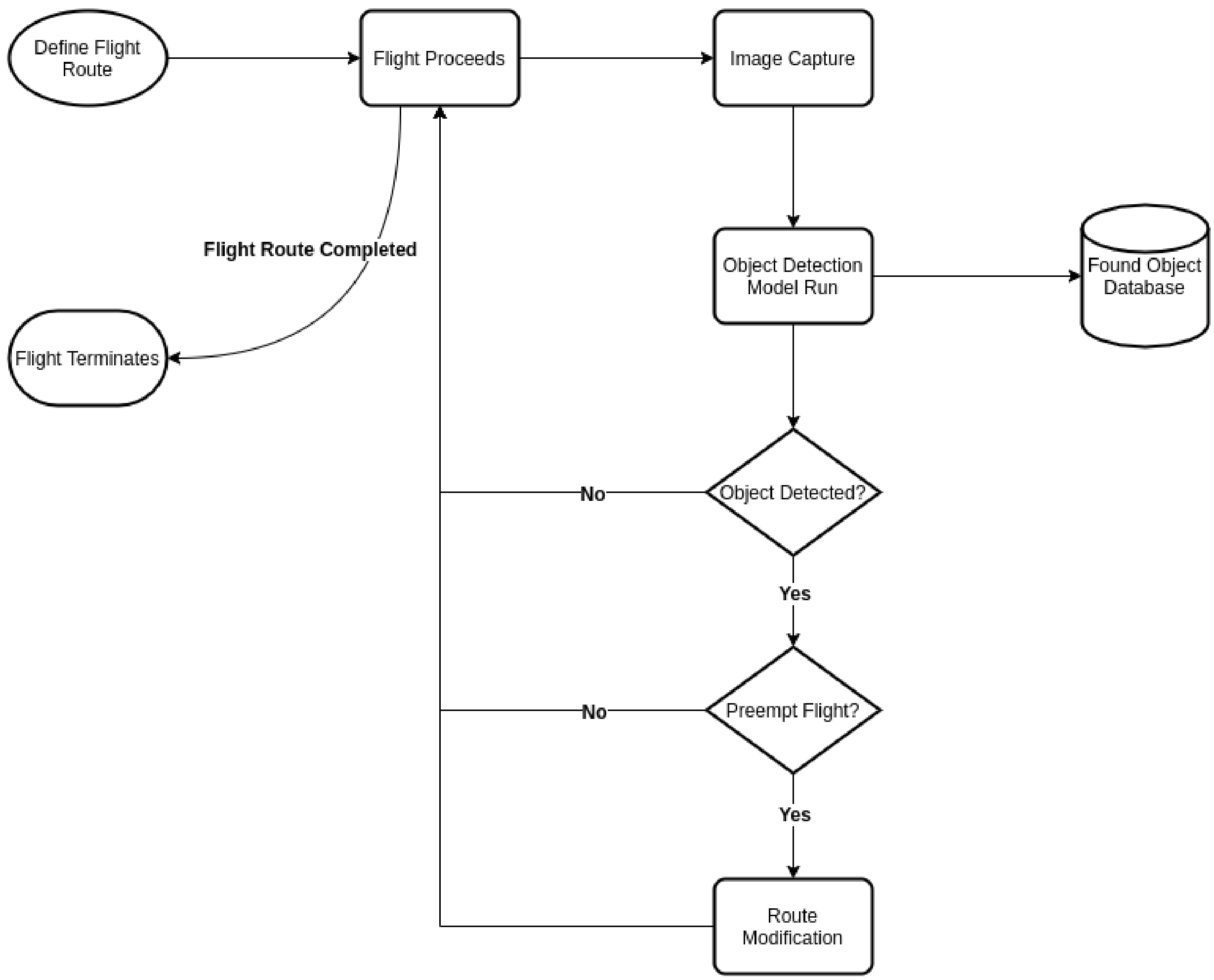

Figure 4.1: Activity Diagram

Chapter 5

Proposed Solution

5.1 Proposed Solution

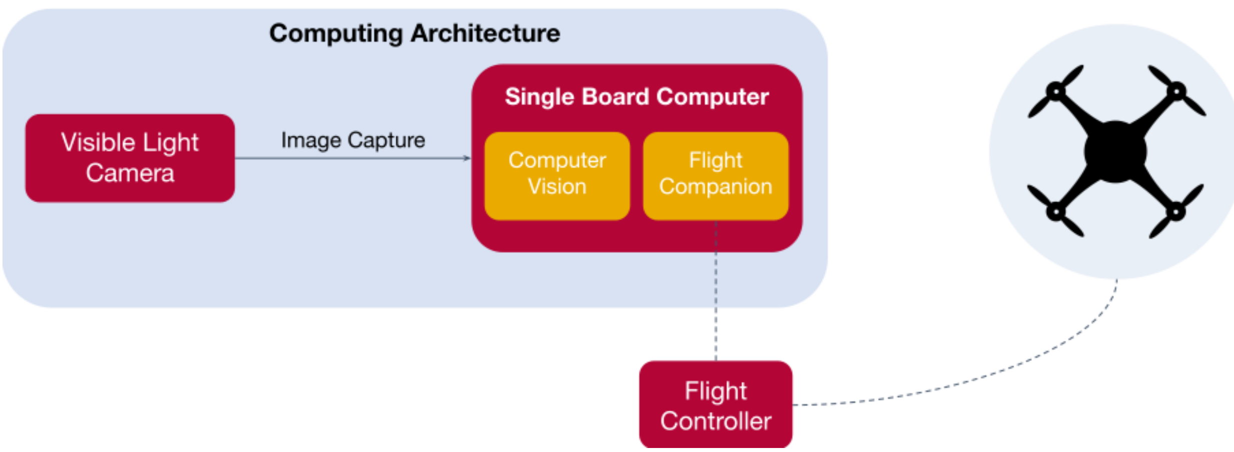

Figure 5.1: Proposed Solution

5.1.1 Hardware

Single Board Computer Nvidia Jetson Nano - 4 GB

A single board computer was imperative to achieve the non-functional design requirements outlined in 3.1, because of their low cost, energy efficiency, size and modularity. After researching performance metrics for single board computers (SBCs) across a variety of indsutry standard computer vision models, it was determined that the NVidia Jetson Nano would be the most cost-effective and performant SBC choice, especially compared to competing platforms. On one hand, the Raspberry Pi series of SBCs would be satisfactory in regard to power usage, but significantly underpowered for computer vision applications. On the other hand, the Google Coral TPU Dev Board boasted incredible inference speeds where comparison was possible, but its major caveat consisted of only being capable of running Tensor Flow Lite (TFLite) models, which would have a significant detriment on the modularity and flexibility of the overall platform.

UAV Frame HolyBro S500v2 Kit

In order to achieve the requirements of competitive flight time (compared to enterprise solutions), an accessible pricepoint, and a suitable size to mount all components, the choice was quickly narrowed down to a *quadcopter*, meaning a UAV with four propellers. A quadcopter is useful for this use-case because...

Flight Controller PixHawk 4

The flight controller acts as the textitbrain of a UAV system. The PixHawk 4 was chosen because it satisfies three primary requirements of this project's design:

- Autonomous flight
- Detailed logging
- Interface to companion computer

5.1.2 Software

Flight Software QGroundControl

QGroundControl boasts a high degree of compatibility with leading consumer and enterprise-level drone systems, including the HolyBro S500 and PixHawk 4 platform we chose for this project.

