Multispectral Camera for Monitoring Plant Health: Testing Report

Ethan Schneider

***Abstract*—Multispectral imaging allows for the characterization of plant health and photosynthetic activity based on reflectance values of plants and their surroundings in specific bands of the electromagnetic spectrum. This project is a continuation of a previous semester project; this report provides an overview of the previous project, as well as plans for further means of data collection and processing for enhanced characterization of plant life. In addition, this report will provide background on the subject of multispectral imaging, and detail how multispectral imaging is used for detection of plant health.**

I. Introduction

A. Multispectral Imaging

Multispectral imaging typically captures 3-15 narrow spectral bands of electromagnetic radiation, in multiple regions of the spectrum including visible light, near infrared (NIR) light, and infrared light. Common bands that are captured include blue (450-520 nm), green (520-590 nm), red (590-690 nm), near infrared (750-900 nm), mid infrared (1550-1750 nm), far infrared (2080-2350 nm), and thermal infrared (10400-12500 nm). Different applications make use of different combinations of these bands as well as others. For precise and accurate measurements, the bands that are captured often cover only a few nanometers of wavelength. There are a few ways that this can be done. A relatively cheap and easy method of acquiring narrow band measurements is to use narrow-band filters placed in front of the lenses of a camera. However, this can have some drawbacks, as it can lead to leakage of undesirable wavelengths. Another method is using sensors designed specifically to detect a certain wavelength or narrow band of light. Such devices are more expensive and often specially developed but are much more accurate and often used in large scale applications like space-borne sensor arrays.

B. Previous Work

The previous project used a Raspberry Pi 2 Model B V1.1 and a Raspberry Pi NoIR Camera module to calculate and display, in approximately real time, the Normalized Difference Vegetation Index (NDVI). The NoIR Camera module, which has the IR blocking filter removed connects to the Raspberry Pi through the camera interface built into the Pi. A narrow-band blue filter was placed over the camera sensor to block red and green wavelengths of light from entering the sensor, while allowing blue and infra-red wavelengths to pass through. The program, a python script, then uses the following equation to approximate an NDVI value for each point in the picture:

$$NDVI = \frac{(NIR - Blue)}{(NIR + Blue)} \quad (1)$$

A colormap is then applied to the resulting grey-scale image to allow for easy interpretation. The project was meant to be battery powered and portable, which was accomplished to an extent, but it was constructed primarily with parts sourced from the Robotics and Machine Intelligence Lab, which limited its operating time and resulted in a slightly unwieldly final design. In addition to the Raspberry Pi and camera module, a 7-inch LCD was used to display the images, a battery pack containing 8 AA batteries was used for power, and a Phoenix Contact DC to DC voltage converter was used to provide 5 volts for the Raspberry Pi from the 12 volts provided by the batteries.

C. Project Overview

For the next iteration, focus will rest upon reducing the overall size of the device, while adding accuracy and functionality. Significant size and weight savings will be made by using a smaller battery pack, likely a type of lithium battery, as well as a smaller DC to DC convertor, as these are the two largest individual components. Additionally, a custom designed and 3D printed enclosure for the entire device will greatly reduce the physical size and make the device easier to handle. The main focus will be on increasing the accuracy and variety of measurements used to evaluate plant health. This will be accomplished through use of an additional camera with narrow-band filter to exclude more undesirable wavelengths of light and thus increase accuracy of the measurements. The addition of a second camera will allow for the capture of additional bands of light to increase the number of measurements being taken. These tasks will be accomplished by replacing the Raspberry Pi 2 with a Raspberry Pi Compute Module 3, and a corresponding IO Board or StereoPi board which allow for two camera connections. The purpose of the new device will differ from the previous; the previous was a proof of concept, while the new one will serve to provide useful data. After the testing phase, and once both cameras are functioning, the data processing will be offloaded from the Raspberry Pi to either an external computer or a cloud computing service, and the new device will eventually exclude a built-in display. Eventually the intention is that the device will be able to be mounted on a drone to be able to provide data on large patches of land like forest canopies or agriculture fields. Another form of functionality that may be added is auto calibration and correction for ambient light intensity by using photoresistors to correct for variances in light intensity of the environment. Finally, once all of the planned functionality has been achieved and in order to achieve the smallest and lightest possible device, a custom PCB will likely be designed and manufactured that includes only the necessary ports and connections. A final enclosure will also be designed to contain and protect the components.

