2018

CLASSIFYING SOIL MOISTURE CONTENT USING REFLECTANCE-BASED REMOTE SENSING

Ali Hamidisepehr
University of Kentucky, ali.hamidisepehr@gmail.com
Author ORCID Identifier:
https://orcid.org/0000-0001-9635-197X

Digital Object Identifier: https://doi.org/10.13023/etd.2018.366

Ali Hamidisepehr, Student

Dr. Michael P. Sama, Major Professor

Dr. Donald G. Colliver, Director of Graduate Studies

CLASSIFYING SOIL MOISTURE CONTENT USING REFLECTANCE-BASED REMOTE SENSING

DISSERTATION

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Colleges of Agriculture and Engineering at the University of Kentucky

By
Ali Hamidisepehr

Lexington, Kentucky

Director: Dr. Michael P. Sama, Assistant Professor of Biosystems & Agricultural Engineering

Lexington, Kentucky

2018

ABSTRACT OF DISSERTATION

CLASSIFYING SOIL MOISTURE CONTENT USING REFLECTANCE-BASED REMOTE SENSING

The ability to quantify soil moisture spatial variability and its temporal dynamics over entire fields through direct soil observations using remote sensing will improve early detection of water stress before crop physiological or economic damage has occurred, and it will contribute to the identification of zones within a field in which soil water is depleted faster than in other zones of a field.

The overarching objective of this research is to develop tools and methods for remotely estimating soil moisture variability in agricultural crop production. Index-based and machine learning methods were deployed for processing hyperspectral data collected from moisture-controlled samples.

In the first of five studies described in this dissertation, the feasibility of using "low-cost" index-based multispectral reflectance sensing for remotely delineating soil moisture content from direct soil and crop residue measurements using down-sampled spectral data were determined. The relative reflectance from soil and wheat stalk residue were measured using visible and near-infrared spectrometers. The optimal pair of wavelengths was chosen using a script to create an index for estimating soil and wheat stalk residue moisture levels. Wavelengths were selected to maximize the slope of the linear index function (i.e., sensitivity to moisture) and either maximize the coefficient of determination (R^2) or minimize the root mean squared error (RMSE) of the index. Results showed that wavelengths centered near 1300 nm and 1500 nm, within the range of 400 to 1700 nm, produced the best index for individual samples; however, this index worked poorly on estimating stalk residue moisture.

In the second of five studies, 20 machine learning algorithms were applied to full spectral datasets for moisture prediction and comparing them to the index-based method from the previous objective. Cubic support vector machine (SVM) and ensemble bagged trees methods produced the highest composite prediction accuracies of 96% and 93% for silt-loam soil samples, and 86% and 93% for wheat stalk residue samples, respectively. Prediction accuracy using the index-based method was 86% for silt-loam soil and 30% for wheat stalk residue.

In the third study, a spectral measurement platform capable of being deployed on a UAS was developed for future use in quantifying and delineating moisture zones within agricultural landscapes. A series of portable spectrometers covering ultraviolet (UV), visible (VIS), and near-infrared (NIR) wavelengths were instrumented using a Raspberry Pi embedded computer that was programmed to interface with the UAS autopilot for autonomous reflectance data acquisition. A similar ground-based system was developed to keep track of ambient light during reflectance target measurement. The systems were tested under varying ambient light conditions during the 2017 Great American Eclipse.

In the fourth study, the data acquisition system from the third study was deployed for recognizing different targets in the grayscale range using machine learning methods and under ambient light conditions. In this study, a dynamic method was applied to update integration time on spectrometers to optimize sensitivity of the instruments. It was found that by adjusting the integration time on each spectrometer such that a maximum intensity across all wavelengths was reached, the targets could be recognized simply based on the reflectance measurements with no need of a separate ambient light measurement.

Finally, in the fifth study, the same data acquisition system and variable integration time method were used for estimating soil moisture under ambient light condition. Among 22 machine learning algorithms, linear and quadratic discriminant analysis achieved the maximum prediction accuracy.

A UAS-deployable hyperspectral data acquisition system containing three portable spectrometers and an embedded computer was developed to classify moisture content from spectral data. Partial least squares regression and machine learning algorithms were shown to be effective to generate predictive models for classifying soil moisture.

KEYWORDS: Remote sensing, Spectroscopy, Soil moisture, Machine learning, Unmanned aircraft system, Ambient light calibration.

Ali Hamidisepehr

August 23, 2018

CLASSIFYING SOIL MOISTURE CONTENT USING REFLECTANCE-BASED REMOTE SENSING

By

Ali Hamidisepehr

Dr. Michael P. Sama

Director of Thesis

Dr. Donald G. Colliver

Director of Graduate Studies

August 23, 2018

To my parents, I could not have done it without you. Thanks for all the unconditional love and support along the way.

ACKNOWLEDGMENTS

I would like to extend my gratitude to the several individuals. I have benefitted from their help and support along the way to this milestone. First and foremost, I would like to thank my advisor, Dr. Michael Sama, for his support, advice, and mentorship over the course of my PhD program. During this period, I learned a lot of technical and research skills from him. Dr. Sama was a wonderful teacher and friend. Without his assistance I could have not accomplished my goal with the current quality. I also appreciate my advisory committee members, Dr. Ole Wendroth, Dr. Michael Montross, and Dr. Joseph Dvorak, for their insights and comments to improve this research.

To my fellow graduate students, thanks for the friendship that made my research work memorable. Specifically, thanks to Aaron Turner, Saket Dasika, Chris Good, and Felipe Pampolini for their support in conducting this research.

Finally, I would like to thank my family for their loving support and encouragement. I am deeply indebted to my parents for their unconditional and continual love and support throughout my career and entire life. To my sister, Fatemeh, and my brother-in-law, Salman. I appreciate all your assistance especially during the last three years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Agricultural irrigation management is increasingly becoming a vital factor to supply enough food to a growing population. Irrigation – as the primary fresh water consumer – has a large influence on water shortage issues (Gleick, 2003).The development of irrigation and nutrient management practices for food production has resulted in substantial increases in crop yield, accounting for over 80% of the gains in the global supply of wheat, rice, and corn since the 1960s (Cassman, 1999). Nearly 23 million hectares of land were irrigated in the U.S. during 2012, accounting for 31% of total U.S. freshwater use (USDA, 2015).

Novel technologies, such as variable-rate irrigation, help to control water usage and result in more efficient irrigation than traditional methods (O'Shaughnessy et al., 2015; Yari et al., 2017). To spatially implement variable-rate irrigation, a prescription map containing information about the actual water status of the field is needed (Buck et al., 2016). Many of the smart irrigation systems available for scheduling water application rely on either soil water holding capacity maps or low spatial resolution sub-soil sensor networks (Yule et al., 2008). Both methods may not be optimized, particularly in instances where the sensing technology is not spatially matched with the application technology. Increasing the spatial resolution of intensive management practices requires the optimization of inputs and can reduce the overall level of inputs required to produce the same output (Raun et al., 2002). High spatial resolution methods (10 m grid or smaller) for identifying water stress typically involve the use of remote sensing of a crop canopy using combinations of visible and near-infrared (Peñuelas et al., 1997) or thermal infrared sensing (Carlson et al., 1981; Nemani & Running, 1989). Traditional deployments include satellite, conventional aircrafts, and ground-based sensors but are limited in terms of cost, temporal, and spatial resolution.

Remote sensing is useful for obtaining field-scale and site-specific information about the drought status in a field and generating near real-time irrigation prescription maps. Reflectance-based remote sensing is a potential method for quantifying soil moisture and delineating moisture management zones. Several studies have focused on using visible, multispectral, or hyperspectral cameras mounted on drones to evaluate crop/soil status at high spatial resolutions. This technology has received much attention in the past few decades for identifying the spatial variability (or behavior) ofwater stress in agricultural applications (Atzberger, 2013; Bernardes et al., 2012; Carlson et al., 1981; Doraiswamy et al., 2005; Nemani & Running, 1989; Peñuelas et al., 1997; Thenkabail et al., 2014). In many applications, the crop was the visual target for indirectly measuring soil or crop parameters. However, there may still be useful information available from direct soil reflectance measurements. The ability to quantify soil moisture variability and its temporal dynamics over entire fields through direct soil observations using remote sensing will improve early detection of water stress before crop physiological or economic damage has occurred, and it will contribute to the identification of zones within a field where soil water is depleted faster than in other zones.

A common way of applying remote sensing in a field is to select a couple of narrow-band ranges of wavelengths with the potential to provide a sufficiently accurate estimation of a field parameter based on reflectance values. Combinations of these narrow-band ranges are used to compute indices that are correlated with crop and soil parameters. For instance, the normalized difference vegetation index (NDVI), which is typically a combination of red and near-infrared bands, is one of the most ubiquitous remote sensing indexes in agriculture for predicting field parameters correlated to crop vigor. The normalized difference water index (NDWI), which typically replaces one of the NDVI bands with a water-absorption band in the near-infrared range, has been shown to be a better performing index than NDVI for estimating water stress (Gao, 1996; Gu et al., 2007).

A limited number of narrow-band wavelengths are used for index generation to reduce sensor cost and complexity. However, by relying on only one or two wavelengths for parameter estimation, information which can be extracted from other wavelengths are ignored. On the processing end, building the model to extract information from hyperspectral data is computationally intense, but it is not necessarily cost prohibitive and it has little bearing on cost once the model for estimating a crop or soil parameter has been developed.

In recent years, new approaches and algorithms (e.g. machine learning algorithms) have been developed and are well suited for handling big datasets with many input variables. Learning algorithms are regularly used in daily life, frequently without being noticed. Learning algorithms try to mimic how the human brain learns by recognizing patterns and rules in a dataset (Jensen et al., 1999). A computer is given a dataset containing a large number of input variables and samples. The response variable value for each set of variables and samples is also known. In this way, a learning algorithm tries to “understand” how a set of inputs produces a specific output. By recognizing patterns in a training dataset, an algorithm is trained and can be used to classify new samples. An advantage of this method is that it enables computers to be trained by learning from experiences without being explicitly programmed using an analytical model or simple empirical model. The algorithm performance generally increases by experiencing more samples, assuming the samples accurately represent the modeled process.

Machine learning is a widely-used technology, and in the last decade, it has been applied to remotely sensed data in the agricultural domain. Specific examples include drought assessment using MODIS and AVIRIS satellite data (Park et al., 2016; Trombetti et al., 2008), forecasting vegetation health using MODIS satellite data (Nay, Burchfield, & Gilliganc, 2016), estimating LAI index using MODIS and CYCLOPS (Verger et al., 2008) and Landsat ETM+ satellite data (Walthall et al., 2004), weed detection using manual RGB imagery (visible light imagery comprised of red, green, and blue pixels) on the ground (Cho et al., 2002; Jafari et al., 2006) and multispectral and RGB imagery on a UAS (Koot, 2014), and plant classification using hyperspectral and RGB images (Moreno et al., 2014).

Traditional deployments of remote sensing include satellite and conventional aircraft, but are limited in terms of cost, temporal resolution, and spatial resolution. Unmanned aircraft systems (UAS), or drones, are relatively new tools for collecting remote sensing data in agricultural applications (Chrétien et al., 2015; Pádua et al., 2017). In one study, traditional methods of remote sensing, including satellites and manned aircraft systems, were compared to a UAS method. It was concluded that UAS were more cost-effective in small fields and UAS were shown to have the potential to provide higher spatial precision data (Matese et al., 2015).

Spectral reflectance data collected using UAS are increasingly used in research to estimate different soil and crop parameters. Example applications of UAS-based remote sensing in the agricultural domain include estimating soil fertility (Bajwa & Tian, 2005), generating vegetation indices (Berni et al., 2009; Candiago et al., 2015), assessing tree crowns for breeding applications (Díaz-Varela et al., 2015), yield estimation (Geipel et al., 2014), plant classification (Hung et al., 2014), weed detection (Koot, 2014), and controlling herbicide applications (Xiang & Tian, 2011).

Advances in spectrometer development have led to more portable systems, or micro-spectrometers, that are particularly suitable for UAS deployment due their small size and mass. One study showed that measurements from UAS-deployed micro-spectrometers were highly correlated with parameters measured at ground level and concluded that the UAS platform could provide a faster method for spectral data collection compared to traditional remote sensing methods (Burkart et al., 2014). Nevertheless, calibrating these sensors for various ambient light conditions and avoiding saturation are challenges needed to be dealt with. Field spectrometers are mostly limited to data collection in a specific period and ambient light condition (Damm et al., 2011; Gao et al., 2002, 2004; Guanter et al., 2006). Using reference tarps is another common approach for compensating against ambient light changes. A measurement from tarps needs to be taken for each measurement from a land target to continuously keep track of ambient light changes (Shanahan et al., 2001). But since it is practically difficult, especially for large field scales, only a few measurements from tarps can be taken during the data collection process. An alternative method is to use a second sensor for measuring ambient light spectra while measuring reflectance (Burkart et al., 2014; Von Bueren et al., 2015).

In spite of all efforts have been made on this area, a low-cost hyperspectral system for estimating soil moisture at different ambient light condition is desired. This system should provide a highly accurate prediction model while the model is kept computationally simple for a common processor. Hence, developing a hyperspectral data acquisition system coupled with a data processing method for estimating soil moisture from a UAS platform and under ambient light conditions will lead to more efficient field monitoring and irrigation management.

1.1 Project Objectives

The overarching objective of this research is to develop tools and methods for remotely estimating soil moisture variability at a pilot scale in agricultural crop production. The proposed study aims to integrate spectral data collected using a UAS-deployed spectrometer with ground reference sampling to determine the ability to predict soil moisture measurements. Specific objectives include to:

1. Determine the feasibility of using "low-cost" index-based multispectral reflectance sensing for remotely delineating soil water content from direct soil and crop residue measurements from down-sampled spectral data.
2. Apply machine learning algorithms to full spectral datasets for moisture estimation and comparing to index-based method from previous objective.
3. Instrument a series of portable spectrometers and integrate into an unmanned aircraft system for autonomous data collection.
4. Recognize different reflectance targets in the greyscale range by compensating against ambient light changes using variable integration time during data collection and machine learning for post processing.
5. Testing the capability of the data acquisition system and predictive models generated by machine learning algorithms to estimate soil moisture under ambient light condition.

1.2 Organization of Thesis

This dissertation is organized in seven chapters. Chapter 1 establishes a broad background information and the general rationale of this research along with specific research objectives. Chapter 2 describes the feasibility of using a customized index-based multispectral reflectance sensing for remotely delineating soil and crop residue water content. Chapter 3 provides a comparison between multiple machine learning methods and an index-based estimation from the hyperspectral dataset collected under controlled-light conditions. Chapter 4 describes the process of instrumenting a series of portable spectrometers and integrating it into an unmanned aircraft system for autonomous data collection. Also, the system evaluation during the 2017 Great American Eclipse is included in this chapter. Chapter 5 describes a study for testing an approach with constant changing of integration time of reflectance spectrometers for recognizing reflectance targets painted with different colors in grayscale range. Also, it shows the optimal method for calibrating reflectance measurements. Chapter 6 takes results from the previous study into account and demonstrates the capability of the data acquisition system in estimating soil moisture under ambient light conditions. Also, prediction accuracy of models generated using multiple machine learning algorithms are evaluated. Chapter 7 concludes major findings from the present research and discusses the future work.

The research presented in this dissertation is either published or submitted for publication in peer-reviewed scientific journals:

1. Hamidisepehr, A., Sama, M. P., Turner, A. P., & Wendroth, O. O. (2017). A Method for Reflectance Index Wavelength Selection from Moisture-Controlled

Soil and Crop Residue Samples. Transactions of the ASABE, 60(5), 1479-1487. (Chapter 2)

2. Hamidisepehr, A., Sama, M. P. (2017). Moisture Content Classification of Soil and Stalk Residue Samples from Spectral Data using Machine Learning. Transactions of the ASABE, Under Review. (Chapter 3)

CHAPTER 2: OBJECTIVE 1: A METHOD FOR REFLECTANCE INDEX WAVELENGTH SELECTION FROM MOISTURE CONTROLLED SOIL AND CROP RESIDUE SAMPLES

2.1 SUMMARY

Reflectance indices are a method for reducing the dimensionality of spectral measurements used to quantify material properties. Choosing the optimal wavelengths for developing an index based upon a given material and property of interest is made difficult by the large number of wavelengths typically available to choose from and the lack of homogeneity when remotely sensing agricultural materials. This study aimed to determine the feasibility of using a low-cost method for sensing the water content of background materials in traditional crop remote sensing. Moisture controlled soil and wheat stalk residue samples were measured at varying heights using a reflectance probe connected to visible and near-infrared spectrometers. A program was written that used reflectance data to determine the optimal pair of narrowband wavelengths to calculate a normalized difference water index (NDWI). Wavelengths were selected to maximize the slope of the linear index function (i.e. sensitivity to moisture) and either maximize the coefficient of determination (R-squared) or minimize root mean squared error (RMSE) of the index. Results showed that wavelengths centered near 1300 nm and 1500 nm, within the range of 400 nm to 1700 nm, produced the best index for individual samples. Probe height above samples and moisture content were examined for statistical significance using the selected wavelengths. The effect of moisture was significant for both bare soil and wheat stalks, but probe height was only significant for wheat stalk samples. The index, when applied to all samples, performed well for soil samples but poorly for wheat stalk residue samples. Index

calculations from soil reflectance measurements was highly linear (R2 > 0.95) and exhibited small variability between samples at a given moisture content, regardless of probe height. Index calculations from wheat stalk residue reflectance measurements were highly variable, which limited the usefulness of the index for this material. Based on these results, it is expected that crop residues, such as wheat stalk residue, will reduce the accuracy of remotely sensed soil surface moisture measurements.

2.2 INTRODUCTION

The development of irrigation and nutrient management practices for food production has resulted in substantial increases in crop yield, accounting for over 80% of the gains in global supply of wheat, rice, and corn since the 1960's (Cassman, 1999). While this development has limited the expansion of agricultural land, it has also resulted in a reduction in biodiversity (Cardinale et al., 2012) and placed a large burden on global water resources (Hatfield, 2015). Nearly 23 million hectares of land were irrigated in the United States during 2012, accounting for 31% of total U.S. freshwater use (USDA, 2015). Many of the smart irrigation systems available for scheduling water application rely on either soil water holding capacity maps or low spatial resolution sub-soil sensor networks (Yule et al., 2008). Both methods may not be optimized – particularly in instances where the sensing technology is not spatially matched with application technology. Increasing the spatial-resolution of intensive management practices can help with optimizing inputs and reducing the overall level of inputs required to produce the same output (Raun et al., 2002).

High spatial resolution methods (10 m grid or smaller) for identifying water stress typically involve the use of remote sensing of a crop canopy using combinations of visible and near-infrared light (Penuelas et al., 1997), or thermal infrared sensing (Carlson et al., 1981; Nemani & Running, 1989). Traditional deployments include satellite, conventional aircraft, and ground-based sensors but are limited in terms of cost, temporal and spatial resolution. Perhaps the most successful adoption of remote sensing technology in production agriculture has been the use of the normalized difference vegetation index (NDVI) to detect crop vigor, which is then correlated to a myriad of parameters in addition to water stress. These include vegetation cover (Carlson & Ripley, 1997), crop nitrogen status (Solari et al., 2008), crop yield (Benedetti & Rossini, 1993), and phenotype (Svensgaard et al., 2014). An alternative to NDVI is the normalized difference water index (NDWI), which typically uses longer wavelengths of light beyond the sensitivity of silicone-based photodiodes (Gao, 1996) and is potentially better suited to identifying crop water stress (Gu et al., 2007).