- Various route-planning tools
- Universal compatibility
- Open-souce software

5.1.3 Developer Tools

JetPack Software Development Kit Includes:

- Jetson Linux Driver Package (L4T)
- Ubuntu 18.04 LTS Build
- CUDA accelerated libraries and APIs
- Samples, documentation and developer tools

NGC Catalog Package Manager

Easy access to AI building blocks

- Models
- Containers
- SDKs

Chapter 6

System Evaluation

6.1 System Evaluation

6.1.1 Computer Vision Performance

6.1.2 Price

6.1.3 Comparison

Model	Application	Power Profile	Inference Speed	Power Consumption	GPU Latency	GPU Load	CPU Load
Inception v4	Classification	5W	10.58 FPS	5.029 W	94.477 ms	56.16%	85.03%
		10W	10.60 FPS	5.570 W	94.340 ms	70.30%	40.31%
VGG-19		5W	10.09 FPS	5.150 W	99.121 ms	58.92%	84.74%
		10W	10.04 FPS	5.004 W	99.596 ms	39.79%	49.45%
ResNet 50		5W	36.67 FPS	5.029 W	27.268 ms	54.56%	86.90%
		10W	36.88 FPS	5.564 W	27.116 ms	63.80%	41.56%
SSD Mobilenet v1	Object Detection	5W	42.56 FPS	4.360 W	23.498 ms	58.49%	94.82%
		10W	42.64 FPS	5.097 W	23.454 ms	77.88%	44.77%
TinyYolo v3		5W	47.65 FPS	3.541 W	20.987 ms	14.25%	95.31%
		10W	47.60 FPS	4.325 W	21.008 ms	24.64%	50.62%
UNet	Semantic Segmentation	5W	16.59 FPS	3.281 W	60.266 ms	0.45%	92.68%
		10W	16.62 FPS	5.882 W	60.177 ms	89.76%	36.53%
Super Resolution	Image Processing	5W	15.26 FPS	6.396 W	65.524 ms	86.82%	78.00%
		10W	15.26 FPS	5.943 W	65.547 ms	82.51%	36.94%
OpenPose	Pose Estimation	5W	14.64 FPS	3.258 W	68.313 ms	0.26%	92.46%
		10W	14.64 FPS	3.968 W	68.320 ms	0.00%	46.42%

Product	**Price**
NVIDIA Jetson Nano 4 GB	$120
HolyBro S500 v2 Kit	$355
Pixhawk 4	$70
8000mAh Battery & Charger	$270
IMX219-77 Camera	$20
Total	**$835**

	Prototype	**Industry Standard Averages**
Flight Time	25.7 min	31.5 min
Computer Vision	Real-time up to ~480p 45 fps	No
Autonomous Flight	Yes	Yes
Modularity	High	Minimal
Price	$835	$6,750