D. Design Constraints - Problem Definition

There is no specific customer with any requirements that this project must adhere to. The purpose of this project is to build on existing work that uses multispectral imaging for characterizing plant health while using relatively cheap and easy to acquire components while adhering to a small budget. This type of device has uses in agriculture and conservation, and while similar devices do exist (see Section II Part G), the information that they are able to provide, while very useful, is not any different than the information that can be had from a cheaper, non-proprietary device, such as the one this project plans to provide.

E. Timeline

The timeline for this project consists of a few milestones at various points throughout this semester and the next at which certain elements of the project should be completed:

- Friday Oct. 25, 2019: Component list finalized, and components ordered
- Wednesday Nov. 27, 2019: Components assembled, capturing images, and performing basic processing
- Friday Dec. 6, 2019: Complete first YouTube video
- Friday Feb. 7, 2020: Image processing offloaded from the Pi to an external device
- Friday Feb 28, 2020: Add functionality to image processing
- Friday March 13, 2020: Perform functionality verification and testing
- Friday March 27, 2020: Design and 3D print enclosure for device
- Thursday April 9, 2020: Complete poster for Scholarship Day, and schedule to have it printed
- Tuesday April 21, 2020: Scholarship Day poster & presentation
- Friday May 1, 2020: Complete final YouTube video

Most of these dates are simply an estimate, the actual implementation of these goals may be completed sooner or later than planned depending on the difficulty of implementation or any unforeseen problems.

F. Budget

The total budget for the project should not exceed $200. Some existing equipment will be used, such as a Raspberry Pi NoIR Camera Module, while general materials like wires and solder will be acquired from the Robotics and Machine Intelligence Lab. Any other necessary materials will be purchased. A Raspberry Pi Compute Module 3 and a standard Raspberry Pi Camera Module can be had for $25 each. A board will be needed to interface with the Raspberry Pi Compute Module, and either a Raspberry Pi Compute Module prototyping board (~$45) or a StereoPi board ($60-90). These components will account for the majority of the project's cost, and will total in the range of $100 to $140, which leaves a significant amount of extra budget for contingency and will allow for flexibility in choosing components or for the possibility of the addition of components for added functionality as the project progresses.

G. Impacts

The impacts of multispectral imaging in the areas of agriculture and conservation are discussed elsewhere (Section I Part A and Section II Parts A-F). Socially and ethically, this project will likely not have any impacts. This project specifically on a large scale, will likely have minimal impact, it is mostly intended to demonstrate that this type of imagery and the data that it provides can be captured accurately and in a similar manner to existing devices while using relatively cheap, off the shelf hardware for a fraction of the cost of existing devices. While there is no specific plan in place at this time, this project could theoretically be used by local farmers or in the college's organic garden to provide information about crop health.

II. Background

A. Simple Ratio

The simple ratio (SR) is the simplest index used for classification of plant health and chlorophyll content. The simple ratio, as the name says, looks at the ratio of the reflectance of light in the near infrared (NIR) band of the spectrum and red band of the spectrum. Green vegetation exhibits low reflectance of light in the red and blue parts of the spectrum, because it is absorbed by chlorophyll when performing photosynthesis. The reflectance of plants is also high in the NIR region, as infrared light does not carry enough energy to allow photosynthesis to take place. The formula for SR is as follows:

$$SR = \frac{\rho_{NIR}}{\rho_{Red}} \tag{2}$$

This formula, when applied to an image, provides a quick way to distinguish green vegetation from other objects, like soil, and to estimate the relative biomass in the image. Additionally, it can be used to help distinguish stressed vegetation from non-stressed vegetation. SR values close to 1 mean that an object has similar reflectance in the NIR and red regions, like soil, while green plants have much higher NIR reflectance than red reflectance, which will result in a value much greater than 1 [1].