Two challenges for remotely sensing crop water stress using traditional methods are the absorption light due to atmospheric moisture and the contribution of soil reflectance on the overall vegetation reflectance spectra. Active ground-based sensors have been shown to overcome atmospheric limitations in nitrogen sensing by providing a light source (Holland et al., 2004; Mullen et al., 2003; Raun et al., 2002). The effect of soil type and conditions on canopy reflectance indices has also been addressed through calibrated indices, such as the soil adjusted vegetation index (SAVI) (Huete, 1988) or by removing the soil contribution from the reflectance spectra (Huang et al., 2009). In all the aforementioned applications, the crop was the visual target for indirectly measuring soil or crop parameters. However, there may still be useful information available from direct soil reflectance measurements. This work aims to study the reflectance spectra of bare soil and crop residue to determine if they can contribute to crop water stress detection. The ability to quantify spatial soil moisture variability and its temporal dynamics over entire fields through direct soil observations using remote sensing will improve early detection of water stress before crop physiological or economic damage has occurred, and it will contribute to the identification of zones within a field in which soil water is depleted faster than in other zones of a field.

The main objective of this study was to determine the feasibility of developing a low-cost reflectance sensor for remotely delineating soil water content from a ground or low altitude UAS-deployed platform. Specific objectives include to:

1. Collect visible and near-infrared spectral response from moisture-controlled soil and crop residue samples.
2. Identify the optimal wavelengths for a normalized index based upon user-defined constraints.
3. Determine if the effect of height of the sensor above the sampled surface is statistically significant.

2.3 MATERIALS AND METHODS

2.3.1 Sample preparation

In this study, soil samples with pre-determined water contents were prepared for two materials: silt-loam soil and wheat residue stalks. These materials were chosen as they represent potential background materials when observing row crops, such as corn or soybean, at early growth stages. 120-mL plastic containers with air-tight removable lids were used to contain the moisture-controlled samples. The soil was air dried, ground and passed through a 2-mm sieve. The initial moisture content (wet basis) of the soil was determined gravimetrically by drying a sample in a convection oven at 105°C for over 24 h and measuring the resulting reduction in mass. The stalks were dried in a similar manner to the soil samples to prevent decomposition. Each container was marked at a depth of 35 mL (for bare soil) and 120 mL (for stalks) and filled to the mark and lightly tapped to firm up the soil/stalks. Seven moisture levels were chosen (air dry (Giada et al., 2003) or 0% (stalks), 5%, 10%, 15%, 20%, 25%, and 30%). The mass of soil/stalks inside each sample container was measured with the mass of the container removed and used to determine the required mass of water to reach the target moisture content. Water was added to each sample using a pipet with a volumetric precision of 0.01 ml and the final mass was recorded. Then the lid of each container was closed, and the samples were equilibrated over several days to let the water redistribute through the sample. Three replications were prepared for each moisture level to minimize the effect of sample preparation error on the result. In total, 21 soil samples and 21 stalk samples were prepared.

2.3.2 Instrumentation Hardware

Reflectance was measured using visible and near-infrared spectrometers (HR400-7-VIS-NIR, NIRQuest512; Ocean Optics; Dunedin, FL) with a tungsten-halogen light source (HL-2000-HP-FHSA; Ocean Optics; Dunedin, FL). A fiber optic reflectance probe (QR200-12-MIXED: Ocean Optics; Dunedin, FL) was used to transmit source light to the sample and reflected light to the spectrometers. The reflectance probe consisted of twelve 200 µm diameter transmission fibers spaced concentrically around two 200 µm diameter reflectance fibers and was 2 m in length. The spectrometers were calibrated to 0 and 100% reflectance by blocking the light source for the background measurement and using a Spectralon calibration target (WS-1-SL; Ocean Optics; Dunedin, FL) for the reference measurement, respectively. The effective spectral range was 400 to 1700 nm with an overlap at 900 nm between the two spectrometers.

A consistent method was needed to position the reflectance probe above each sample to minimize bias and reduce variability due to probe height. A reflectance test fixture (Figure 2-1) was designed and fabricated to consistently position the spectrometer reflectance probe above the sample surface. The fixture consisted of three main components that were 3D printed from black ABS plastic: a sample holder for centering the sample container underneath the probe, an outer probe mount that rested directly on top of the sample surface, and an inner probe mount for setting the height of the probe above the sample surface. The inner probe mount had stainless-steel dowel pins pressed into the sidewall that slid down guides in the outer probe mount. The height of the probe was set by rotating the inner probe mount inside the outer probe mount at one of five height index points. The probe heights were evenly spaced between 0.64 to 5.76 cm in increments of 1.28 cm. The 24.8° field-of-view (FOV) of the reflectance probe resulted in a sampling area of approximately 0.06 to 5 cm^2. The sampling diameters and areas for all heights are shown in Table 2-1. The maximum height was selected based on the reflectance probe FOV and the sample size to limit the side walls of the outer probe mount from affecting the reflectance measurement.

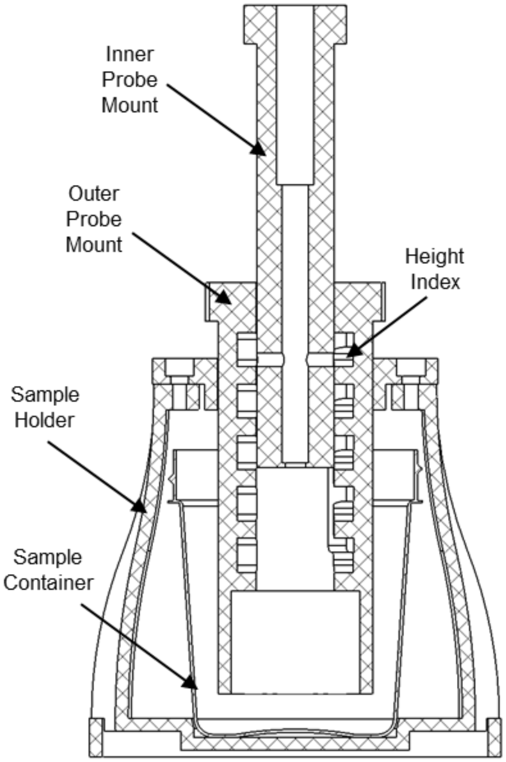


Figure 2-1: Cut-view of the reflectance test fixture used to position the spectrometer probe above a soil/stalk sample.

Table 2-1: Reflectance probe heights and resulting sampling diameters and areas

Height #	**Probe Height (cm)**	**Sampling Diameter (cm)**	**Sampling Area (cm^2)**
H1	0.64	0.28	0.06
H2	1.92	0.84	0.55
H3	3.20	1.40	1.54
H4	4.48	1.96	3.02
H5	5.76	2.52	4.99

2.3.3 Data Collection

OceanView (Version 1.4.1; Ocean Optics; Dunedin, FL) software was used to configure the spectrometers and record the reflectance response. A graphical program was written that calculated reflectance from each spectrometer, combined the two results into a single array, and graphed the results in real-time. The system was calibrated every time the height of the probe was changed and whenever the spectrometers and light source were powered on. The software was configured to record three measurements per sample. Given that there were three samples for each moisture content, three replications for each sample, and three measurements for each replication, there were 27 total reflectance responses for every combination of moisture content and height. This replication structure was intended to reduce the influence of variability in sample preparation, reflectance probe position, and sensor noise on the resulting index calculation.

Reflection measurements were normalized between 0 and 1 (0 and 100%) by subtracting the background measurement intensity from the raw measurement intensity and reference target measurement intensity, and taking the ratio of the resulting differences (Equation 1).

$$R_\lambda = \frac{M_\lambda - C_\lambda^0}{C_\lambda^1 - C_\lambda^0} \tag{1}$$

Where:

R was the normalized reflectance measurement from a sample (%)

M was the raw measurement intensity from a sample (A/D counts)

C^0 was the background measurement intensity with the light source obstructed (A/D counts)

C^1 was the reference target measurement intensity (A/D counts)

λ was the specific wavelength (nm)

Each reflectance measurement was stored in a tab-delimited text file containing the spectral response along with the spectrometer settings. A file naming scheme was used to label each text file to better facilitate post-processing. Filenames included: a sample code for identifying sensor height, replication and the sample container; a string corresponding to the data type within OceanView; and a local timestamp (Figure 2-2). A MATLAB script (R2015b; The Mathworks; Natick, MA) was written to access all text files from a single folder and categorized them using the filename sample code. The script stored data as columns in a single Excel spreadsheet with the corresponding sample codes as headers in the first row of each column.

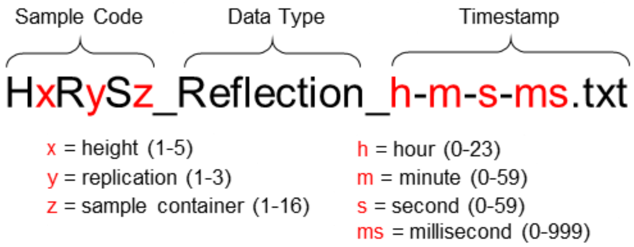


Figure 2-2: Filename format for output text files of the spectrometers. Fixed values are shown in black and variables are shown in red.

2.3.4 *Data Analysis*

A second MATLAB script was written to perform data analysis. The script read in the entire dataset, calculated normalized indices for all pairs of wavelengths, and identified the "best" pair based upon user defined criteria (139). The normalized index was composed of two distinct narrowband ranges identified by their center wavelengths and obtained for every possible pair in ascending order over the 400 to 1700 nm range (Equation 2).

$$I_{\lambda 1,\lambda 2} = \frac{R_{\lambda 1} - R_{\lambda 2}}{R_{\lambda 1} + R_{\lambda 2}} \qquad (2)$$

Where:

$I_{\lambda 1,\lambda 2}$ = normalized index for wavelengths centered at $\lambda 1$ and $\lambda 2$ (-1 to 1)

$R_{\lambda 1}$, $R_{\lambda 2}$ = average reflectance at wavelengths centered at $\lambda 1$ and $\lambda 2$ (%)

$\lambda 1 > \lambda 2$ to reduce the number of computations by a factor of two

Selecting the "best" pair of wavelengths for calculating a reflectance index to predict moisture content implied several assumptions and required constraints to simplify the optimization process. It was assumed that the low-cost sensor would use either a silicon or indium-gallium-arsenide (InGaAs) photodetector coupled with narrow-band filters to detect specific wavelengths of visible and NIR light. For this study, the bandwidths were set to ± 25 nm and assumed to be uniformly distributed about a center wavelength. Preliminary reflectance index calculations using manually-selected wavelengths revealed a linear relationship between sample moisture content and the normalized index. Moreover, sensor height above the sample had little effect on index values. Therefore, a linear regression model was used to estimate moisture content based on the average normalized index measurement. Three optimization parameters were initially chosen: the coefficient of determination (R-Squared) of the linear regression between moisture content and the reflectance index; the root mean squared error (RMSE) between the actual and predicted moisture content; and the slope of the linear regression, which represented sensitivity. The pair of wavelengths with the highest slope, the lowest RMSE, and the highest R-Squared were considered the optimal solution by maximizing sensitivity and minimizing error. The optimization parameters were stored for each normalized index calculation and plotted in the form of slope versus R-Squared and slope versus RMSE to determine if local optima or a global optimum existed.

A third script was written to determine the performance of the index for predicting moisture (134). The "best" wavelengths resulting from the previous step were used as inputs and the normalized index for all samples was computed. A statistical analysis was conducted to determine if probe height above the sample was statistically significant. The experiment was set up with a factorial design in moisture content and height (7×5) for bare soil. The data were subjected to variance analysis and appropriate means separation was conducted using statistical software (JMP 12; SAS; Cary, NC). The linear regression model from the average normalized index and the individual index values were used to determine a 95% prediction interval.

2.4 Results and Discussion

2.4.1 Spectrometer Calibration

The purpose of the calibration was to remove non-uniformity in spectral response due to the variability in the light source, optical fibers, and spectrometer detector with respect to wavelength. Figure 2-3 illustrates the raw intensity reference response from the spectrometers with the probe set to height H3 above the calibration target and the light source adjusted to maximize intensity without saturation at any wavelength of either spectrometer. The visible spectrometer always saturated before the NIR spectrometer and thus determined the intensity of the light source. Heights H4 and H5 used the full light source intensity and therefore did not use the full intensity range of either spectrometer. The other three heights produced similar responses that were scaled along the intensity axis. The intensity axis represented the raw analog-to-digital (A/D) converter result from the spectrometers' photodetectors. The visible spectrometer was a 14-bit resolution (0-16383) measurement and the NIR spectrometer was a 15-bit resolution (0-32767) measurement.

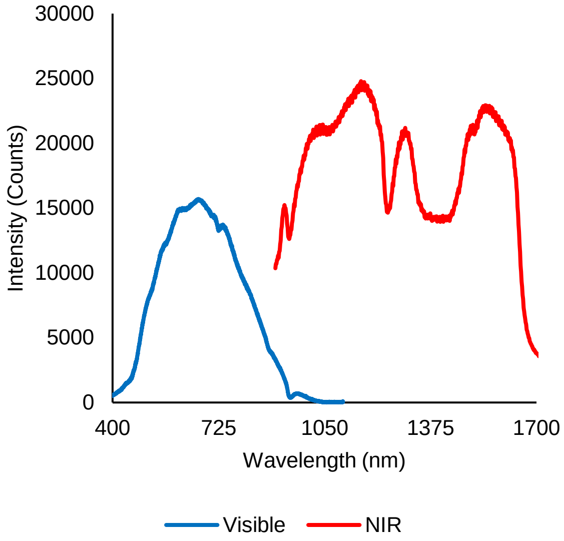


Figure 2-3: Intensity of reflected light versus wavelength for the reference measurement at height H3 (3.2 cm).

Figure 2-4 illustrates the raw intensity background response from the spectrometers when the light source was blocked. The small variations across wavelengths were due to noise in the spectrometer detector. The NIR spectrometer had a large offset from zero as compared to the visible spectrometer, which was due to operating in a high-gain mode. The high-gain mode was necessary to obtain a sufficient signal from the NIR spectrometer when using a single light source and reflectance probe.

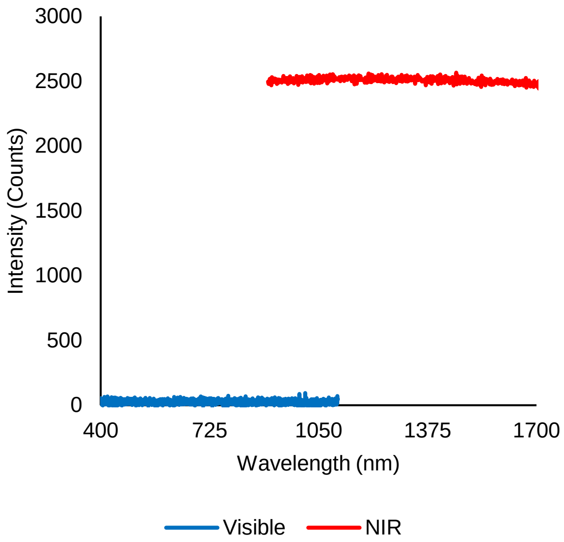


Figure 2-4: Intensity of reflected light versus wavelength for the background measurement at height H3 (3.2 cm).

Figure 2-5 shows the results of the calibration process where non-uniformity had been removed across all wavelengths when calculating background reflectance and the reflectance from the reference target. Data from both spectrometers were spliced into a single dataset by removing data from the visible spectrometer past 900 nm and combining it with all data from the NIR spectrometer. Note that wavelengths less than 500 nm still deviated from the desired 0 and 100% reflectance for the background and reference measurements, respectively. This was due to low relative sensitivity of the visible spectrometer below this wavelength and indicated that more noise should be expected when using wavelengths in this range to calculate indices.

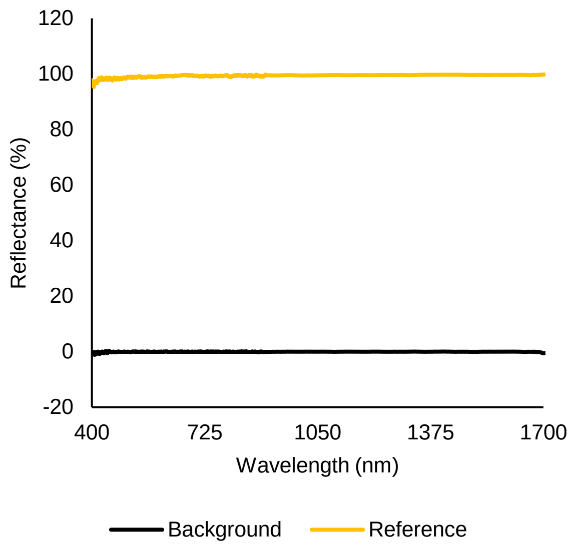


Figure 2-5: Calibrated and combined reflectance response for a background and reference measurement at height H3 (3.2 cm).

2.4.2 Bare Soil

Figure 2-6 shows the reflectance for varying water content of soil samples versus wavelength. Each series is the average of all samples at a particular moisture content across all heights. The general spectral response of the soil samples was an increase in relative reflectance as wavelength increased. Drier samples typically reflected more light on average, but there were instances where the average reflectance across all wavelengths was not in order. For example, both the 25% and 30% moisture content samples measured at height H3 reflected more light than the 20% moisture content sample. This phenomenon was likely caused by small variations in the distance between the measured area and the spectrometer probe. Despite efforts to control the exact distance with a reflectance probe test fixture, uncontrolled variations in the soil surface shape (i.e., flat, convex, concave) likely had a substantial impact on average reflectance. Given that the soil surface in the field could never be carefully controlled on the scale that was relevant to this experiment, no further adjustments to the sample were made. The non-ordered progression of average reflectance also illustrated why a two- or more-wavelength index was crucial for modeling the relationship between reflectance and moisture content. No single wavelength will produce a monotonic relationship with suitable sensitivity. However, it was observed that the relative dip in reflectance between 1400 nm to 1500 nm, when compared to other wavelengths for the same moisture content, exhibited a clear pattern. As moisture content increased, the relative reflectance inside this range tended to decrease while the rest of the spectral response followed a consistent profile.

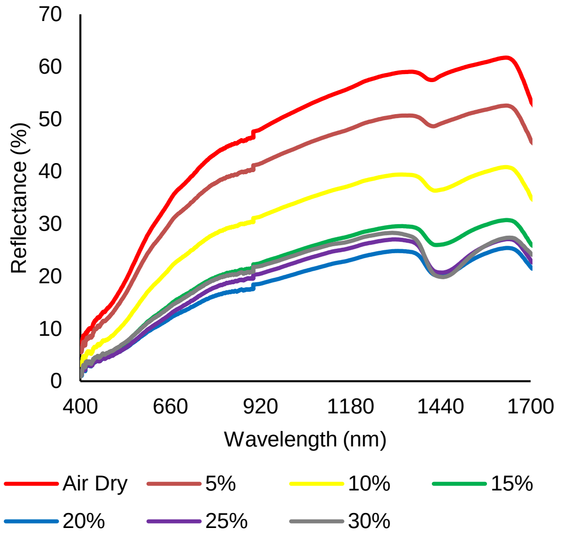


Figure 2-6: Average relative reflectance versus wavelength for varying nominal soil moisture contents in bare soil.

The transition between the visible and near-infrared spectrometers at 900 nm produced a noticeable artifact in the relative reflectance measurement. Increasing the number of spectrometer calibration points between 0% and 100% reflectance might have mitigated this non-linearity but a simpler solution was to ensure that wavelengths near this transition were not used when calculating an index.

As previously stated, the goal of the optimization process for selecting the “best” pair of wavelengths used to calculate a moisture content prediction index was to select the index that produced the largest slope while either maximizing the R-Squared or minimizing the RMSE of the index function. Without knowing the relationship between the constraints, it was difficult to prioritize one constraint over the other. Rather than arbitrarily weighting each constraint, the resulting relationship between all pairs of wavelengths was plotted for both slope versus R-Squared (Figure 2-7) and slope versus RMSE (Figure 2-8).

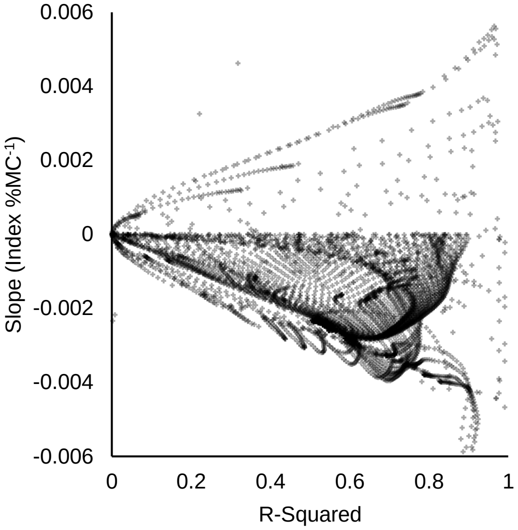


Figure 2-7: The slope of linear regression of reflectance and moisture content vs. R-Squared on bare soil

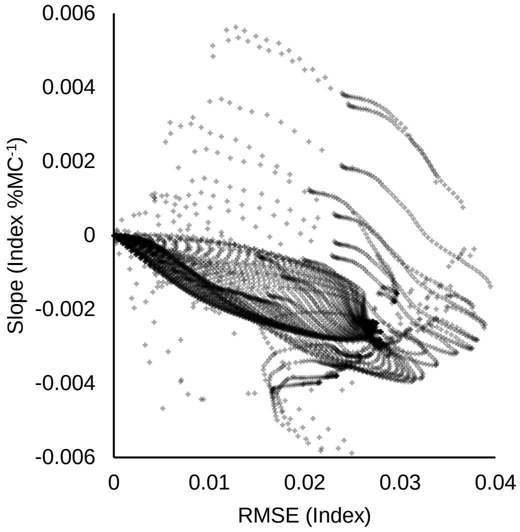


Figure 2-8: The slope of linear regression of reflectance and moisture content vs. RMSE on bare soil

The resulting shapes revealed if local or global optima existed and illustrated an interesting trend between slope and either R-Squared or RMSE. Points tended to follow deterministic paths as one wavelength was changed relative to another. The majority of slopes were negative, which was a result of the index equation in conjunction with the positive trend in relative reflectance. Longer wavelengths generally exhibited a larger relative reflectance than shorter wavelengths, which produced a negative term in the index numerator. The sign could be fixed positive by always assigning the higher relative reflectance wavelength to λ_1. Figure 2-7 also reveal why R-Square and RMSE alone were not adequate for selecting the appropriate index wavelengths. As RMSE decreased, so did the slope of the index, which reduced the sensitivity of the index to moisture content. Similarly, the wavelengths that resulted in the highest slope also had a RMSE very close to zero (Figure 2-8).

There was no global optimum when using RMSE but R-Squared did produce a grouping of indices where both the slope and R-Squared were close to their respective maxima. The two wavelengths that produced this relationship were centered near 1300 nm and 1500 nm. When using RMSE, a peak occurred at a slope of approximately 0.0058 Index %MC^{-1} and an RMSE of 0.013. The corresponding wavelengths for this index were also centered near 1300 nm and 1500 nm.

The index values from 50 nm wide bands centered at 1300 nm and 1500 nm for all samples are shown in Figure 2-9 along with the linear regression model and 95% prediction interval. Variability in the calculated index among samples at a given moisture content tended to increase as moisture content increased. Average index values varied from 0 to 0.15 for soil samples at 3.3% to 30% moisture content, respectively.

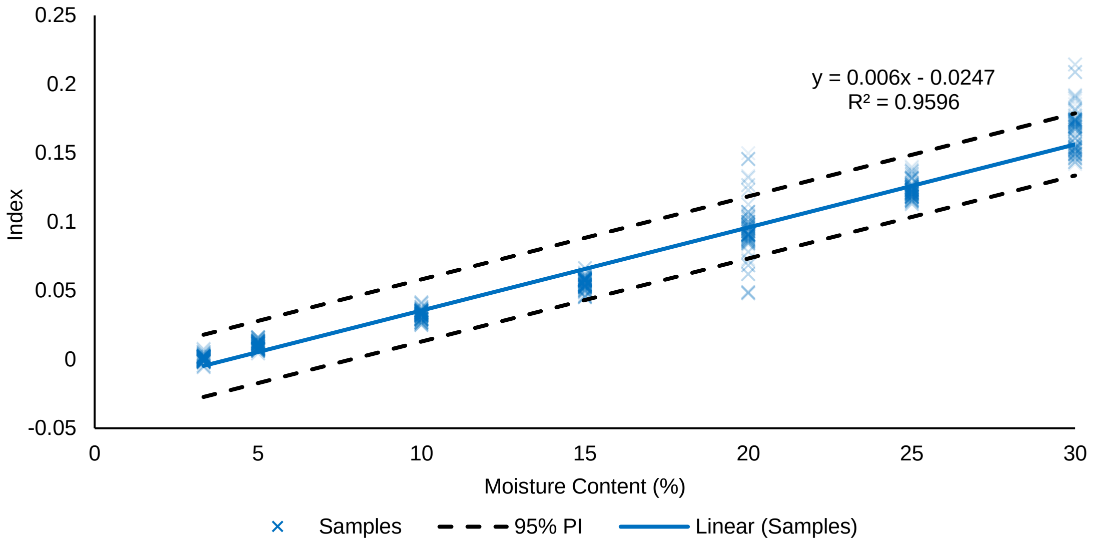


Figure 2-9: Normalized index for bare soil using 50 nm bands centered at 1300 and 1500 nm along with the linear regression and 95% prediction interval.

2.4.3 Wheat Stalk Residue

Wheat residue stalks produced a similar spectral response to bare soil, where the reflectance generally increased with respect to wavelength and a dip occurred between 1400 and 1600 nm (Figure 2-10). Both the discontinuities between the two spectrometers and the dip at the water absorption bands were more pronounced, while the average difference in reflectance between moisture contents was smaller. Again, the discontinuities could have been better addressed through a more complex calibration process but that was deemed unnecessary for this experiment. The average reflectance was not well correlated with moisture content and was likely driven by effective height of the stalk surface, which was less carefully controlled than the soil surface due to the physical structure of the stalks.

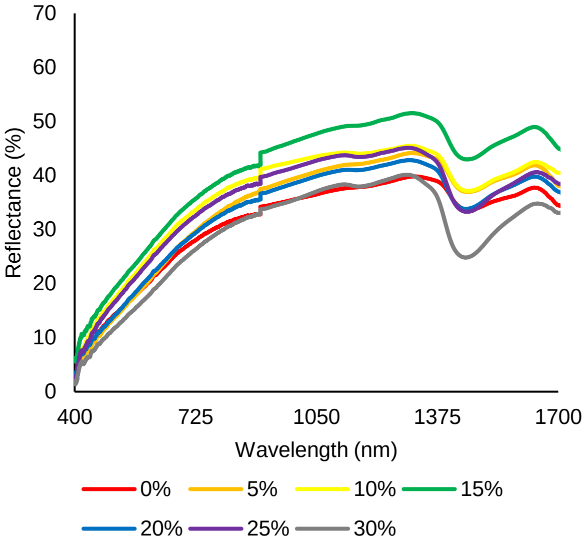


Figure 2-10: Average relative reflectance versus wavelength for varying nominal moisture of wheat residue stalks along with the linear regression and 95% prediction interval.

Since similar patterns between reflectance and wavelengths existed between bare soil and stalks, it was expected that the optimization process would provide a pair of “best” wavelengths close to the results of bare soil. Plots between R-square (Figure 2-11) and RMSE (Figure 2-12) versus slope revealed similar patterns as wavelengths were incrementally changed but the overall shapes differed from the results of bare soil. In both instances, optima occurred at smaller slopes and either lower R-squared or higher RMSE values, indicating that the index would not likely perform as well as it did for bare soil. However, the local optima still corresponded to the same pair of wavelengths near 1300 nm and 1500 nm, which indicated that the same sensor may function, albeit less accurately, in areas that include both bare soil and wheat stalk residue. A single pair of wavelengths across a variety of soil and crop material compositions would be advantageous for applying a low-cost sensor across varying commodities and production practices.

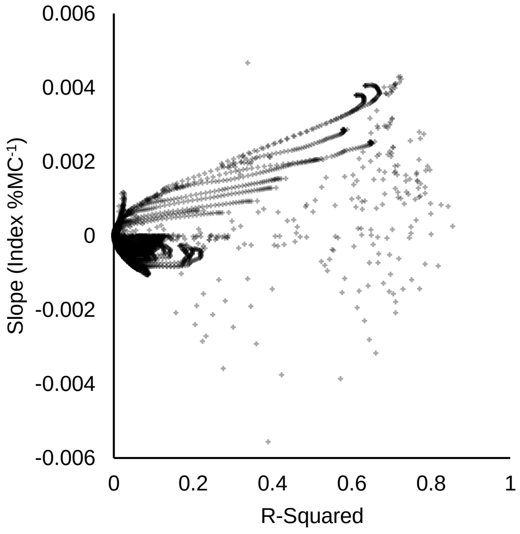


Figure 2-11: The slope of linear regression of reflectance and moisture content versus R-squared on residue stalks

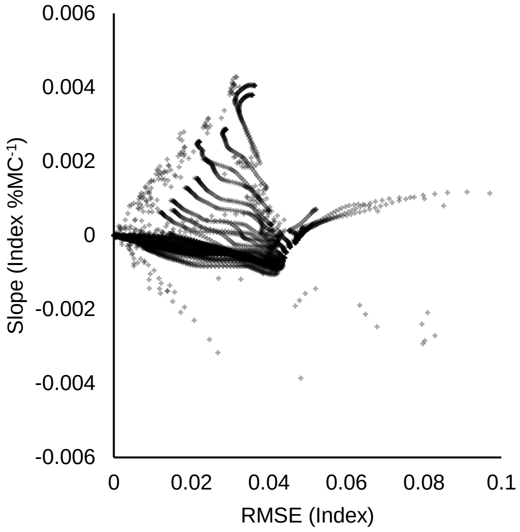


Figure 2-12: The slope of linear regression of reflectance and moisture content versus RMSE on residue stalks

The index values from 50 nm wide bands centered at 1300 nm and 1500 nm wavelengths for all samples are shown in Figure 2-13 along with the linear regression model and 95% prediction interval. Variability in index calculation among samples at a given moisture content was large for all moisture contents, thus reducing the usefulness of the index for stalk moisture content. Two possible explanations for why the index failed to perform as well for stalks as it did for soil include: (1) the non-uniform height of the stalk relative to the reflectance increased variability; and (2) the water absorbed by the stalks was not uniformly distributed, i.e. the moisture at the stalk surface did not necessarily represent the average moisture content.

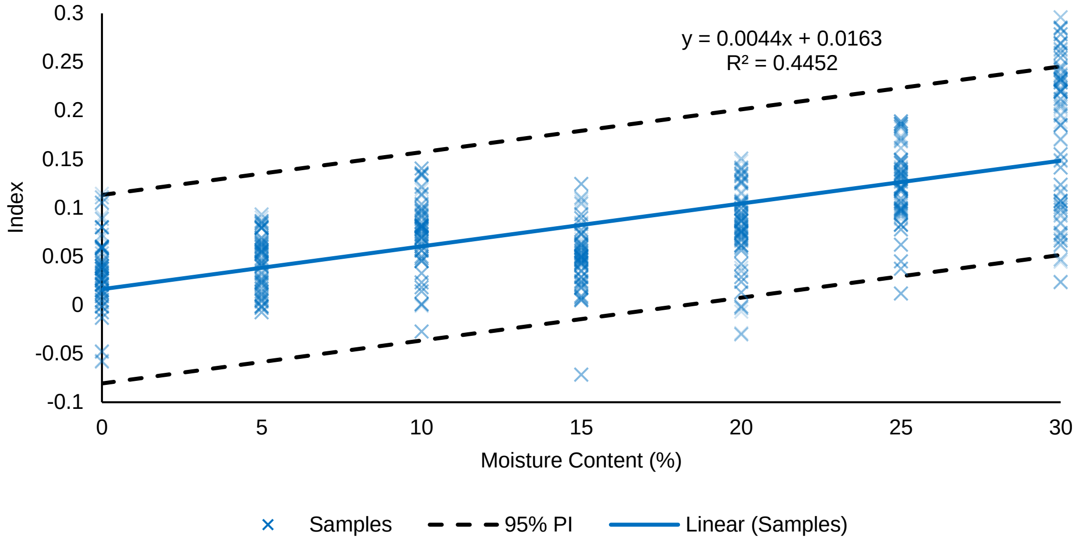


Figure 2-13: Normalized index for wheat stalk residue using 50 nm bands centered at 1300 and 1500 nm.

2.4.4 *Sensor Height*

Average index values for individual sensor heights above surface and moisture contents are shown in Table 2-2 for the bare soil and wheat stalk residue data. Results for bare soil showed a strong direct relationship between moisture content and the index value, while results for wheat stalk residue showed a weaker direct relationship. The sensor height above the sample was observed to influence the average reflectance but the effect on the index calculation was not known. Therefore, a multifactor analysis of variance (ANOVA) $(\alpha = 0.05)$ was used to determine if sensor height and moisture content significantly affected the index result (Table 2-3). The ANOVA revealed that moisture content was significant while height was not for bare soil, and that both moisture and height were significant for wheat stalk residue. This result indicates the difficulty that low-cost field sensors may encounter when observing heterogeneous ground cover. Careful control of the height of the sensor, and perhaps classification of the ground cover, may be necessary for remotely sensing soil surface moisture content.

Table 2-2: Average index measurements for bare soil and wheat stalk residue at varying moisture contents and sensor heights.

Sample	**Moisture**	**Index Value**				
	Content (%)	**H1**	**H2**	**H3**	**H4**	**H5**
Bare Soil	3.33	-0.0006	0.0016	0.0011	0.0027	0.0006
	5.0	0.0088	0.0111	0.0100	0.0122	0.0104
	10.0	0.0337	0.0335	0.0324	0.0316	0.0316
	15.0	0.0532	0.0581	0.0561	0.0557	0.0555
	20.0	0.0814	0.0936	0.0994	0.1001	0.1012
	25.0	0.1217	0.1230	0.1219	0.1253	0.1267
	30.0	0.1661	0.1716	0.1722	0.1615	0.1575
Wheat Stalk Residue	0.0	0.0101	0.0461	0.0201	0.0514	0.0497
	5.0	0.0236	0.0404	0.0487	0.0423	0.0574
	10.0	0.0105	0.0668	0.0784	0.0822	0.0939
	15.0	0.0422	0.0225	0.0551	0.0606	0.0599
	20.0	0.0457	0.0704	0.1021	0.0831	0.0863
	25.0	0.0705	0.0956	0.1330	0.1408	0.1362
	30.0	0.1908	0.1597	0.2056	0.2113	0.1835

Table 2-3: Parameter estimates and significance testing of height and moisture on the index.

Sample	**Parameter**	**Estimate**	**Std. Error**	**t Ratio**	**Prob. > |t|**
Bare Soil	Height	0.0006356	0.000851	0.75	0.4604
	Moisture	0.0061931	0.000129	48.04	< 0.0001
Wheat Stalk Residue	Height	0.0108258	0.003532	3.07	0.0044
	Moisture	0.0045207	0.000499	9.05	< 0.0001

2.5 CONCLUSIONS

Moisture controlled soil and wheat stalk residue samples were prepared and measured at varying heights using a reflectance probe connected to visible and near-infrared spectrometers. A computer program was written that used reflectance data to determine the optimal narrowband wavelengths when calculating a NDWI based upon user-defined constraints and the statistical significance of height and moisture content were determined for the "best" pair. Constraints for this study were configured to maximize the slope of the index (i.e. sensitivity to moisture) and either maximizing the R-squared or minimizing the RMSE of the index function. A linear model was chosen to represent the index when fitting parameters. Results showed that wavelengths centered near 1300 nm and 1500 nm, within the range of 400 nm to 1700 nm, produced the best index for individual samples. An advantage of this pair of wavelengths is that they can be sensed from a single type of sensor using narrowband optical filters. The 1500 nm band, when measured from an active ground-based sensor, will provide spectral information not available when using passive aerial or satellite based remote sensing methods due to absorption from atmospheric moisture. The index, when applied to all samples, performed well for the soil samples but poorly on the wheat stalk residue samples. Index calculations from soil reflectance measurements were highly linear ($R^2 > 0.95$) and exhibited small variability between samples at given moisture content, regardless of measurement height. Index calculations from wheat stalk residue reflectance measurements were highly variable, which limited the usefulness of the index for this type of material. Based on these results, it is expected that crop residues, such as wheat stalk residue, will reduce the accuracy of remotely sensed soil surface moisture measurements. Future work should include heterogeneous samples that include both soil and crop residue in varying proportions to determine the composite response. As new low-cost sensors are developed, the optimization parameters used to determine the “best” wavelengths should be refined based on actual sensor response, rather than ideal assumptions.

CHAPTER 3: OBJECTIVE 2: MOISTURE CONTENT CLASSIFICATION OF SOIL AND STALK RESIDUE SAMPLES FROM SPECTRAL DATA USING MACHINE LEARNING

3.1 SUMMARY

Remotely sensed spectral data are commonly used to quantify material properties in agricultural applications. Typically, only a few distinct spectral bands are selected and formulated into a reflectance index to avoid expensive computations while it causes more prediction inaccuracies due to ignoring other useful wave bands. Machine learning presents an alternative approach for quantifying material properties from spectral data due to the ease at which it can be used to process large datasets. This study aimed to test several commercially available machine learning algorithms using spectral data collected from moisture-controlled silt-loam soil and wheat stalk residue samples. The spectral data used in this analysis were previously used to develop a normalized difference water index (NDWI) for remotely quantifying the moisture content of background materials by selecting a pair of narrowband wavelengths. However, results showed mixed performance for index-based processing. In this study, raw spectral data were preprocessed using partial least squares (PLS) regression to optimize the number of input components. The components were fed into 20 different machine learning algorithms available in MATLAB and the best two performing methods were compared to the index-based method. Cubic support vector machine (SVM) and ensemble bagged trees methods produced the highest composite prediction accuracies of 96% and 93% for silt-loam soil samples, and 86% and 93% for wheat stalk residue samples, respectively. Prediction accuracy using the index-

based method was 86% for silt-loam soil and 30% for wheat stalk residue. A potential limitation of both machine learning methods was the discrete classification of moisture content rather than the continuous output of the index-based method. However, the substantial improvement of prediction accuracy of individual samples likely outweighs concerns about limited precision.

3.2 INTRODUCTION

Agricultural irrigation management is increasingly becoming a vital factor to supply enough food to a growing population. Irrigation – as the primary fresh water consumer – has a large influence on water shortage issues (Gleick, 2003). Remote sensing is useful for obtaining field-scale information about the drought status in a field and has received much attention in the past few decades for identifying water stress in agricultural applications (Atzberger, 2013; Bernardes et al., 2012; Carlson et al., 1981; Doraiswamy et al., 2005; Nemani & Running, 1989; Peñuelas et al., 1997; Thenkabail et al., 2014).