B. Normalized Difference Vegetation Index

The Normalized Difference Vegetation Index (NDVI) takes advantage of some reflective properties of chlorophyll, which are a part of plants that allow them to absorb energy from light and perform photosynthesis. Two types of chlorophyll exist, chlorophyll A and B. The absorbance characteristics of the two types of chlorophyll differ slightly, but they exhibit very similar trends. In general, chlorophyll absorbs light in the blue and red regions of the visible spectrum, and reflects green wavelengths, which is why humans perceive plants as green. This is because the wavelengths associated with red and blue light are able to be used in the process of photosynthesis, while photons at shorter wavelengths tend to be so energetic that they can cause damage to a plant's cells and DNA, and longer wavelengths do not carry enough energy to allow photosynthesis to take place. Plants have evolved to reflect these longer wavelengths, including in the near-infrared part of the spectrum (about 700-

1100 nm). The following equation describes how NDVI values are calculated:

$$NDVI = \frac{(NIR-Red)}{(NIR+Red)} \quad (3)$$

Where NIR represents the spectral reflectance measurements acquired in the near-infrared region, and Red stands for the spectral reflectance measurements acquired in the red (visible) region. This equation results in values in the range of -1.0 to +1.0. This value is directly related to the photosynthetic capacity and energy absorption of the plant(s) being photographed. A variation of NDVI, called the Normalized Difference Red-Edge Index, or NDRE, follows the same principles as NDVI, but uses reflectance measurements acquired in a narrow band of the spectrum between the transition of red to infra-red called red-edge (around 730 nm).

$$NDRE = \frac{(NIR-Red\_Edge)}{(NIR+Red\_Edge)} \quad (4)$$

The red-edge band of the spectrum is more capable of penetrating plants, as lower wavelengths of red light are absorbed by chlorophyll in the first few layers of the plant [1]. Typically, regions containing dense foliage or plants with high amounts of chlorophyll tend to result in NDVI and NDRE values in the range of 0.2-0.8, while inorganic surfaces tend to result in values closer to 0 or negative values.

C. Photochemical Reflective Index

The Photochemical Reflective Index (PRI) is another reflective measurement, like NDVI, that is able to characterize plant photosynthetic efficiency. PRI is calculated as follows:

$$PRI = \frac{(\rho 550 - \rho 531)}{(\rho 550 + \rho 531)} \quad (5)$$

Where ρ represents reflectance at the specified wavelength (531 or 550 nm). ρ531 was chosen because it is strongly correlated with epoxidation state (EPS) in plants, which is an indicator of short-term changes in photosynthetic activity. ρ550 represents a reference wavelength, 550 nm was chosen simply because it results in less drift in the reflectance vs EPS relationship. EPS is a convenient method of expressing the relative concentrations of the xanthophyll cycle pigments. The concentrations of these pigments, zeaxanthin (Z), antheraxanthin (A), and violaxanthin (V), change with variations in the amount of absorbed photosynthetically active radiation (PAR). As the amount of PAR increases, the concentration of zeaxanthin increases, while the concentration of violaxanthin decreases. Likewise, the opposite is true when the amount of absorbed PAR decreases. EPS itself can also be calculated, though not through remote sensing:

$$EPS = \frac{(V+0.5*A)}{(V+A+Z)} \quad (6)$$

EPS must be calculated using area-based molar concentrations of violaxanthin, antheraxanthin, and zeaxanthin. PRI measures carotenoid pigments in live plants, which are used as indicators of photosynthetic light use efficiency, and plant stress [3]. Like NDVI, the values produced by the equation for PRI range from -1.0 to +1.0.

D. Chlorophyll Index

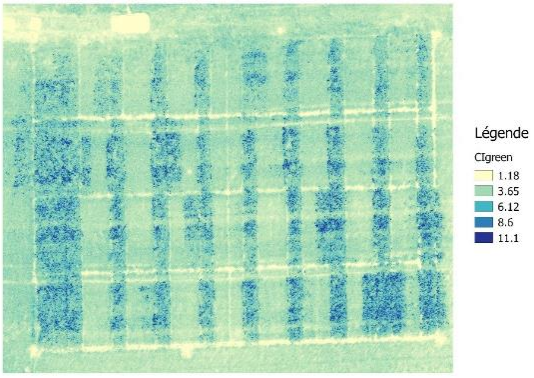
The chlorophyll index (CI) measures the total chlorophyll content of plants. Chlorophyll is used in the photosynthetic process to harness the energy contained in photons to create useful compounds for the plant. The chlorophyll index is sensitive to variations in the content and concentration of chlorophyll in many species of plants. There are two methods of calculating the chlorophyll index, CI green and CI red-edge, both of which are used in remote sensing. The formula for CI green is as follows:

$$CI_{Green} = \frac{\rho_{NIR}}{\rho_{Green}} - 1 \quad (7)$$

Where ρ_{NIR} represents the spectral reflectance measurements acquired in the NIR region of the spectrum (typically around 850 nm), and ρ_{Grenn} represents the spectral reflectance measurements acquired in the green part of the spectrum (around 530 nm). The formula for CI red-edge is as follows:

$$CI_{Red-Edge} = \frac{\rho_{NIR}}{\rho_{Red-edge}} - 1 \quad (8)$$

Where ρ_{NIR} represents the spectral reflectance measurements acquired in the NIR region of the spectrum (typically around 850 nm), and $\rho_{Red-edge}$ represents the spectral reflectance measurements acquired in a narrow band of the spectrum between the transition of red to infra-red (around 730 nm) [1].



Example image using chlorophyll index to measure chlorophyll content in an agricultural setting [1].

E. Leaf Water Content

Leaf water content is often a large factor limiting the efficiency of photosynthesis in plant life. At this time there is no method of determining leaf water content non-destructively using only thermal infrared, NIR, and visible wavelengths, as these approaches are often influenced by other factors of the plants biology and are unsuitable for all species. [4]. However, there has been research that suggests for certain species, machine learning may be able to determine leaf water content non-destructively using visible and NIR wavelengths of light [4].

F. Leaf Area Index

Leaf area is an important factor in overall photosynthetic activity, the more area of green vegetation that is present and exposed to sunlight, the more photosynthesis is able to occur. In the past, the only method of measuring leaf area was either destructive and resulted in the loss of some of a plant's leaves, or relied on leaves falling from trees to determine the leaf area relative to the overall area observed. However, with the

introduction of remote sensing, there has been more interest and research into determining leaf area non-destructively using remote sensing. Leaf area can be estimated in two primary ways, the first involves the computation of spectral vegetation indices, which uses relationships between various indices and leaf area to estimate leaf area of the region. The second method uses bidirectional reflectance distribution function (BRDF) models. Using an optimization procedure, an inverted BRDF model with radiometric measurements, an estimate can be made of leaf area. However, the process is compute intensive, and obtaining the required input parameters can be difficult. Relatively new research has been done that shows that by inverting a BRDF model with a limited number of datapoints, and ensuring clean data, this information can be used either to fit a leaf area index equation, or to train a neural fuzzy system, both of which can then be applied to large scale remote sensing imagery to estimate leaf area measurements [5].

G. Existing Work

Multispectral cameras for use in agriculture and conservation do exist and are in production. Examples include Sentera's variety of sensors, the Parrot Sequoia, the Tetracam ADC, and the MicaSense RedEdge Sensor. However, these devices are proprietary and relatively expensive, generally around 3,000 to 5,000 US dollars. The theory behind the functionality of this type of device is already well established, but there is much work to be done to decrease costs while encouraging advancements in remote sensing.

III. Design

A. Components

The initial device will be constructed using a Raspberry Pi Compute Module 3+ Lite, mounted to a StereoPi board that interfaces with two camera modules, one of which has had its infrared filter removed. A Raspberry Pi Compute Module was chosen due to its small size and its ability to interface with two camera modules. A standard Raspberry Pi has a single camera connector, so the only way to connect more than one is by multiplexing the cameras into the single camera port. The Raspberry Pi Compute Module, on the other hand, is built into the form factor of a SO-DIMM module which allows for many more GPIOs and interfacing options and supports two camera interfaces out of the box. The StereoPi board was chosen over the other options due to its small size, which will allow for flexibility down the road when it comes to mounting the device on a vehicle or using it in the field. The StereoPi board also has the necessary I/O for interfacing with two cameras, as well as HDMI, Ethernet, and two USB which will be useful for development and configuring the Compute Module, while not having any other unnecessary I/O that would only serve to increase the size and weight of the board. The only downside of using a StereoPi is its relatively (in this scenario) high cost. At $70, it costs significantly more than most of the other options, but its ease of use, small size, and tailored I/O justify the cost. A range of general-purpose prototyping boards for use with the Raspberry Pi Compute Module are also available and were considered for this project. While these boards are generally cheaper than a StereoPi board, they usually include far more I/O options than are necessary for this project and are significantly larger. The ease of use is the same as the StereoPi, but when it comes time to move the project from development to integration into a final device, the large size would become a hindrance. Finally, designing a custom PCB that included only the necessary I/O was considered, but this option was ruled out because of the expected amount of development time and complexity in designing a board and having it manufactured. The StereoPi board accomplishes the same thing without the added development time. A Pugh matrix describing the design and component options and the relevant parameters can be seen below:

	Standard RPi with mux	StereoPi	Prototyping Board	Custom PCB
Cost	+	-	0	-
Size	-	+	-	+
I/O	+	+	+	+
Complexity	-	+	+	-
Total	0	2	1	0

B. Methodology

The first iteration of this device will focus on the capture of the necessary wavelengths of light in order to determine plant health, before moving on to the processing and displaying of the data. The setup is quite simple: The Raspberry Pi Compute Module is inserted into the SO-DIMM connector of the StereoPi board, and the two camera modules are connected to the camera interfaces with ribbon cables. A micro SD card flashed with a Raspbian image is inserted into the micro SD card slot of the StereoPi board. 5-volt power is provided to the power leads on the StereoPi board which provides power to the board itself as well as the Raspberry Pi, both camera modules, and any connected peripherals. Once everything was assembled, the Raspberry Pi was booted and configured. The camera interfaces were enabled, and the proper functioning of both cameras simultaneously was validated. A narrow band filter was added to the NoIR camera to allow IR wavelengths to pass while blocking red wavelengths.

After ensuring that both cameras were functioning correctly, work began on automating the capture of images through use of a Python script. The original approach was to automate the capture of images using a shell script with the reason being that a shell script could be automatically started upon boot of the Raspberry Pi, and it allows for easy control over camera parameters when performing image captures. Additionally, to minimize latency and the amount of elapsed time between captures of the two cameras, it would have been necessary to use a daemon running the PiCamera process continuously in the background. Without this process running, a capture command first has to initialize the PiCamera process and camera settings before actually taking a picture, which can elapse around 1 second between the command and the picture being captured. With the process running in the background, it simply must listen for a capture command, and is able to capture an image within milliseconds of receiving the command.

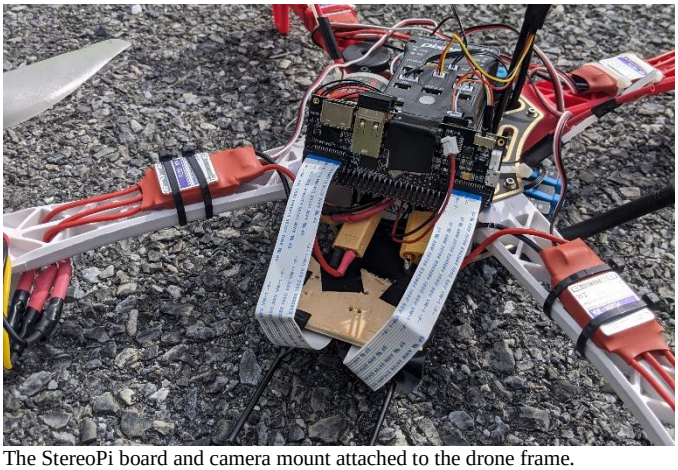
However, after initial testing, it became clear that the use of shell scripts would end up requiring more effort than simply using a single Python program. The daemon that would have been required to use shell scripting was itself composed of two python scripts, so it was simpler to code a single python program to capture and process the images. The final python program first initializes camera settings before beginning a continuous image capture sequence. Rather than use a daemon to keep the camera process running at all times, this program creates a video stream from the cameras and pulls images from the stream at set time intervals, currently once every second. The image captured from the video stream is actually two separate images, one from each sensor taken at the same time and displayed next to each other. The first step in processing the images for gathering data is to split the image into its left and right components. The left image contains the data from the NoIR camera module, while the right image contains the data from the normal camera module. After saving the original image and each of the left and right captures, the next step is to split each image into its red, blue, and green components. The left image splits into red, green, and blue channels. However, the applied filter blocks red and green wavelengths, meaning the red channel corresponds to NIR reflectance, and the green channel does not contain any useful information. The right image splits into red, green, and blue channels, and each of these contain useful information for their corresponding sections of the spectrum. From there, the NDVI is calculated using the NIR reflectance values from the NoIR camera and the red reflectance values from the normal camera. A colormap is applied after performing a contrast stretch, both of which help make the output image easier to interpret. After the NDVI image has been saved, the Chlorophyll Index is calculated using NIR reflectance values from the NoIR camera module and the green reflectance values from the normal camera module. The minus one from the Chlorophyll Index formula was excluded, as this made it more straight forward to process and display the image. As with the NDVI, the Chlorophyll Index image is saved after performing a contrast stretch and applying a colormap. See Appendix A for the full program.

The drone was constructed using off the shelf components that were either found in the Computer Engineering Lab or purchased online. The drone consists of a frame, four motors, four electronic speed controllers (ESCs), a GPS module, a flight controller, a radio receiver, a telemetry transceiver, and a battery. The flight controller is a Pixhawk PX4, which was chosen because it supports an open-source flight control firmware called ArduPilot. ArduPilot has built in autonomous capabilities, and the ability to set missions for the drone to fly autonomously. Ardupilot provides stabilization, which is important for clear images, and supports a large number of sensors and fail safes. Additionally, with the use of a pair of transceivers, the drone can be monitored from a ground station, and its current objective can be updated at any time. Power for the Raspberry Pi and camera modules is provided by a regulated 5-volt output of the power distribution board mounted underneath the top plate of the drone frame.

IV. Testing & Results

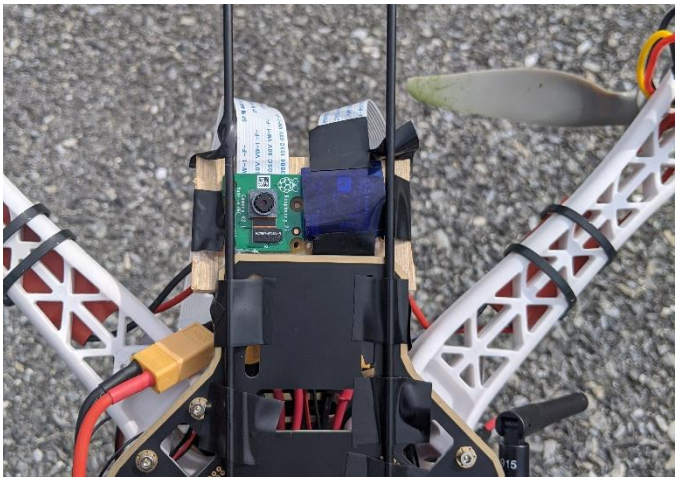
A. Camera

The camera functionality was continually tested throughout the programming process. Once the program was completed, the camera system was initially tested by capturing and displaying images taken from a stationary position on the ground. After these preliminary tests, the camera was mounted on the drone and tested once more from the airborne, moving platform.



The StereoPi board and camera mount attached to the drone frame.

Due to limited time and resources caused by the Coronavirus and the subsequent suspension of in-person classes and access to the college's campus, the camera system was mounted to the drone using tape and hot glue, and the camera mounting board was constructed of plywood. The original intention was to 3D print a mounting mechanism, but this was not possible without access to the robotics lab on campus.



Camera module mounting mechanism, with narrow band filter on NoIR camera module. Viewed from bottom of drone.

B. Drone

After the drone had been fully constructed, it was flight tested. It was able to be controlled manually or flown autonomously and had a maximum flight time of around 15 minutes. The transceiver and ground station also functioned

correctly, allowing for the pilot to see real time telemetry of the drone like GPS position, altitude, and battery level.

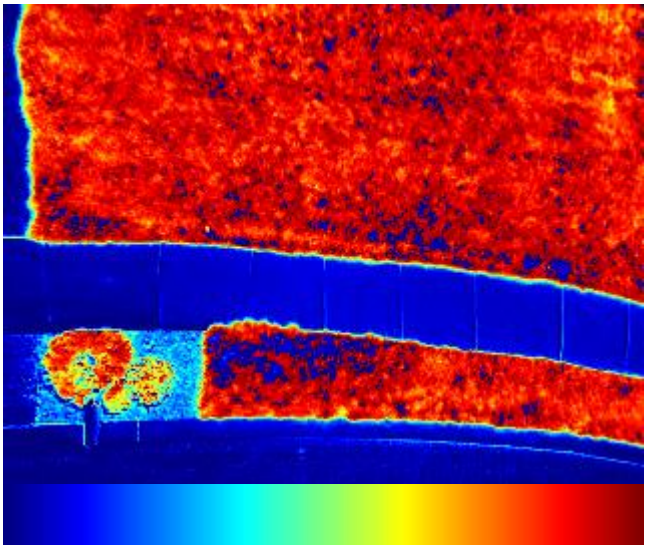
The drone in flight during a test flight.