A common way of applying remote sensing in a field is to select a couple of narrow-band ranges of wavelengths with the potential to provide a sufficiently accurate estimation of a field parameter based on reflectance values. Combinations of these narrow-band ranges are used to compute indices that are correlated with crop and soil parameters. For instance, the normalized difference vegetation index (NDVI), which is typically a combination of red and near-infrared bands, is one of the most ubiquitous remote sensing indexes in agriculture for predicting field parameters correlated to crop vigor. The normalized difference water index (NDWI), which typically replaces one of the NDVI bands with a water-absorption band in the near-infrared range, has been shown to be a better performing index for estimating water stress (Gao, 1996; Gu et al., 2007).

A limited number of narrow-band wavelengths are used for index generation to reduce sensor cost and complexity. However, by relying on only one or two wavelengths for parameter estimation, information which can be extracted from other wavelengths are ignored. On the processing end, building the model to extract information from hyperspectral data is computationally intense, but it’s not necessarily cost prohibitive and it has little bearing on cost once the model for estimating a crop or soil parameter has been developed.

In recent years, new approaches and algorithms (e.g. machine learning algorithms) have been developed and are well suited for handling big datasets with many input variables. Learning algorithms are regularly used in daily life, frequently without being noticed. For example, web search engines commonly use learning algorithms to rank web pages. Learning algorithms try to mimic how the human brain learns by recognizing patterns and rules in a dataset (Jensen et al., 1999). A computer is given a dataset containing a large number of input variables and samples. The response variable value for each set of variables and samples is also known. In this way, a learning algorithm tries to “understand” how a set of inputs produces a specific output. By recognizing patterns in training dataset, an algorithm is trained and can be used to classify new samples. An advantage of this method is that it enables computers to be trained by learning from experiences without being explicitly programmed using an analytical model or simple empirical model. The algorithm performance generally increases by experiencing more samples, assuming the samples accurately represent the modeled process.

In general, there are two main classifications for machine learning algorithms: unsupervised learning and supervised learning. Unsupervised learning allows users to approach problems with little or no idea what the results should look like. The structure can be derived from data, where the effect of the variables is not necessarily known. Supervised learning applies when there are a large number of samples, where each sample pairs a number of input and output values.

Machine learning is a widely-used technology, and in the last decade, it has been applied to remotely sensed data in agricultural domain. Specific examples include drought assessment using MODIS and AVIRIS satellite sensors (Park et al., 2016; Trombetti et al., 2008), forecasting vegetation health using MODIS satellite sensors (Nay, Burchfield, & Gilligan, 2016), estimating LAI index using MODIS and CYCLOPS (Verger et al., 2008) and Landsat ETM+ satellite data (Walthall et al., 2004), weed detection using manual RGB imagery on the ground (Cho et al., 2002; Jafari et al., 2006) and multispectral and RGB imagery on a UAS (Koot, 2014), and plant classification using hyperspectral and RGB images (Moreno et al., 2014).

Multiple learning algorithms have been specifically used in this study for estimating soil moisture content. Among them, support vector machines (SVMs), artificial neural networks, and Bayesian methods have resulted in more powerful models based upon examples in the literature. Neural networks are considered as a traditional non-linear machine learning method for estimating soil moisture. However, it is hypothesized that simpler methods, in particular SVMs, may have the same performance or even outperform neural networks for estimating soil moisture. In addition, SVMs provided more robustness against noise in the training process (Ahmad et al., 2010; Pasolli et al., 2011; Wu et al., 2007). Bayesian methods are another prevalent machine learning method for estimating soil moisture and were also shown to perform similarly to neural networks for estimating soil moisture content (Notarnicola et al., 2008).

In a previous study by Hamidisepehr et al. (2017), visible and near-infrared spectrometers with an effective measurement range between 400 nm and 1700 nm were deployed to measure reflectance on moisture controlled soil and crop residue samples. In that study, a brute force optimization method was developed to determine the optimal pair of wavelengths used to create a moisture predicting index that maximized sensitivity of the index to changes in moisture content while minimizing error. Results showed that wavelengths centered around 1300 nm and 1500 nm produce the linear index model with the highest sensitivity to moisture content (slope), highest coefficient of determination (R^2) of the linear regression between the index values and moisture content, and lowest root mean squared error (RMSE) between predicted and actual moisture contents. The emphasis for using two wavelengths, rather than the full spectrum, was towards the development of a low-cost narrow-band sensor for field use. While results for soil samples appeared promising in terms of the ability to model the index response to soil moisture content, crop residue moisture content was difficult to accurately predict. Furthermore, prediction accuracy for soil and crop residue moisture contents were 86% and 30%, respectively, when index values from individual samples were used to classify moisture content. This study aims to expand upon that work by applying machine learning algorithms to the full spectral data to improve the prediction accuracy of moisture content in soil and crop residue samples.

The main objective of this study was to determine if machine learning could be used to develop a prediction model using relative reflectance data collected from moisture-controlled soil and crop residue samples. Specific objectives include:

1. Determine the appropriate number of spectral wavelengths to be used as input into a machine learning algorithm.
2. Test pre-configured machine learning algorithms available in MATLAB to determine which method produced the highest prediction accuracy.
3. Compare the results of the machine learning methods to a reflectance index-based method.

3.3 MATERIALS AND METHODS

3.3.1 Sample Preparation and Data Collection

The spectral dataset from Hamidisepehr et al. (2017) contained three replications of seven different moisture contents for separate silt-loam soil samples and wheat stalk residue samples. Relative reflectance was measured nine times at five different probe heights (0.64, 1.92, 3.20, 4.48, and 5.76 cm) above the sample surface – corresponding to five different sampling areas (0.06, 0.55, 1.54, 3.02, and 4.99 cm^2), given the 24.8° field-of-view of the backscatter reflectance probe used. Probe height and the order of samples within a given probe height were randomly selected to randomly distribute error resulting from drift in the measurement system. The backscatter reflectance system used to collect the spectral data was recalibrated against a white Spectralon calibration target every time the probe height was changed. A more detailed description of the instrument setup can be found in Hamidisepehr et al. (2017). The sampling scheme resulted in 945 measurements per material type across the range of established moisture contents.

3.3.2 Spectral Data Preprocessing

There were a total of 1024 distinct wavelengths measured in the spectral range of 400 to 1700 nm. While each wavelength could be considered an input variable to a machine learning algorithm, many of the wavelengths do not contain unique information (i.e. they were highly correlated to other wavelengths). Therefore, the raw spectral data were compressed using a partial least squares (PLS) regression method using MATLAB (R2015b; The Mathworks; Natick, MA) to reduce the number of input components and speed up data processing. The MATLAB function *plsregress* was used to apply the PLS regression method to the raw spectral data (141). The function returned two parameters that were useful for determining the appropriate number of components to be used in the machine learning process – estimated mean squared prediction error and the variance explained in the output parameter (moisture content). In order to find the optimal number of input components, estimated mean squared prediction error and variance explained in moisture content were plotted against the number of input components. The number of input components which provided a low prediction error and high variance explained in moisture content was selected as the optimal number of input variables to the machine learning algorithms. Each component was the combination of multiple correlated wavelengths and, given the nature of the machine learning process, no information regarding the actual wavelengths associated with a component was necessary.

3.3.3 Machine Learning Method

Each preprocessed spectral measurement included an array of relative reflectance data (input variables) that corresponded to a given moisture content (response variable). The goal of the machine learning process was to develop an empirical model that could be used to classify the moisture content of samples that were not used to train the model. In this study, soil water content classification levels were set to 3.3%, 5%, 10%, 15%, 20%, 25%, and 30% for soil samples, and water content of 0%, 5%, 10%, 15%, 20%, 25%, and 30% for stalk residue samples. After training the model, the prediction accuracy was tested by comparing the frequency of correct classifications across an independent testing dataset to identify the algorithm(s) and resulting model(s) that best estimated moisture content across both material types.

The entire dataset for each material type was randomly divided into three subsets: a training dataset, a validation dataset, and a testing dataset. In this study, 70%, 15%, and 15% of a dataset was allocated to training, validation, and testing, respectively. In the training dataset, the weights of all variables were automatically adjusted as the model was trained. The validation dataset was used to minimize overfitting and verified that any increase in accuracy over the training dataset yielded an increase in accuracy over a data set that has not been previously shown to the machine. If the accuracy over the training dataset increased, but the accuracy over the validation dataset remained the same or decreased, then the model was overfitted and training should be stopped. Finally, the testing dataset was used to assess the predictive accuracy of the model (Figure 3-1). The entire training/validation/testing process was repeated ten times to investigate the variability in classification performance for different machine learning algorithms.

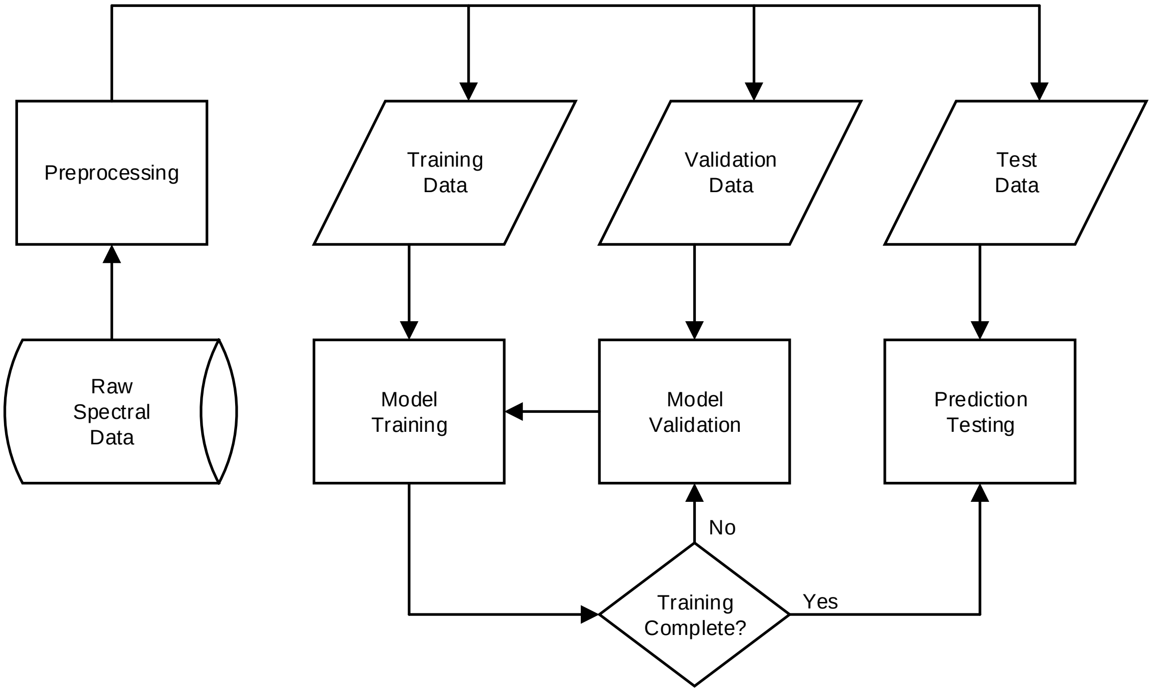


Figure 3-1: Learning process in a machine learning algorithm. Raw spectral data are preprocessed and subdivided into independent training, validation, and test datasets. The model is trained and validated until improvement in the model reaches a minimum threshold, and then the model is tested to determine the accuracy of prediction.

There were 20 different predeveloped machine learning algorithms available using the Classification Learner App in MATLAB at the time of this study. These algorithms included, among others, decision trees, support vector machines (SVM), nearest neighbor classifiers, ensemble classifiers, and a two-layer feed-forward neural network. The preprocessed dataset was organized as a matrix with reflectance components in columns and measurement samples in rows (Table 3-1). The last column contained the actual moisture content value for each sample. The components generated from PLS regression, while generated from sets of wavelengths, are no longer in units of relative reflectance (i.e., 0-100% of a reference white target). Combinations of wavelengths that were negatively correlated produced negative component values. Each machine learning algorithm was fed the same pre-processed dataset to determine which methods performed well for both soil and wheat stalk residue samples.

Table 3-1: Format of preprocessed data used for evaluating machine learning algorithms. A total of 20 components from each of 945 samples representing seven moisture contents were used to train, validate, and test 20 different machine learning algorithms.

Measurement Number	**Component**				**Moisture Content (%)**
	1	2		20	
1	-0.03101544	-0.031569856	…	-0.008929973	3.33
2	-0.03105864	-0.031872024	…	-0.010776602	3.33
:	:	:		:	:
945	0.024192134	0.020611746	…	0.070482093	30

3.3.4 Machine Learning vs. Index-Based Method Comparison

The normalized difference water index for estimating moisture content in silt-loam soil and wheat stalk residue defined in Hamidisepehr et al. (2017) produced a continuous value moisture content output. In order to make a more direct comparison between machine learning and index-based methods, the index-based moisture estimations were classified into the same levels used in the machine learning assessment by rounding to the nearest classification level. The prediction accuracy for each known moisture level and the overall prediction accuracy were determined by computing the percentage of correct predictions.

An important difference in the analysis in contrast to the machine learning method was that independent testing data were not used when classifying moisture levels from index values. Rather, the same data used to fit the index model were used. Therefore, it is expected that the prediction accuracy of the index-based method is overestimated as compared to what would have occurred had separate datasets been used for building the index model and testing.

3.4 RESULTS AND DISCUSSION

3.4.1 Spectral Data Preprocessing

Figure 3-2 illustrates estimated mean squared prediction error and percent variance explained in moisture content for a varying number of components. The percent variance explained in moisture content increased as number of components increased, which was expected, although there were diminishing returns as the number of components increased. Over 90% of the variance in moisture content was explained by four components and over 95% by 24 components. The estimated mean squared prediction error initially decreased until reaching a minimum of 6.4% at 20 components, and then began to increase. By generating a model with only 20 components, the lowest estimated mean squared prediction error and 94.7% of the variance explained in moisture content were achieved. Thus, 20 components were selected as the optimal number of components to preserve accuracy while reducing computation time and overfitting. Any increase in the number of components beyond 20 for this dataset would only result in a marginal increase in percent variance explained in moisture content while also increasing the estimated mean squared prediction error due to overfitting.

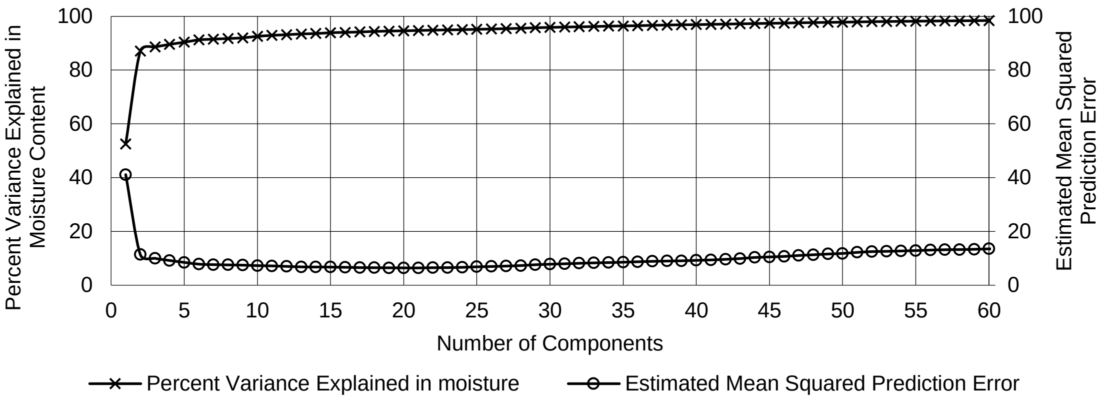


Figure 3-2: Percent variance explained in moisture and estimated mean squared prediction error for 1 to 60 components.

The dimensionality of the dataset was reduced significantly from 1024 wavelengths to 20 components. Thus, the process of generating the model and estimating moisture was substantially faster and the likelihood of overfitting was decreased by removing redundant components. A typical reduction in training time of 50% was observed for most algorithms after preprocessing raw spectral data.

3.4.2 Machine Learning Method

All 20 different pre-configured machine learning algorithms available in MATLAB at the time of this study were tested to determine the prediction accuracy for both silt-loam soil and wheat stalk residue samples. The prediction accuracy of the machine learning algorithms is shown in Figure 3-3. In general, the moisture content of silt-loam soil samples was more accurately predicted than wheat stalk residue samples. The cubic support vector machine (SVM) and ensemble bagged trees methods stood out from the other methods due to the relatively high prediction accuracy for both material types, which was desirable due to the likelihood of observing both materials in combination under field conditions. The ensemble bagged tree method was also the global optimum when considering both material types, having the highest combined prediction accuracy for both materials. An important note was that, because the training, validation, and testing datasets were randomly distributed, the results presented varied slightly with individual prediction accuracies varying by a few percent each time the machine learning process was repeated. The standard deviation in prediction accuracy is shown in Figure 3-3 as error bars and represents the results of ten replications randomly distributing the full data between training, validation, and testing datasets. Increasing the number of samples would likely have reduced the variation in prediction accuracy between processing attempts.

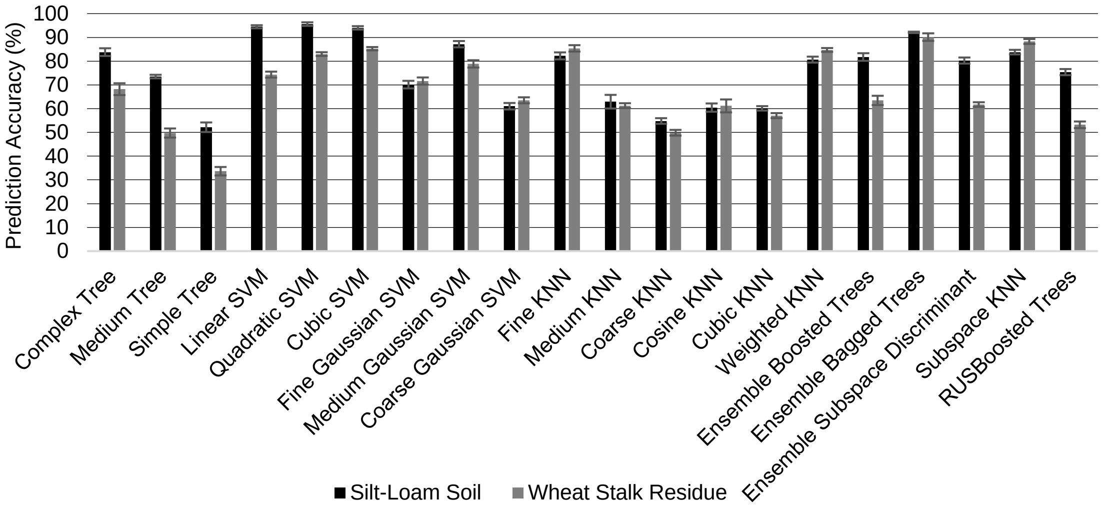


Figure 3-3: Prediction accuracy for 20 machine learning algorithms applied to relative reflectance data from moisture-controlled silt-loam soil and wheat stalk residue samples.

Figure 3-4 and Figure 3-5 illustrate the prediction accuracy by moisture content for both materials when using the cubic SVM and ensemble bagged tree methods. The rows represent predicted moisture content and columns represent actual moisture content. In this matrix form, if the moisture is predicted correctly, the measurement would be placed on the diagonal of the matrix. Elements not on the diagonal are the result of misclassification for a given sample. The numbers displayed in each cell indicate the frequency of occurrence as a percentage of predictions for a given moisture content and the relative opacity of the cell corresponds to the same percentage. The overall accuracy of an algorithm is represented by the ratio of the number of correct predictions and the total number of predictions. The overall accuracy for silt-loam soil samples were 96% for cubic SVM and 93% for ensemble bagged trees. The overall accuracies for wheat residue stalks samples were 86% for cubic SVM and 93% for ensemble bagged trees. The most frequent prediction inaccuracies in soil samples happened at lowest and highest moisture content levels in both algorithms. For residue stalks samples, higher percentages of prediction inaccuracies were scattered across all moisture content levels. As with the previously described small variation in overall prediction accuracy each time the dataset was processed, minor variations in individual moisture content prediction accuracies also occurred.

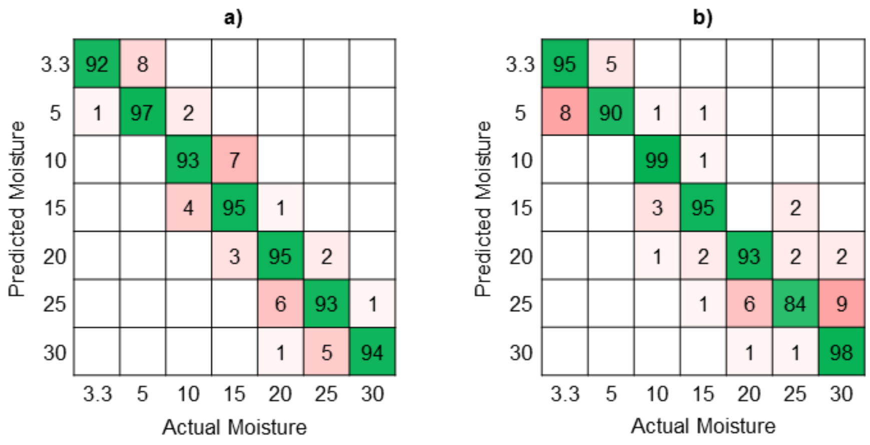


Figure 3-4: Machine learning classification results for silt-loam soil samples using a) Cubic SVM; b) Ensemble Bagged Trees.

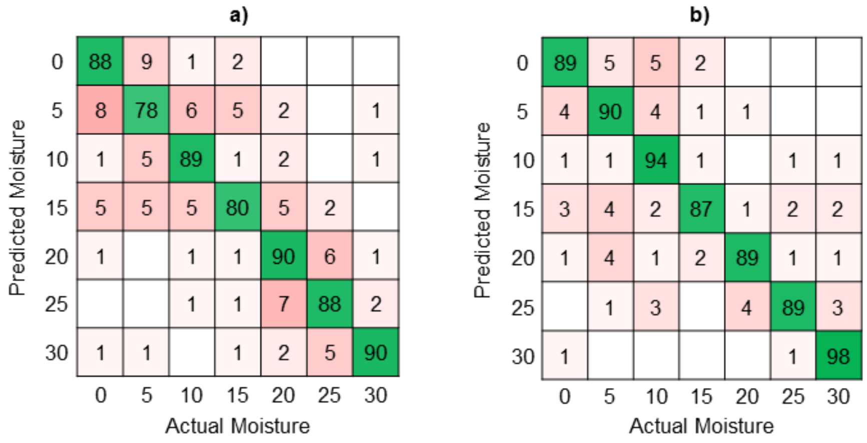
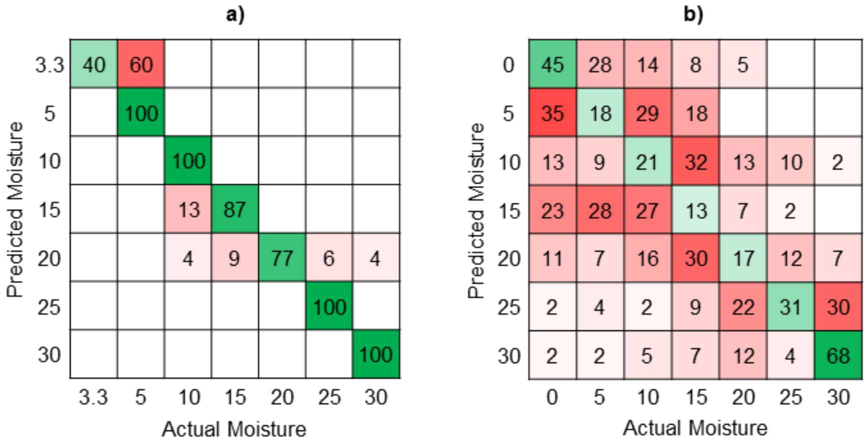


Figure 3-5: Machine learning classification results for residue stalks samples using a) Cubic SVM; b) Ensemble Bagged Trees.

3.4.3 Machine Learning vs. Index-Based Method Comparison

Figure 3-6 illustrates the prediction accuracy by moisture content of the index-based method. The average prediction accuracies for both materials were substantially higher when using cubic SVM and ensemble bagged trees machine learning methods as compared to what was obtained from the index-based method in the previous work, especially for wheat stalk residue samples. Most if the improvement in soil moisture classification was attributed to just one classification level, 3.3% MC, and the remainder came from two additional classification levels, 15% MC and 20% MC. In fact, the remaining moisture contents were better predicted using the index-based method than with machine learning. However, the average prediction accuracy improved by 10% when using the Cubic SVM method. The index-based method performed poorly at moisture content levels where there were large overlaps in index values. Averaging multiple samples would improve the performance of the index-based classification method.



a)

	3.3	5	10	15	20	25	30
3.3	40	60					
5		100					
10			100				
15			13	87			
20			4	9	77	6	4
25						100	
30							100

b)

	0	5	10	15	20	25	30
0	45	28	14	8	5		
5	35	18	29	18			
10	13	9	21	32	13	10	2
15	23	28	27	13	7	2	
20	11	7	16	30	17	12	7
25	2	4	2	9	22	31	30
30	2	2	5	7	12	4	68

Figure 3-6: Index-based method classification results for a) silt-loam soil samples and b) wheat stalk residue samples.

There are several considerations when comparing the index-based and machine learning methods for moisture content prediction. Reflectance indices produce a continuous output whereas machine learning algorithms typically classify the sample into a predetermined bin. The relative improvement in prediction accuracy of the machine learning algorithms shown in Table 3-2 are partly a result of only having seven classification levels. If more moisture content treatments had been used, it is likely that the overall performance of the machine learning algorithms would have degraded due to the smaller thresholds for classifying a measurement. On the other hand, having a continuous output may not be inherently more useful for irrigation management due to a limited number of controllable output states. Machine learning as it was applied in this study represents a less precise, but more accurate method for moisture content prediction. Another key advantage to machine learning over index-based methods is the ability to accurately predict the moisture content of a single sample, which is how the data were compared in this study. The previous work by Hamidisepehr et al. (2017) showed that moisture controlled sample replications produced a wide range of reflectance index values, and when taken on average produced a highly linear relationship between silt-loam soil moisture content and the index value (R^2 = 0.96) and a moderately linear relationship between wheat stalk residue moisture content and the index value (R^2 = 0.45). At least in case of the silt-loam soil samples, collecting multiple measurements and averaging them rather than relying on just one would have a substantial impact on moisture content prediction accuracy.

Table 3-2: A comparison of overall prediction accuracy between index-based method and two most accurate machine learning methods for silt-loam soil and wheat stalk residue samples

Material	**Index-Based Method**	**Machine Learning Methods**	
		Cubic SVM	**Ensemble Bagged Trees**
Silt-Loam Soil	86%	96%	93%
Wheat Stalk Residue	30%	86%	93%

3.5 CONCLUSION

Relative reflectance spectral data from moisture-controlled silt-loam soil and wheat stalk residue samples was used to test the ability of several machine learning algorithms to predicted moisture content from the spectral data. This method was in contrast to an index-based method used in a previous study of the same spectral data. Previous work has tended to focus on a pair of wavelengths rather than the full spectrum. It was hypothesized that the machine learning approach would yield better prediction accuracy because of utilizing a larger number of components from the spectral data. The appropriate number of components for this dataset was determined to be 20 using PLS regression. The components were fed into 20 different machine learning algorithms, from which cubic SVM and ensemble bagged trees produced the highest combined prediction accuracy for silt-loam soil samples (over 93%) and wheat stalk residue samples (over 86%). This represented a substantial improvement over the index-based method, where only two wavelengths were used to develop a moisture prediction model. The high variability in performance between machine learning methods demonstrates the importance of trying multiple methods for a given dataset rather than simply selecting one based upon previous work. Several machine learning methods resulted in unacceptable low moisture classification performance, or a large deviation in classification performance between multiple material types that are frequently observed together in nature.

The results of this study were from laboratory prepared samples of individual material types measured under controlled conditions. Field application of this work will require additional considerations including, among other factors, more complex distributions of materials and variability in ambient light. Future work should include testing of this process under field conditions to demonstrate the applicability as a high-throughput method for remotely sensing moisture content of soils and crop residues.

CHAPTER 4:OBJECTIVE 3: INSTRUMENTING LOW-COST SPECTRAL REMOTE SENSING ABOARD A SMALL UNMANNED AIRCRAFT SYSTEM AND A METHOD FOR AMBIENT LIGHT COMPENSATION

4.1 SUMMARY

Small unmanned aircraft systems (UAS) are a relatively new tool for collecting remote sensing data at dense spatial and temporal resolutions. This study aimed to develop a spectral measurement platform capable of being deployed on a UAS for future use in quantifying and delineating moisture zones within agricultural landscapes. A series of portable spectrometers covering ultraviolet (UV), visible (VIS), and near-infrared (NIR) wavelengths were instrumented using a Raspberry Pi embedded computer that was programmed to interface with the UAS autopilot for autonomous reflectance data acquisition. A second set of identical spectrometers were fitted with calibrated irradiance lenses to measure ambient light energy during reflectance data acquisition. Data were collected during the 2017 Great American Eclipse in Russellville, Kentucky while observing a reflectance target to determine the ability to compensate for ambient light conditions. A compensation routine was developed that scaled raw reflectance data by sensor integration time and ambient light energy. Results indicated the potential for mitigating the effect of ambient light when passively measuring reflectance on a portable spectral measurement system.

4.2 INTRODUCTION

Efficient irrigation management is one of the most important issues producers face in arid and semi-arid areas.(Kang et al., 2000; Perry, 2007) Novel technologies, such as variable-rate irrigation, help to control water usage and result in more efficient irrigation.(O'Shaughnessy et al., 2015; Yari et al., 2017) To spatially implement variable-rate irrigation, a prescription map containing information about the actual water status of the field is needed.(Buck et al., 2016) Remote sensing is one possible method to obtain water status at field-scales necessary for generating near real-time irrigation prescription maps. Traditional deployments of remote sensing include satellite and conventional aircraft, but are limited in terms of cost, temporal resolution, and spatial resolution. Unmanned aircraft systems (UAS), or drones, are relatively new tools for collecting remote sensing data in agricultural applications.(Chrétien et al., 2015; Pádua et al., 2017) In one study, traditional methods of remote sensing, including satellites and manned aircraft systems, were compared to a UAS method. It was concluded that UAS were more cost-effective in fields smaller than five hectares and UAS were shown to have the potential to provide higher spatial precision data.(Matese et al., 2015)

Reflectance-based remote sensing is a potential method for quantifying soil moisture and delineating moisture management zones. Several studies have focused on using visible, multispectral, or hyperspectral cameras mounted on drones to evaluate crop/soil status at high spatial resolutions. Spectral reflectance data collected using UAS are extensively used in research to estimate different soil and crop parameters. Example applications of UAS-based remote sensing in the agricultural domain include estimating soil fertility(Bajwa & Tian, 2005), generating vegetation indices(Berni et al., 2009; Candiago et al., 2015), assessing tree crowns for breeding applications(Díaz-Varela et al., 2015), yield estimation(Geipel et al., 2014), plant classification(Hung et al., 2014), weed detection(Koot, 2014), and controlling herbicide applications.(Xiang & Tian, 2011)

Advances in spectrometer development have led to more portable systems, or micro-spectrometers, that are particularly suitable for UAS deployment due to their small size and mass. One study showed that measurements from UAS-deployed micro-spectrometers were highly correlated with parameters measured at ground level and concluded that the UAS platform could provide a faster method for spectral data collection (Burkart et al., 2014). A subsequent study, where a visible micro-spectrometer was used to measure reflectance, showed that the remote sensing estimations were highly correlated to ground spectral measurements collected with a portable field spectrometer (Von Bueren et al., 2015). In another study, a spectrometer coupled with a camera was mounted on a UAS to measure reflectance values from different targets. The same model spectrometer had previously been used to collect data from a satellite in orbit, and when data collected from these two platforms were compared, it was confirmed that the UAS provided an efficient platform for collecting spectral data (Tsouvaltsidis et al., 2015).

Calibration of hyperspectral measurement systems is challenging due to the large number of factors that can influence spectral response. For lab-based spectrometry, measurements are taken under controlled light conditions, which cannot be applied with UAS-deployed spectrometers under field condition with frequent changes in ambient light. The empirical line method is one of the common approaches for calibrating hyperspectral images against variable illumination. In this approach, tarps or panels with known relative reflectance are placed in a field during the data collection. By finding the relationship between known reflectance values and digital count output of the sensor, an equation is obtained and then applied to all measurements to complete the calibration process (Burkart et al., 2014; Von Bueren et al., 2015). The data collection period is limited since changing sun angle during data acquisition affect the reflectance (Bajwa & Tian, 2005). Transient cloud cover can also substantially affect the amount of ambient light present over short durations. Another shortfall is the practical limitation of having tarps or other reference targets in all images, especially when high resolution data is desired or a large area is covered (Zeng et al., 2017). Devising a method that can keep track of ambient light changes while measuring the reflectance from a spectral target is desired. By automating this measurement process through concurrent ambient light detection, a compensated reflectance can be obtained for every single wavelength in the spectrum at a low cost and under various ambient light conditions (Cocks et al., 1998; Eismann, 2012).

Previous work by Hamidisepehr et al. (2017) showed that relative reflectance in the visible and near-infrared range could be used to optimally develop a normalized difference water index (NDWI) that predicted soil moisture content from direct soil observations. However, the experiment relied on a controlled light source, which is not practical for UAS-based spectral measurements. In this study, the overall goal was to develop a spectral sensing platform suitable for UAS deployment and to measure the reflectance from a reference target to assist with the development of a calibration procedure that is functional over a wide range of ambient light conditions. Specific objectives included:

1. Instrument a series of portable spectrometers and integrate into a UAS for autonomous data collection.
2. Develop a method to compensate for ambient light conditions and sensor integration time when collecting raw spectral reflectance measurements.

4.3 MATERIALS AND METHODS

4.3.1 Sensor Instrumentation

Two spectral data acquisition systems were used in this study: an updward-facing ambient light system to measure solar irradience at ground level, and a downward-facing reflectance system to measure reflectance from a target located at ground level. Both systems consist of three spectrometers (STS, Ocean Optics, Dunedin, FL) in the ultraviolet (UV), visible (VIS), and near-infrared (NIR) ranges (Table 4-1). Ambient light and reflectance raw spectral measurements were reported as 14-bit digital count values in integer format. The ambient light spectrometers were fitted with optical diffusers and factory calibrated in compliance with NIST practices. Calibration data were used to convert raw spectral measurements at each wavelength from an integer count value to units of energy. The reflectance spectrometers were fitted with collimating lenses to set the field-of-view (FOV) and align the light entering the spectrometers.

Table 4-1 Model number and lens type for individual spectrometers in ambient light and reflectance systems.

System	Spectrometer Type	Model Number	Optics
Ambient light	UV	STS-UV-L-25-400-SMA	CC-3-DA
	VIS	STS-VIS-L-50-400-SMA	CC-3-DA
	NIR	STS-NIR-L-25-400-SMA	CC-3-DA
Reflectance	UV	STS-UV-L-100-400-SMA	74-DA
	VIS	STS-VIS-L-100-400-SMA	74-DA
	NIR	STS-NIR-L-100-400-SMA	74-DA

4.3.2 Data Acquisition System

The target application for this measurement system was to automate the collection of spectral reflectance data from a UAS platform while compensating for varying ambient light conditions. This meant data from multiple spectrometers would need to be remotely recorded at defined locations or intervals. To accomplish this requirement, each system was coupled with a Raspberry Pi 3 (RPi) (B V1.2, Raspberry Pi Foundation, Cambridge, United Kingdom) as an embedded data acquisition system to control the measurement process. The spectrometer manufacturer provided a software development kit (SDK) for the RPi that was preinstalled on a Raspbian distribution of Linux (Version 7; Raspberry Pi Foundation, Cambridge, United Kingdom). The SDK configured the RPi as a web server, facilitating wireless control of individual spectrometers via a WiFi connection to a PC. Since WiFi control of the UAS-deployed system would not be practical in production agricultural applications due to limited range, a pulse-width-modulation (PWM) to digital converter was used to allow the UAS autopilot to trigger measurements using a digital input on the RPi. The UAS was fitted with a commercial autopilot system (A3, DJI, Nanshan District, Shenzhen, China) and configured to output a PWM signal corresponding to the "shutter" command normally used to trigger the capture of imagery from an on-board camera. As a result, triggering the spectrometer was identical to taking a picture from the perspective of mission planning software. Figure 4-1 shows the major components used in the reflectance and ambient light systems.

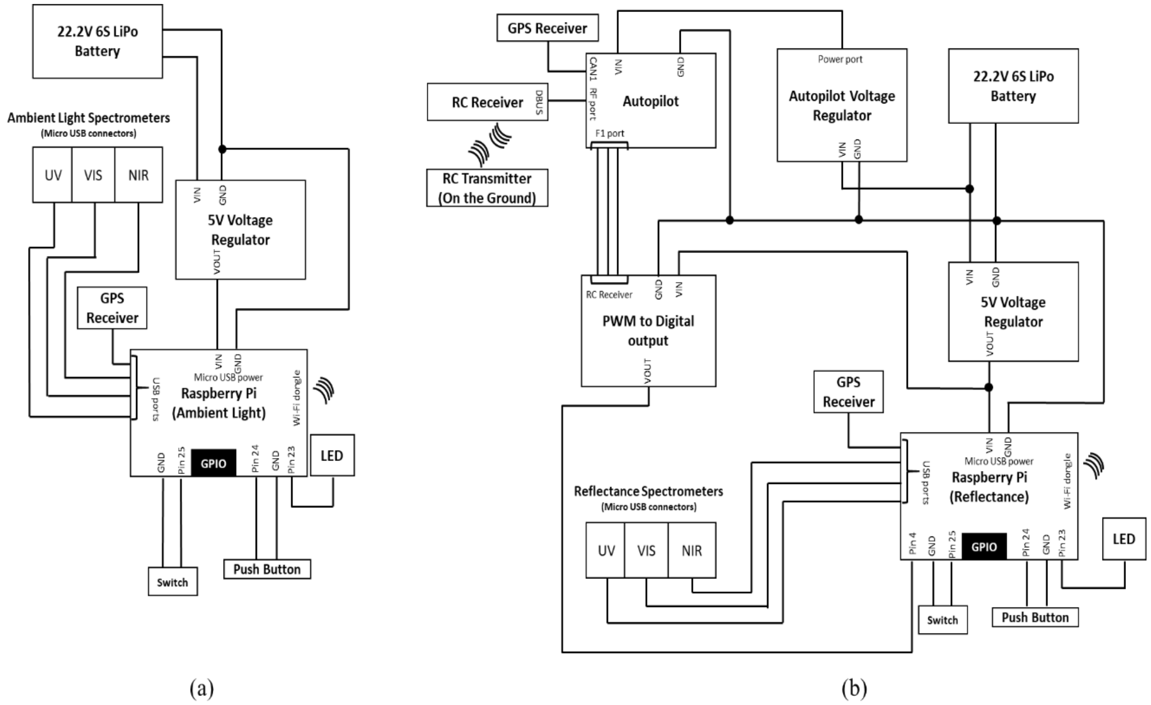


Figure 4-1 Hardware block diagram schematic of the data acquisition system including major components for (a) ambient light and (b) reflectance measurement systems.