C. Results

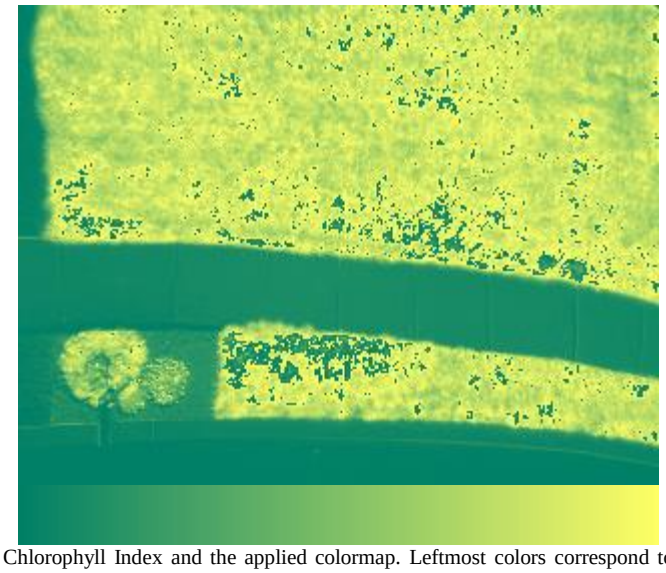
The drone and camera system were able to successfully operate simultaneously and allow for the capture of NDVI and CI images. From one of the test flights:

Left: Raw image from NoIR camera module with blue filter. Right: Raw image from normal camera module.

From the raw images that were captured, the NDVI and Chlorophyll Index were calculated:



NDVI and the applied colormap. Dark blue corresponds to inorganic matter (dirt, concrete, mulch, etc.). Colors to the right correspond to higher concentrations of photosynthetic activity.



Chlorophyll Index and the applied colormap. Leftmost colors correspond to inorganic matter (dirt, concrete, mulch, etc.). Colors to the right correspond to higher concentrations of chlorophyll.

D. Discussion

While some of the original goals of the project were not met, like offloading the image processing from the Pi, this project was a success. The camera and drone allow for the capture and interpretation of data for large pieces of land with regards to both NDVI and Chlorophyll Index. Time constraints as well as realizations about some of the goals led to differences in the final product from what was proposed. The offloading of the image processing ended up being unnecessary and overly time consuming; the Raspberry Pi Compute Module is more than powerful and efficient enough to process the images in real time while in flight without effecting performance, and this ends up being more convenient, as the images are ready to be viewed immediately upon landing.

Significant advances were made over the previous version of this project with the addition of a second camera sensor, which allowed for true NDVI calculation, and not just an approximation of the index, as well as the ability to calculate the Chlorophyll Index. The use of a drone also significantly improves the functionality of the system. While the final product of the project was different than initially intended, and less polished than hoped for, it represents a significant advancement from the previous version, and was successful in achieving the overall goals for functionality.

REFERENCES

[1] "Vegetation Indicies for Chlorophyll | CI – MTCI – NDRE – ND705 – ND550 – mNDblue," hiphen-plant.com. http://www.hiphen-plant.com/blog/vegetation-indices-2/ [Accessed: Sep. 20, 2019].

[2] J. Whitmarsh, and Govindjee, "The Photosynthetic Process," *Concepts in Photobiology: Photosynthesis and Photomorphogenesis*, 11-51, Jan. 1999. [Online]. Available: http://www.life.illinois.edu/govindjee/paper/gov.html. [Accessed: April 20, 2019].

[3] J. A. Gamon, J. Peñuelas, and C. B. Field, "A Narrow-Waveband Spectral Index That Tracks Diurnal Changes in Photosynthetic Efficiency," *Remote Sensing of Environment*, 41, 35-44, Feb. 1992. [Online]. Available: https://www.creaf.uab.es/global-ecology/Pdfs_UEG/RemSensEnv1992.pdf. [Accessed: April 15, 2019].