Components inside the autopilot included: a voltage regulator (A300P-PMU PRO, DJI) to regulate input voltage to the autopilot, a GPS receiver (A300P-GPS COMPASS PRO, DJI) to update the time and the location of the UAS, and a radio control (RC) transmitter (GL858A, DJI) and receiver (R810A, DJI) to control the UAS manually and also to monitor the flight via a live video feed.

Both ambient light and reflectance systems were equipped with similar supporting components including: a voltage regulator (2858, Pololu, Las Vegas, NV) to set the RPi input voltage at 5 V, a GPS receiver (BU-353S4, USGlobalSat, Chino, CA) with a USB connection to synchrnoize the local time on the RPi to Universal Coordinated Time (UTC), and three LEDs to indicate the system was on, if sensors were taking measurement, and to show if the system was expecting an external signal to trigger a measurement or if sampling was to proceed automatically on a pre-programmed interval. A PWM-to-digital converter (2801, Pololu, Las Vegas, NV) was used in the reflectance system to allow the autopilot to trigger functions on the RPi via a logic level on a digital input pin. For UAS operations, the ambient light system was configured to collect data on a regular interval during UAS deployment, rather than triggering remotely, to remove the need for a wireless trigger between the UAS and ambient light system. For ground operations, both systems were configured to collect data on a regular interval. Ambient light and reflectance spectral data were interpolated to a synchronous time interval prior to compensating reflectance values for ambient light.

Two Python scripts were written and configured to run immediately upon startup. The first script continously polled digital input pins associated with pushbutton switches and the PWM-to-digital converter. Upon receiving the appropriate signal, or in the event the system was configured to take measurements at a predefined interval, the script would generate a series of HTML function calls to the SDK web server causing the spectrometers to take measurements and the RPi to record the data on the onboard SD card. Each measurement produced a new file with a unique filename. Filenames consisted of the spectrometer serial number and a local date/time stamp (144). The second script initiated the GPSD Linux library that facilitated communication with the GPS receiver and regularly updated the local date and time on the RPi to UTC. This was necessary because the RPi does not have a real-time clock for keeping track of local time while powered down and is normally configured to set the local time from a network time protocol (NTP) time server via the internet (142). Figure 4-2 and Figure 4-3 demonstrate the data collection process in each set of spectrometers when operating at a regular interval and through a digital input trigger, respectively. An on-off pushbutton switch mounted to each system was used to determine which process was implemented so that identical programs resided on the ambient light and reflectance systems.

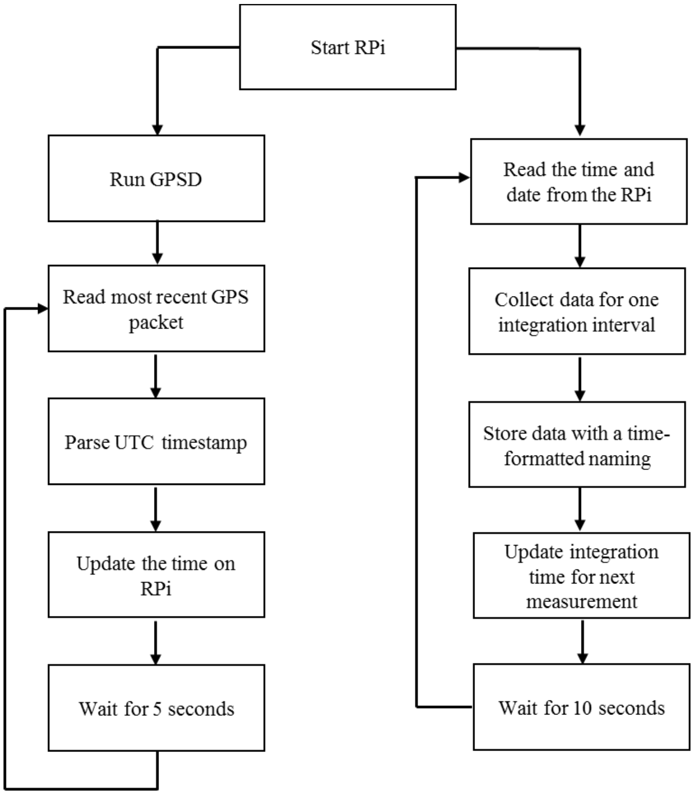

Start RPi
Run GPSD
Read most recent GPS packet
Parse UTC timestamp
Update the time on RPi
Wait for 5 seconds
Read the time and date from the RPi
Collect data for one integration interval
Store data with a time-formatted naming
Update integration time for next measurement
Wait for 10 seconds

Figure 4-2 Embedded control and data acquisition software block diagram for the ambient light system.

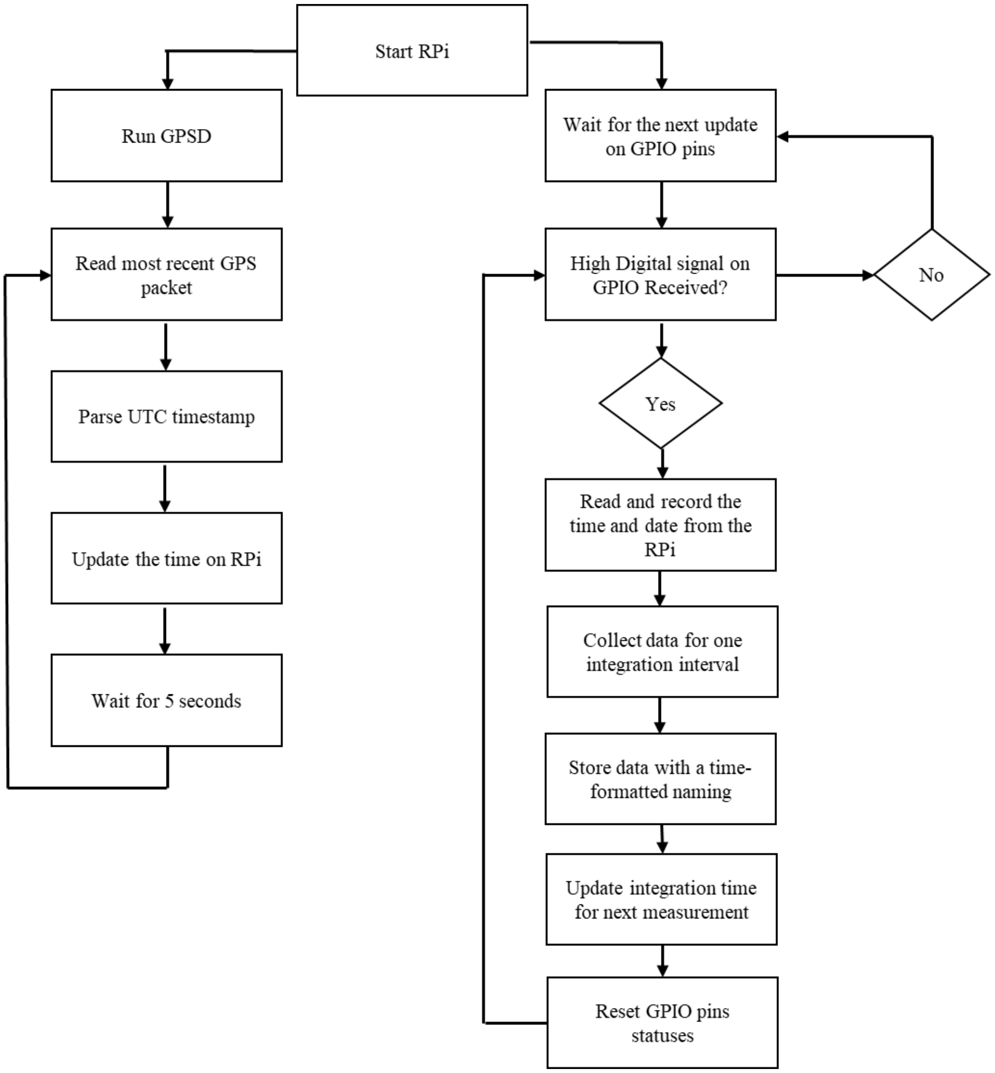

Start RPi
Run GPSD
Read most recent GPS packet
Parse UTC timestamp
Update the time on RPi
Wait for 5 seconds
Wait for the next update on GPIO pins
High Digital signal on GPIO Received?
No
Yes
Read and record the time and date from the RPi
Collect data for one integration interval
Store data with a time-formatted naming
Update integration time for next measurement
Reset GPIO pins statuses

Figure 4-3 Embedded control and data acquisition software block diagram for the reflectance system.

Each path leaving the *StartRPi* block represents an individual Python script. In the left path, the UTC timestamp is extracted from the most recent GPS data packet and used to update the time on the RPi at a 5-second interval. The right path differ slightly depending on whether the system is intended to be triggered on a regular interval based on a timer (Figure 4-2) or by a change of state on a digital input corresponding to a signal from the

PWM-to-digital converter when used with the autopilot (Figure 4-3). Regardless of the trigger method, the RPi stored the current local time and recorded data from the spectrometers into tab-delimited text files. Figure 4-4 shows a picture of the reflectance system mounted on a multi-rotor UAS. More information on dimensions of the data acquisition systems can be found in 173.

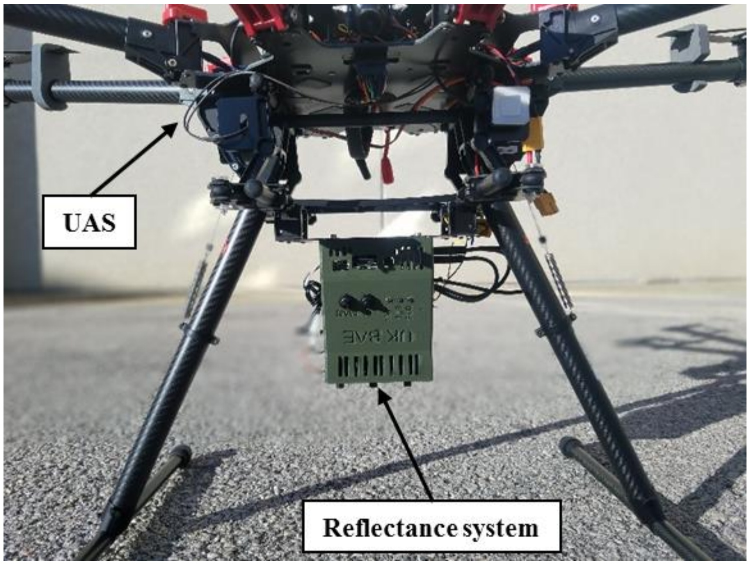


Figure 4-4 Reflectance system mounted on a DJI S1000+. Spectral data are recorded at pre-defined GNSS waypoints by triggering the shutter command in the UAS autopilot.

4.3.3 Reflectance Target and Test Stand

A reference target was fabricated and calibrated against a white standard. The reflectance target consisted of a 30.5 cm × 30.5 cm × 1.9 cm piece of plywood with a 0.08 cm thick Teflon sheet glued to the surface. A threaded insert was mounted at the center of the reference target to allow the reference target to be mounted to a standard surveying tripod during field use. The reference target was calibrated using a backscatter reflectance system that consisted of: visible and near-infrared spectrometers (HR400-7-VIS-NIR, NIRQuest512, Ocean Optics, Dunedin, Fla.), a tungsten-halogen light source (HL-2000-HP-FHSA, Ocean Optics), a fiber optic backscatter reflectance probe (QR200-12-MIXED, Ocean Optics), and a Spectralon calibration target (WS-1-SL, Ocean Optics). It was assumed that the reflectance target had a constant spectral response at different temperatures and atmospheric conditions.

A test stand was used to consistently position the target underneath the reflectance system. The height of stand is adjustable for different areas of coverage and has a square base for positioning the reflectance target. Spectrometers were located on the top of the stand above the target and off-center to prevent the middle spectrometer from sampling the threaded insert (Figure 4-5). The height of reflectance spectrometers was adjustable, and the area sampled and overlap between individual spectrometers was a function of the height of the reflectance spectrometers lenses above the reflectance target and the field of view (FOV) (171).

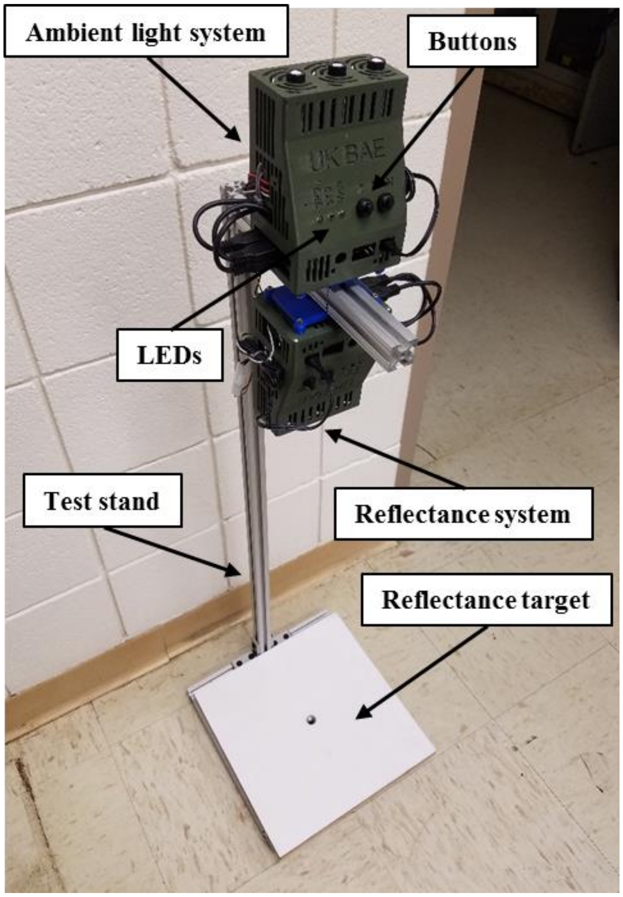


Figure 4-5 Stand for mounting spectrometers and placing reference target underneath sensors.

Table 4-2 and Figure 4-6 show the elliptical area covered at different heights above the reflectance target. In this setup, ambient light spectrometers were co-located on the top of reflectance spectrometers to mitigate variability in ambient light at the sensor location and the reflectance target.

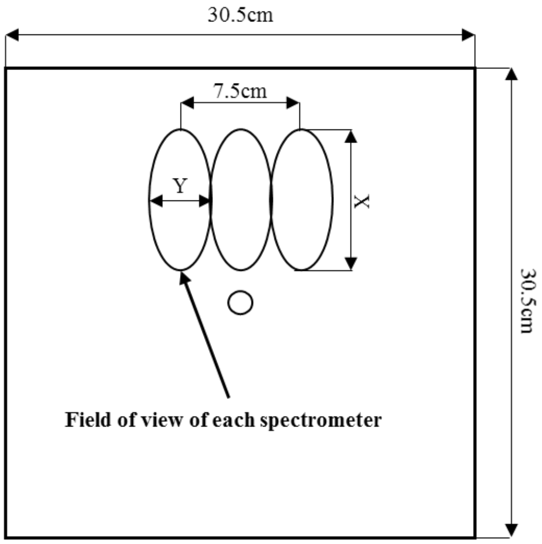


Figure 4-6 Schematic of area covered by a STS spectrometer on the reflectance target.

Table 4-2 Major (X) and minor (Y) axis dimensions of the FOV covered by the STS spectrometers using 74-DA lenses at different sensor heights above the reflectance target.

FOV	**Height Above Target (m)**			
Dimensions	**1**	**0.75**	**0.5**	**0.25**
Y (m)	0.04	0.03	0.02	0.01
X (m)	0.09	0.06	0.04	0.02

4.3.4 Integration Time

In addition to ambient light conditions, integration time of the reflectance spectrometers was considered as an important parameter in reflectance measurements. Integration time is effectively a form of gain on the input signal – if set too high, the spectrometer will produce a saturated output, and if set too low, the output will lack sufficient detail for target classification. Doubling the integration time was assumed to have a similar effect of doubling the ambient light intensity. In this study, integration time for each reflectance spectrometer was manually adjusted to maximize the output signal without saturation in any wavelength during maximum ambient light intensity and kept constant during data collection. The integration times for the ambient light spectrometers were set according to the manufacturer's recommendations to use the factory solar irradiance calibrations. Considering different integration times on each spectrometer, there were different time intervals for consecutive measurements for the ambient light and reflectance systems. Each spectral measurement reported was an average of 5 sequential measurements and there are three replications for each measurement, totaling 15 measurements on each spectrometer per measurement interval. Table 4-3 shows the integration time for each spectrometer and measurement interval for each system.

Table 4-3 Integration time and measurement interval for ambient light and reflectance spectrometers

Spectrometers	**Integration Time (ms)**			**Measurement Interval (s)**
	UV	**VIS**	**NIR**	
Ambient light	1000	180	1000	29
Reflectance	70	35	55	16

Ideally, each reflectance measurement should be paired with an ambient light measurement taken at the same time. However, since integration times varied based on the spectrometer type and function, measurements were not temporally synchronized. To compensate for asynchrnonous sampling, ambient light measurements were interpolated using two adjacent measurements and weighted according to the difference between the time of ambient light measurement and the time of reflectance measurement (ordinary kriging method).

4.3.5 *Compensation Equation*

Ambient light and sensor integration time were assumed to have a linear relationship with the raw measurement intensity from the reflectance system. Thus, the raw measurement intensity for each reflectance spectrometer was divided by the ambient light energy measured by the corresponding ambient light spectrometer and the integration time of the reflectance spectrometer. Compensating for ambient light and sensor integration time was accomplished using

$$R_\lambda = \frac{M_\lambda}{S_\lambda \times t_i} \tag{1}$$

where R were the compensated reflectance measurements from a sample (counts × μJ^{-1} × ms^{-1}), M were the raw measurement intensities from the reflectance spectrometer (counts), S were the ambient light energies (μJ), t_i was the integration time of the reflectance spectrometer (ms), and λ were the center points of each wavelength (nm).

4.3.6 *Spectral Data Collection*

Spectral data were collected during the 2017 Great American Eclipse at the Russellville-Logan County Airport in Russellville, KY on August 21, 2017. The spectrometer systems were mounted to the test frame and configured to record automatically between approximately 1:15 pm and 4:00 pm EDT (Figure 4-7). Data collection encompassed the entire eclipse, including totality. The stand was oriented such that no shadows from components above the target would be cast on the target, which would have resulted in an offset between the ambient light system and the reflectance system.

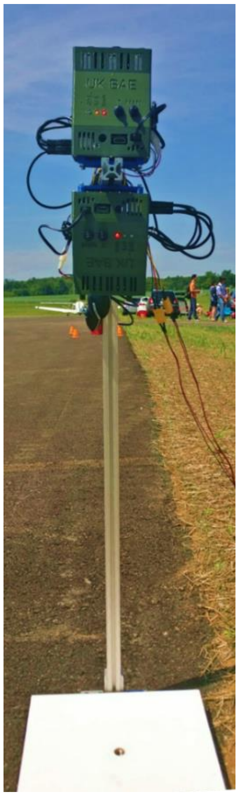


Figure 4-7 Data acquisition during the Great American Eclipse 2017

Data processing and analysis were performed in MATLAB. A script was written to process data in four steps: 1) importing reflectance and ambient light data, 2) synchronizing reflectance and ambient light data, 3) applying the solar irradiance calibration to the ambient light data, and 4) compensating reflectance data with calibrated irradiance data and sensor integration time (158).