[4] X. Jin, C. Shi, C. Yu, T. Yamada, and E. Sacks, "Determination of Leaf Water Content by Visible and Near-Infrared Spectrometry and Multivariate Calibration in Miscanthus," *Frontiers in Plant Science*, May 2017, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5437372/. [Accessed: Apr. 28, 2019].

[5] J. Qi, Y. Kerr, M. Moran, M. Weltz, A. Huete, S. Sorooshiam, and R. Bryant, "Leaf Area Index Estimates using Remotely Sensed Data and BRDF Models in a Semiarid Region," *Remote Sensing of Environment*, 73, 18-30, Sep. 2000. [Online]. Available: https://s3.amazonaws.com/academia.edu.documents/41457075/Leaf_Area_Index_Estimates_Using_Remotely20160122-21992-113qvtz.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1556667216&Signature=rVv3dGJRZnSeLjzxpLQ7rT9rfAg%3D&response-content-disposition=inline%3B%20filename%3DLeaf_Area_Index_Estimates_Using_Remotely.pdf. [Accessed: Apr. 29, 2019

Appendix A: Real Time Code

```
import os
import time
from datetime import datetime
import picamera
from picamera import PiCamera
import cv2
import numpy as np

def contrast_stretch(im):
        in_min = np.percentile(im, 5)
        in_max = np.percentile(im, 95)
        out_min = 0.0
        out_max = 255

        out = im - in_min
        out *= ((out_min - out_max)/(in_min - in_max))
        out += in_min

        return out

# ettings
countdown = 1                   # Interval between image captures (seconds)
cam_width = 1280                # Cam sensor width
cam_height = 480                # Cam sensor height

scale_ratio = 0.5

# Camera resolution height must be dividable by 16, and width by 32
cam_width = int((cam_width+31)/32)*32
cam_height = int((cam_height+15)/16)*16
print ("Used camera resolution: "+str(cam_width)+" x "+str(cam_height))

# Buffer for captured image settings
img_width = int (cam_width * scale_ratio)
img_height = int (cam_height * scale_ratio)
capture = np.zeros((img_height, img_width, 4), dtype=np.uint8)
print ("Scaled image resolution: "+str(img_width)+" x "+str(img_height))

# Initialize camera
camera = PiCamera(stereo_mode='side-by-side', stereo_decimate=False)
camera.resolution=(cam_width, cam_height)
camera.framerate = 20
camera.hflip = False

counter = 0
t2 = datetime.now()
print ("Starting photo sequence")
for frame in camera.capture_continuous(capture, format="bgra", use_video_port=True,
resize=(img_width, img_height)):
        t1 = datetime.now()
        cntdwn_timer = countdown - int((t1 - t2).total_seconds())

        # If cowntdown is zero -> record next image
        if cntdwn_timer == -1:
                counter += 1
```

```
            # Save raw image
            cv2.imwrite('raw' + str(counter) + '.png', frame)

            # Split image to left and right
            imgLeft = frame[0:240, 0:320]
            imgRight = frame[0:240, 320:640]
            leftName = str(counter) + 'nir.png'
            rightName = str(counter) + 'r.png')
            cv2.imwrite(leftName, imgLeft)
            cv2.imwrite(rightName, imgRight)

            # Split left & right into components
            b, g, nir, a = cv2.split(imgLeft)
            b1, g1, r, a1 = cv2.split(imgRight)

            # Calculate NDVI
            bottom = (nir.astype(float) + r.astype(float))
            bottom[bottom == 0] = 0.01
            ndvi = (nir.astype(float) - r) / bottom
            ndvi = contrast_stretch(ndvi)
            ndvi = ndvi.astype(np.uint8)
            ndvi = cv2.applyColorMap(ndvi, cv2.COLORMAP_JET)
            cv2.imwrite('ndvi' + str(counter) + '.png', ndvi)

            # Calculate Chlorophyll Index
            b1 = g1.astype(float)
            b1[b1 == 0] = 0.01
            ci = nir.astype(float) / b1
            ci = contrast_stretch(ci)
            ci = ci.astype(np.uint8)
            ci = cv2.applyColorMap(ci, cv2.COLORMAP_SUMMER)
            cv2.imwrite('ci' + str(counter) + '.png', ci)

            t2 = datetime.now()
            time.sleep(1)
            cntdwn_timer = 0
            next

    key = cv2.waitKey(1)

    # Press q to quit
    if (key == ord("q")):
            break

    print("Photo sequence finished")
```