4.3.7 Weather Station Data Collection and Spectral Comparison

During this experiment, a weather station (HOBO U30, Onset Computer Corporation, Bourne, MA) was deployed to collect atmospheric data including ambient light, temperature, and humidity. Ambient light data from the weather station pyranometer (S-LIB-M003, Onset Computer Corporation) was used to track the progress of the eclipse and serve as a reference for the spectrometers in the event any anomalous data were collected. The pyranometer measured solar power per unit area however, it did not provide separate intensities for each wavelength. Instead, it computed a weighted average from the spectrum ranging 300 to 1100 nm and output a single value from each measurement. Weights for different wavelengths were provided in the user manual (HOBO_DataLogger) and used to compare results between the pyranometer and the ambient light spectrometers. Data from all three ambient light spectrometers were spliced together to form a single spectrum encompassing the same range as the pyranometer prior to computing the weighted average at each sample time.

Before applying weights to ambient light spectrums, units were converted from digital count values at individual wavelengths to a single estimate of power in watts using a few steps. First, calibration coefficients provided by the spectrometers manufacturer were applied to ambient light measurements to change the units from counts to joules. Second, by dividing the energy in joules by integration time for each ambient light spectrometer, instantaneous power was obtained in watts. Measurements taken by weather station sensor were in watts per square meter, which represented the amount of light energy absorbed by the sensor in a second over a unit surface area; however, the area inside the spectrometers that energy was received was not known. Therefore, a linear regression model was used to determine the relationship between power per unit surface area and instantaneous power – effectively modeling the surface area of the spectrometers.

4.4 RESULTS AND DISCUSSION

4.4.1 Reference Target

To benchmark the reflectance target, spectral responses were compared to a reference target which established 0% and 100% relative reflectance for the backscatter reflectance system. Figure 4-8 shows that the relative reflectance of the reflectance target fell between 60% and 82% of the reference target. While not critical for this experiment, the reference target calibration would allow the data presented to subsequently be benchmarked to a reference standard if needed. It also indicated that the reflectance target did not uniformly reflect light across the wavelengths sampled during the experiment and tended to absorb NIR light at a higher rate than VIS light.

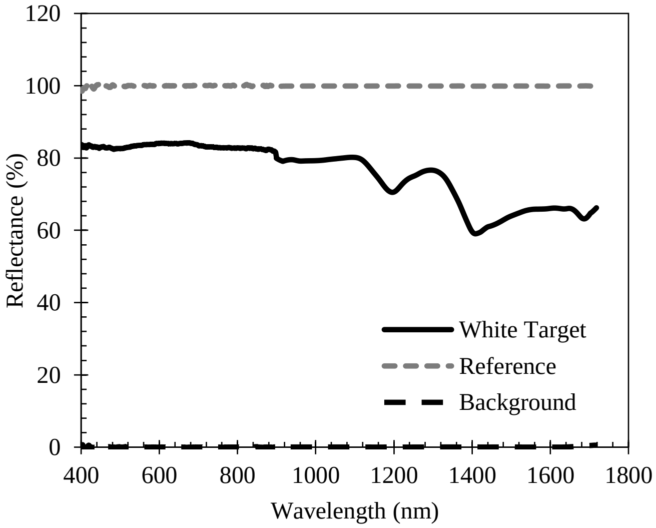


Figure 4-8 Reflectance target spectrum versus reference and background spectrums

4.4.2 Ambient Light and Reflectance Measurements

In Figure 4-9, the raw spectral data from the ambient light spectrometers can be seen in uncalibrated units of 14-bit integer counts. Each line represents an individual measurement and there were 413 measurements in total for each ambient light spectrometer. The variability in intensity for most of the wavelengths was high as a result of ambient light changes during the eclipse. Low-intensity responses represent the measurements taken near totality. On the other hand, high intensity responses indicate the measurements were taken near the beginning or end of the eclipse. While not immediately apparent, a single "dead" pixel was identified at 876.2 nm in the NIR spectrometer (Figure 4-9c), which became obvious when computing the compensated reflectance (Figure 4-12c).

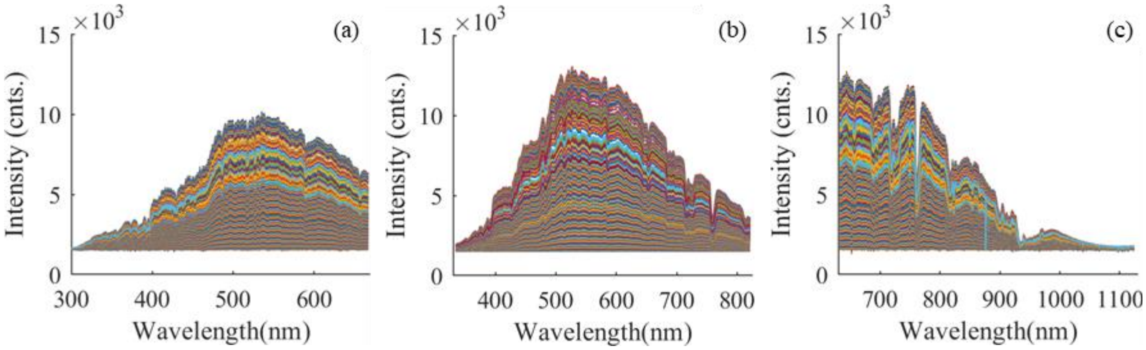


Figure 4-9 Raw ambient light measurements during the eclipse for (a) UV; (b) VIS; (c) NIR spectrometers

Interpolating ambient light measurements to the reflectance measurement sampling interval and applying calibration coefficients produced spectral responses in units of microjoules (Figure 4-10). Applying the calibration coefficients removed much of the spectral variability inherent in the ambient light systems that occurs near the upper and lower wavelengths. The resulting spectral responses closely match each other in areas where the sampled wavelengths intersect with some minor variations. The relatively small signal strength at the boundaries of each spectrometer are most likely responsible for discontinuities between spectral energy values across spectrometers.

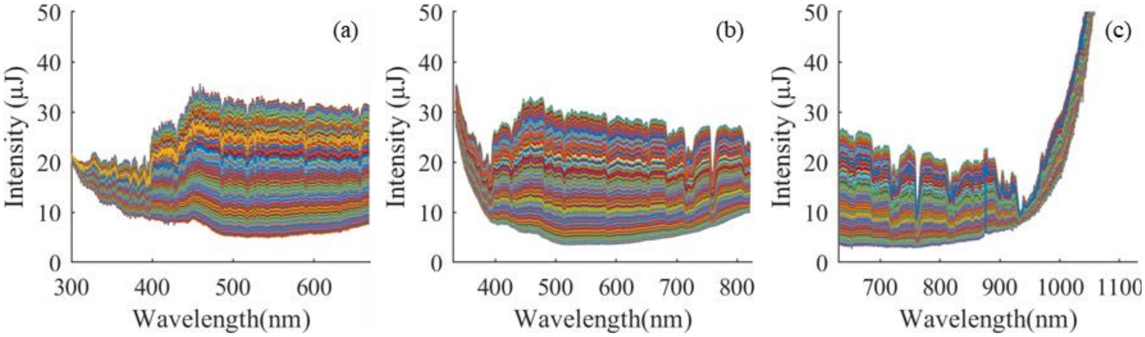


Figure 4-10 Calibrated ambient light measurements during the eclipse for (a) UV; (b) VIS; (c) NIR spectrometer

The raw reflectance data are shown in Figure 4-11 exhbiting similar responses as the raw ambient light measurements with similar variability in intensity due to varying ambient light intensity during the eclipse. Again, the lowest raw reflectance intensity occurred during totality, and the high-intensities represented measurements at the beginning or the end of the eclipse. The VIS reflectance spectrometer had a single “hot” pixel at 787.4 nm (Figure 4-11b) that was ignored during subsequent analysis of the compensated reflectance spectrum.

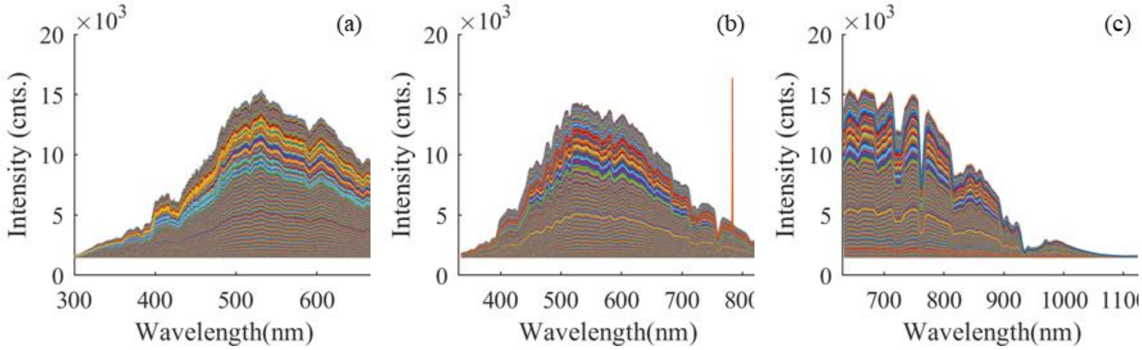


Figure 4-11 Raw reflected light measurements from a constant target during the eclipse for (a) UV; (b) VIS; (c) NIR spectrometers.

Calibrated ambient light and spectrometer integration times were combined with the raw reflectance intensity measurements using Equation 1 (Figure 4-12). The intensity unit changed accordingly to integer counts per microjoule of ambient light energy per millisecond of spectrometer integration time. The result of the compensation was a large reduction in variability centered around an average response with only a few outliers for each spectrometer. This meant that if a spectral measurement was taken from the reflectance target at a random ambient light condition, the compensated spectrum would have a high probability of being located near the average response.

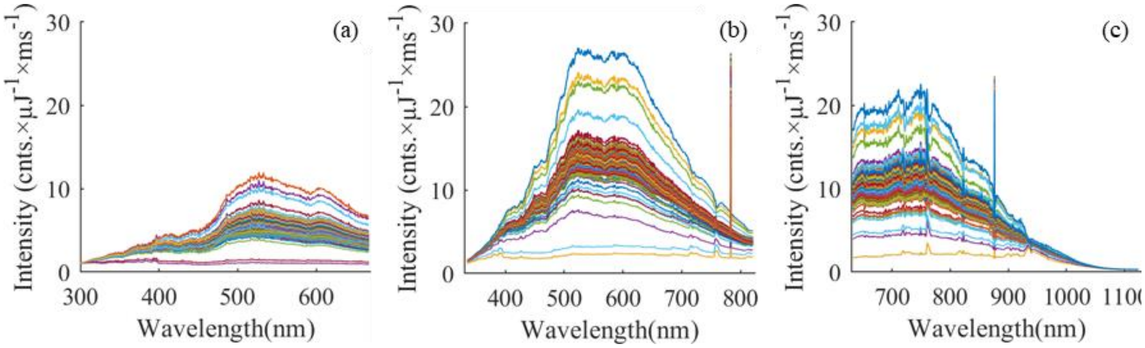
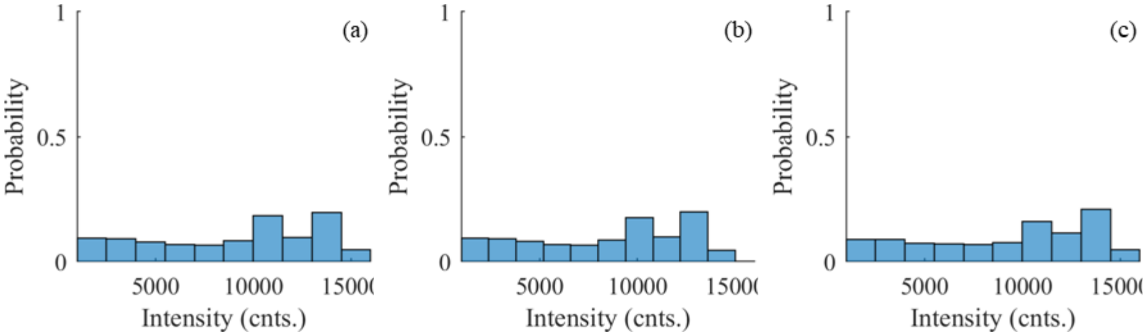


Figure 4-12 Compensated reflected light measurements from a constant target during the eclipse for (a) UV; (b) VIS; (c) NIR spectrometers

4.4.3 Compensation Evaluation

Discrete probability density functions (PDFs) were used to visualize the reduction in variability due to ambient light conditions. One wavelength was selected for each spectrometer, which had the highest intensity variability in raw reflectance measurements. Figure 4-13 shows how the compensation process improved the probability for classifying the reflectance target when considering a single wavelength. It was desired to have approximately the same relative intensity for different measurements taken regardless of ambient light conditions. It can be observed that the intensity is highly scattered in raw reflectance measurements; whereas, after compensation, the probability is high only at one or two intensity groups. In other words, since the target is constant, the compensated reflectance intensity varied little over a wide range of ambient light conditions. When applied to all wavelengths, a unique signature for the target would be defined and it could be recognized using this signature, regardless of ambient light conditions.



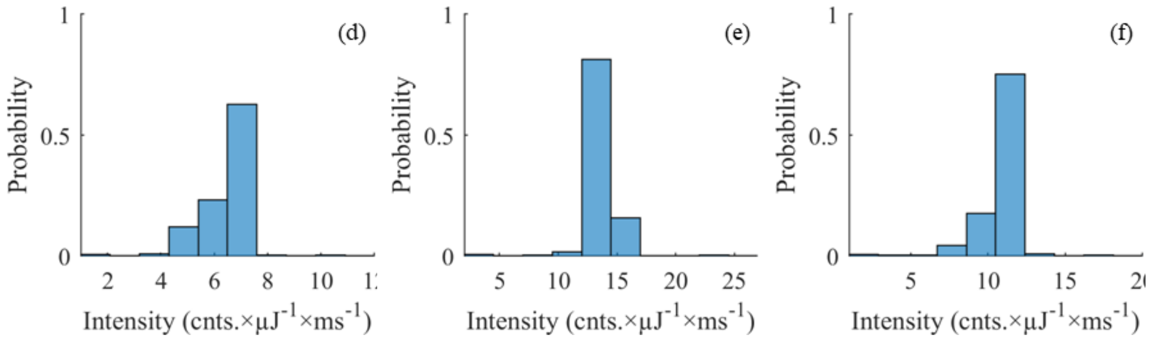


Figure 4-13 Discrete PDFs of the reflectance intensity from the constant target at the wavelength with the peak reflectance intensity: (a) UV, (b) VIS, and (c) NIR before compensation; and (d) UV, (e) VIS, and (f) NIR after compensation.

4.4.4 Validation of Ambient Light Spectrometers using Weather Station Pyranometer Data

A pyranometer mounted on a portable weather station was used as a benchmark for tracking changes in the ambient light system measurements during the eclipse. A linear regression model was used to determine the relationship between power per unit surface area from the pyranometer and instantaneous power from ambient light spectrometers and to model the cross-sectional area of the spectrometer at different times of measurement (Figure 4-14). A high coefficient of determination revealed the relationship was highly linear. (R-squared=0.97). Some hysteresis between the two sensors is evident as two distinct groupings that trended apart before and after the eclipse.

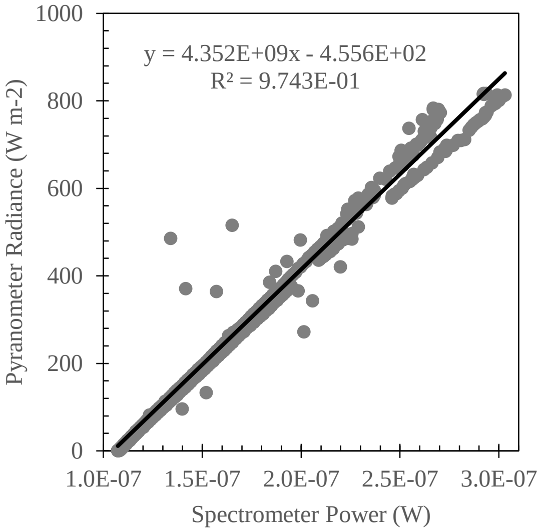


Figure 4-14 Linear regression between pyranometer radiance and ambient light spectrometer power

The linear regression model was used to represent the cross-sectional area of the spectrometers. Figure 4-15 illustrates how solar radiance varied during the eclipse using both ambient light spectrometers and the pyranometer. As expected, a similar pattern was observed between the two sensors with the largest deviations occurring at the beginning and end of the eclipse. The directional change in the offset is a result of the hysteresis exhibited in Figure 4-14. Momentary dips in solar radiance caused by transient clouds were detected by both systems, confirming they were not anomalies. The lowest solar radiance was measured around 14:30 EDTcorresponding to totality.

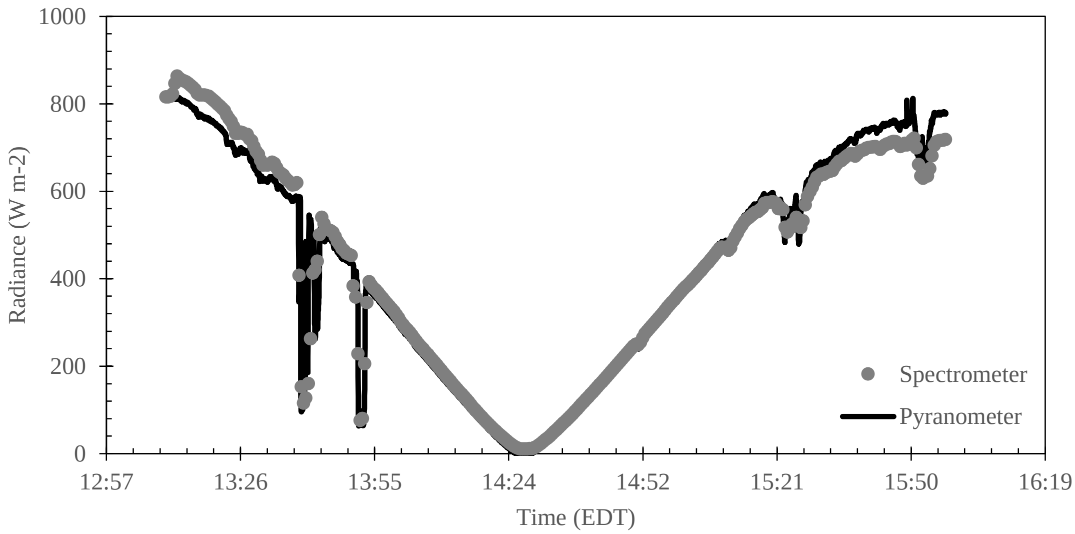


Figure 4-15 Comparing ambient light data collected by ambient light sensor on a weather station and ambient light spectrometers

4.5 CONCLUSION

In the first part of this study, a platform was developed to be deployed on a UAS to measure the reflectance intensity from a target. Two sets of portable STS spectrometers in three ranges of UV, VIS, and NIR were used along with a RPi to form a reflectance system and an ambient light system. In the second part of this study, a method for compensating for ambient light conditions and sensor integration time was developed and tested during the 2017 Great American Eclipse. Results showed a large variability in reflected light intensity due to significant changes in sun radiance. Reflectance values were compensated using ambient light measurements and integration time. Compensated reflectance values exhibited a consistent spectral signature for measurements taken at different ambient light conditions. This method will be useful for future field work where ambient light conditions cannot be controlled and the sensor integration time may need to be adjusted to optimize the sensitivity of the spectrometer. Future work should include testing the ability to classify different targets at varying ambient light conditions and to automatically adjust the integration time of each reflectance spectrometer based on previous measurements to maximize sensitivity.

CHAPTER 5: OBJECTIVE 4: CLASSIFYING REFLECTANCE TARGETS FROM HYPERSPECTRAL DATA COLLECTED UNDER AMBIENT LIGHT CONDITIONS USING A PASSIVE LOW-COST REMOTE SENSING SYSTEM

5.1 Summary

The main objective of this study was to develop a spectral measurement instrument for deployment on a small unmanned aircraft system (sUAS) and to test the ability of the system to classify distinct targets across a wide range of ambient light conditions. A series of portable spectrometers covering ultraviolet (UV), visible (VIS), and near-infrared (NIR) wavelengths were instrumented using an embedded computer and programmed to interface with the sUAS autopilot for autonomous data acquisition. A second set of identical spectrometers were fitted with calibrated irradiance filters to capture ambient light during data acquisition. This study aimed to determine the feasibility of using this low-cost method for classifying six grayscale reflectance targets under different ambient light conditions. Three compensation modes with variable integration time were developed to update integration time on the reflectance system based on ambient light conditions (M-1, M-2, and M-3). Sensor integration time was automatically updated after each measurement to optimize the subsequent measurement. Spectral data processing was conducted in two steps. First, raw spectral data were preprocessed using a partial least squares (PLS) regression method to compress highly correlated wavelengths and to avoid overfitting. Next, various machine learning algorithms were trained, validated and tested to determine the overall prediction accuracy of each algorithm for differentiating reflectance targets. The resulting compensated reflectance exhibited a consistent spectral profile and average intensity across a wide range of ambient light conditions for each target. Results indicated the potential for mitigating the effect of ambient light and optimizing integration time when passively measuring reflectance on a portable spectral measurement system. Eventually, it was observed that data collected with VIS spectrometer, with M-1 compensation mode, and using quadratic discriminant method provided a perfect target recognition.

5.2 INTRODUCTION

In precision agriculture, it is aimed to recognize the variability in field parameters using sensors before making decisions for applying agricultural inputs (Zhang & Kovacs, 2012). Remote sensing is currently among the most widely studied topics in precision agriculture (Mulla, 2013). For instance, unmanned aerial systems (UASs) are relatively new tools for being applied in remote sensing projects (Adão et al., 2017; Khanal et al., 2017) and have become very popular for agricultural applications. UAS-based farm studies have covered a wide range of applications including sensing biomass and nitrogen status (Hunt et al., 2005), monitoring wheat (Lelong et al., 2008), and monitoring rangelands (Rango et al., 2009). UASs provide a more versatile method for remote data collection with a high resolution compared to satellite and ground-based methods (Rudd et al., 2017).

In UAS-based projects, multispectral, thermal, or RGB cameras are most commonly deployed for monitoring a field and for estimating its parameters (Bendig et al., 2014; Berni et al., 2009; Hamidisepehr et al., 2017; Kelcey & Lucieer, 2012; Paredes et al., 2017). Traditionally, a couple of narrow-band ranges, which potentially have a high capability in estimating one or more field parameters, are selected (Candiago et al., 2015; Kalisperakis et al., 2015) to create an index like normalized difference vegetation index (NDVI). But in this way, the information that can be extracted from the other wavelength ranges are either filtered out or ignored. Most of the commercially available sensors are designed to work in one or two ranges of wavelengths to reduce sensor cost and data processing complexity.

Portable spectrometers are relatively inexpensive tools which can derive a complete spectrum from a broad spectral effective range and due to their small size they can be mounted on a UAS platform (Burkart et al., 2014; Von Bueren et al., 2015). In the both of these studies, two STS spectrometers were deployed. One spectrometer was oriented towards the ground and measured the reflectance from a reference white target. The other was mounted on a UAS to measure reflectance from actual land targets. The ratio of actual target reflectance and the reference target was considered as compensated reflectance from the target. Obtained spectrums can be analyzed partially or entirely to estimate different agricultural indices in a field such as NDVI, NDWI, and LAI. Nevertheless, calibrating these sensors for various ambient light conditions and avoiding saturation are challenges needed to be dealt with. Field spectrometers are mostly limited to data collection in a specific period and ambient light condition (Damm et al., 2011; Gao et al., 2002, 2004; Guanter et al., 2006). Using reference tarps is another common approach for compensating against ambient light changes. To keep track of ambient light changes continuously, a measurement from tarps needs to be taken for each measurement from a land target

(Shanahan et al., 2001). However, since it is practically difficult especially for large field scales, only a few measurements from tarps can typically be taken during the data collection process.

Using machine learning algorithms is a data processing method which provides an opportunity to process massive datasets like full spectra with a large number of input variables and samples and make a prediction model for unseen samples. Different machine learning algorithms have already been used for classification of hyperspectral images (Melgani & Bruzzone, 2004), weed detection (Koot, 2014), plant disease detection (Rumpf et al., 2010), biotic stress detection (Behmann et al., 2015), water quality monitoring (Kim et al., 2014), human learning (Matveeva et al., 2016; Mousavi et al., 2016), and many other applications. Several studies focused on developing algorithms and methods for feature selection to reduce the dimensionality of very large datasets (Serpico & Bruzzone, 2001; Serpico & Moser, 2007). By compressing the dataset, the other issue derived from large spectral data can be addressed (Ye et al., 2017).

In this study, a data acquisition system for collecting hyperspectral data consisted of two sets of STS spectrometers coupled with Raspberry Pi (RPi) embedded computers were used (Hamidisepehr & Sama, 2018). This study aimed to expand upon previous work by devising a method to compensate the portable spectrometers integration time (measurement period) against varying ambient light conditions by updating integration time for each measurement. Designing a dynamic compensation process on a UAS platform would enable spectral data to be collected over a wide range of ambient light conditions with limited impact on sensor sensitivity. Specific objectives included:

1. Develop a variable integration time to automate compensating reflectance measurements against ambient light on the spectral data acquisition system from Chapter 4.
2. Test the system on multiple targets and assess classification accuracy on multiple models generated by different machine learning algorithms.

5.3 Materials and Methods

5.3.1 Hardware Setup

Two data acquisition systems were deployed for data collection process – an updward-facing ambient light system for measuring ambient light intensity and a downward-facing reflectance system for measuring reflectance from a target located underneath sensors. Each system consisted of three STS spectrometers (STS, Ocean Optics, Dunedin, FL) in the ultraviolet (UV), visible (VIS), and near-infrared (NIR) ranges as data collection devices and a Raspberry Pi 3 (RPi) (B V1.2, Raspberry Pi Foundation, Cambridge, United Kingdom) as an embedded data acquisition system for controlling the measurement process. A test stand was used to hold both systems and target consistently relative to each other.

5.3.2 Reflectance Targets

Five 0.3 m square plywood targets painted in varying shades of gray, and one target laminated with a 0.8 mm thick sheet of PTFE, were fabricated as reflectance targets to be placed underneath the spectrometers. Each was painted with a different color in grayscale range (Figure 5-1). The relative reflectance of each target was measured using visible and near-infrared spectrometers (HR400-7-VIS-NIR, NIRQuest512; Ocean Optics; Dunedin, FL) with a Spectralon calibration target (WS-1-SL; Ocean Optics; Dunedin, FL) serving as the calibration reference target. Nine spectral measurements were taken at uniformly spaced locations and averaged.

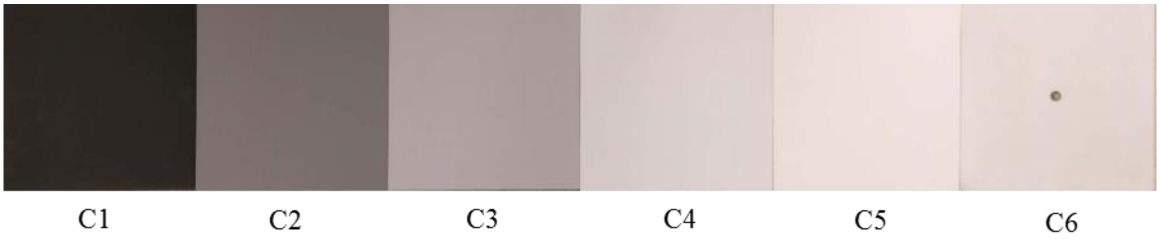


Figure 5-1: Reflectance targets in the greyscale range

5.3.3 Data Collection

The data were collected at five days on September 14, 15, 18, 19, and 21 of 2017 on the roof of Charles E. Barnhart Building in Lexington, Kentucky. The test stand was oriented so that shadows would not be made on the targets since it would change reflectance spectrum. Data were collected periodically in ten-second intervals. Each measurement interval included three individual measurements that were stored in a tab-delimited text file. Each file includes the time of measurement and the serial number of the spectrometer to facilitate tracking measurements with different time and with different spectrometers.

5.3.4 Updating Integration Time

Integration time is the amount of time that a spectrometer sensor is exposed to light. Increasing the integration time has a similar effect to applying a gain to the spectral signal, making patterns or unique features more discernable. Increasing integration time by an excessive amount, however, reduces sampling rate and will eventually cause saturation in spectral data at one or more wavelengths. A saturated measurement is not useful for signal classification. Hence, the optimal situation is for each measurement to be taken with the maximum integration time that does not result in saturation.

Reflectance intensity varies when the ambient light condition changes e.g. due to clouds and sun angle. A fixed target will produce varying spectral signatures using a spectromter if integration time is set constant. In order to set an appropriate integration time for each specific ambient light condition, a method to update integration time based on the ambient light condition and the spectral response from the last measurement was used. In other words, each measurement was considered as a feedback for the next measurement. In this process it was assumed that subsequent measurements are observing targets with similar spectral reflectance.

The process started with setting an initial integration time on each reflectance spectrometer and recording a measurement. A continuously running Python script then read in the most recent measurement. Outliers in the spectral data due to hot and dead pixels at certain wavelengths were detected and removed. The maximum intensity of the spectrum was determined and compared to the maximum possible intensity without saturation (2^{14}-

1 = 16,383). The maximum intensity was set to 12,000 counts as a safe threshold. Thus, the updated integration time for a given target was calculated from equation 5-1.

$$IT_{k+1} = \frac{M_k}{M_{max}} * IT_k \tag{5-1}$$

where:

IT_{k+1} was the integration time for the next spectral measurement from a given target (ms)

M_k was the maximum raw measurement intensity in the current spectrum from a target (A/D counts)

M_{max} was the maximum possible raw measurement intensity without saturation (A/D counts)

IT_k was the integration time for the current spectral measurement intensity from a given target (ms)

It was assumed that there is a linear relationship between integration time and reflectance intensity. By updating the integration time based on the ambient light condition, it is expected that the reflectance intensity spectrum ideally becomes the same for different ambient light conditions if the same target is measured. Figure 5-2 shows hypothetically how a spectrum changes by updating integration time. Hence, by any detectable change in ambient light, the integration time would be updated for the next measurement.

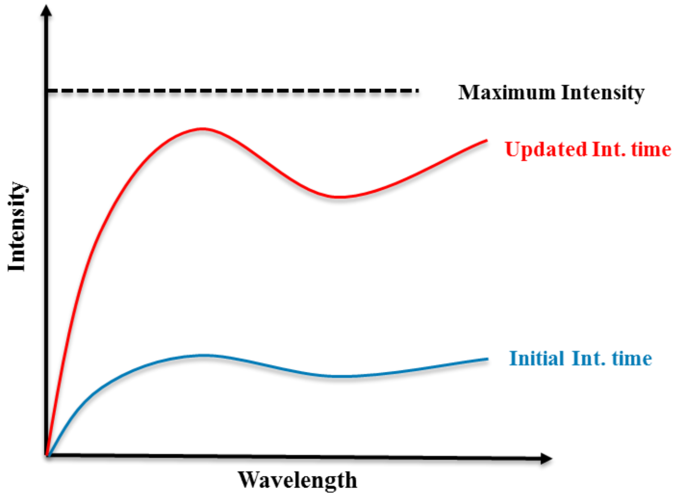


Figure 5-2: Schematic of updating integration time based on initial and maximum reflectance intensity

On the other hand, integration times for ambient light spectrometers were kept constant. These integration times were defined by the manufacturer and set to one second on both UV and NIR spectrometers, and 180 milliseconds for VIS spectrometer to measure solar irradiance.

5.3.5 Compensating for Ambient Light

Data from two sets of spectrometers for each target were collected over five days. Since the integration time on ambient light and reflectance spectrometers were different, the measurements were not temporally synchronized. Due to longer integration times on ambient light spectrometers, fewer measurements were obtained compared to reflectance measurements. The ambient light measurements were interpolated to the moments when reflectance was measured. Ambient light measurements were calibrated using coefficients for different wavelength provided by the manufacturer. After applying coefficients (counts / J), the intensity unit changed from counts to microjoules (μJ) (Equation 5-2).

$$CA_{\lambda} = \frac{A_{\lambda}}{coeff._{\lambda}} \tag{5-2}$$

Where:

CA was the calibrated ambient measurement intensity (J)

A was the raw ambient light measurement intensity (A/D counts)

λ was the specific wavelength (nm)

Three compensation modes were considered for the calibrating reflectance measurements and each mode was evaluated based on the predictive power of generated models 164):

1. Raw reflectance (counts) as M-1
2. Dividing reflectance data by its corresponding integration time (counts/ms) as M-2 (equation 5-3)

$$R_{\lambda} = \frac{I_{\lambda}}{IT} \tag{5-3}$$

Where:

R was the calibrated reflectance measurement intensity (counts/ms)

I was the raw reflectance measurement intensity (A/D counts)

IT was the integration time in the specific measurement (ms)

λ was the specific wavelength (nm)

3. Dividing reflectance data by ambient light measurement taken at the same moment and the corresponding integration time of the reflectance spectrometer (counts×μJ^{-1}×ms^{-1}) as M-3 (equation 5-4).

$$R_\lambda = \frac{I_\lambda - 1500}{CA_\lambda * IT} \tag{5-4}$$

Where:

R was the calibrated reflectance measurement intensity (counts×μJ^{-1}×ms^{-1})

I was the raw reflectance measurement intensity (A/D counts)

IT was the integration time in the specific measurement (ms)

CA was the calibrated ambient measurement intensity (μJ)

λ was the specific wavelength (nm)

5.3.6 Preprocessing

Each spectrometer covered a range of several distinct wavelengths (UV between 184nm-667nm, VIS between 338nm-825nm, and NIR between 634nm-1124nm and with 0.5nm step) which each can be considered as an input variable in a predictive model. Since many of the wavelengths are highly correlated, they can be combined to reduce the dimensionality of the dataset. A preprocessing method, partial least square (PLS) regression, was used to compress dataset, solve collinearity issues, speed up subsequent processing. The optimal number of components after preprocessing was obtained using two parameters, i.e. the estimated mean squared prediction error and the variance explained in the output variable. The number in which a high variance in output was explained with a low prediction error was considered as the optimal number of input components. The preprocessing step, including compressing the dataset and finding the optimal number of components, was conducted using MATLAB (R2017a; The Mathworks; Natick, MA).

5.3.7 Machine Learning

Using the Classification Learner app in MATLAB (R2017a), 22 leaning algorithms, including decision trees, discriminant analysis, support vector machines (SVM), nearest neighbor classifiers, and ensemble classifiers, were used to train models for identifying targets based on their spectral measurements. The data was fed into individual algorithms as a matrix where columns represented wavelengths (predictors) and rows represented instances of each measurement. The last column (response) was allocated to target codes (C1 through C6). The dataset was divided to: 70% training dataset, 15% validation dataset, and 15% for testing dataset. For each measurement in training dataset, predictors or input variables were paired with response or actual output. It is necessary to have the output of every row data for supervised learning. The validation dataset was used to determine how well the model has been trained based on the predicting expected output. Model properties, such as classification error and overfitting index are estimated during the validation step. Finally, in the testing dataset, unseen data to the machine are applied, and the prediction power of a model is estimated by comparing the correct output and the predicted output. By doing these three steps for each learning algorithm, 22 models were trained, and their prediction accuracies were obtained. Each treatment was run five times with randomly distributed training, validation, and testing data to find the best algorithm which has a higher accuracy in different combinations of spectrometer type and compensation mode.

5.3.8 Statistical Analysis

Three spectrometers and three compensation modes were considered in this experiment. It was desired to see if there were any significant differences between various types of spectrometers and the methods of compensation in terms of prediction accuracy in target recognition. This test shows in which mode of compensation and with which spectrometer type, the prediction accuracy is higher than others and also if the difference between groups is significant. Then, the optimal algorithm was tested for each combination of compensation mode and spectrometer type to check the significance of these two parameters on the predictive power of the model. The experiment was set up with a factorial design using spectrometer type and compensation mode (3×3). The data were subjected to analysis of variance and a multiple comparisons test was conducted using MATLAB (R2017a). At the end of this analysis, the spectrometer type and compensation mode which provided higher overall accuracy were chosen as the optimal selection. The null hypothesis was that there is no significant difference between spectrometer type and compensation mode with the prediction accuracy of the optimal model.

5.4 RESULTS AND DISCUSSION

5.4.1 Benchmarking

Targets were benchmarked using the reference target so that the reference target reflected nominally 100 percent at all wavelengths after laboratory spectrometer calibration. Figure 5-3 shows how different grayscale targets reflected a constant light versus the reference target and when the light source was blocked (background). Darker targets clearly reflected less compared to lighter targets. Targets C4 and C5 resulted in relative reflectances in excess of 100%, indicating they were "brighter" than the calibration standard over a range of wavelengths. A consistent trend across all targets was a decrease in relative reflectance as wavelength increased.

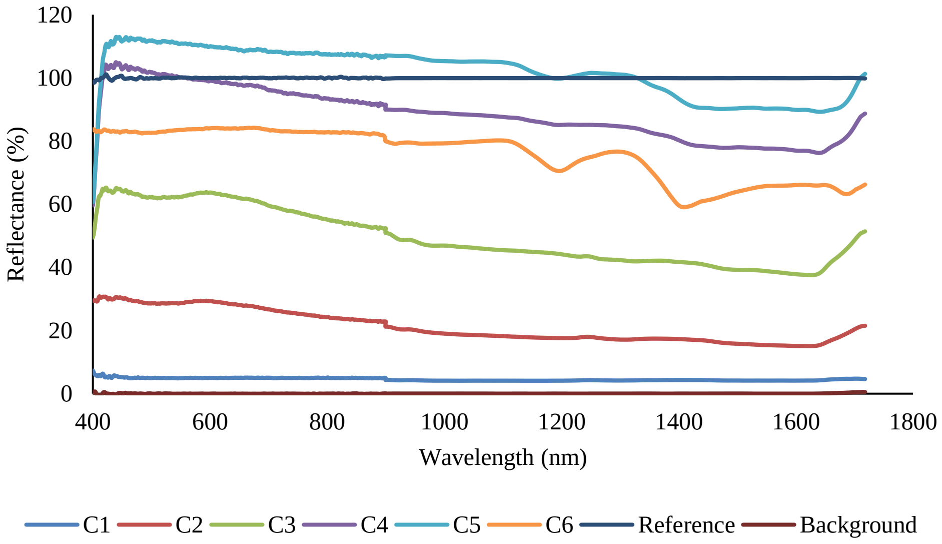


Figure 5-3: Spectrums of reflectance targets with lab spectrometers calibrated with the Spectralon reference target

5.4.2 *Ambient and Reflectance Measurements*

In this section, spectrums obtained from each mode of compensation for the UV spectrometer are shown. Due to the similarity of the process, the compensation process only on one spectrometer type has been shown. Data from the other two spectrometer types are located in 174. Figure 5-4 shows the raw ambient light spectra. It can be observed that there was large variability in ambient light condition due to measurement at different ambient light conditions. This variability directly affects reflectance measurements, which illustrates the necessity of compensating reflectance measurements against ambient light conditions.

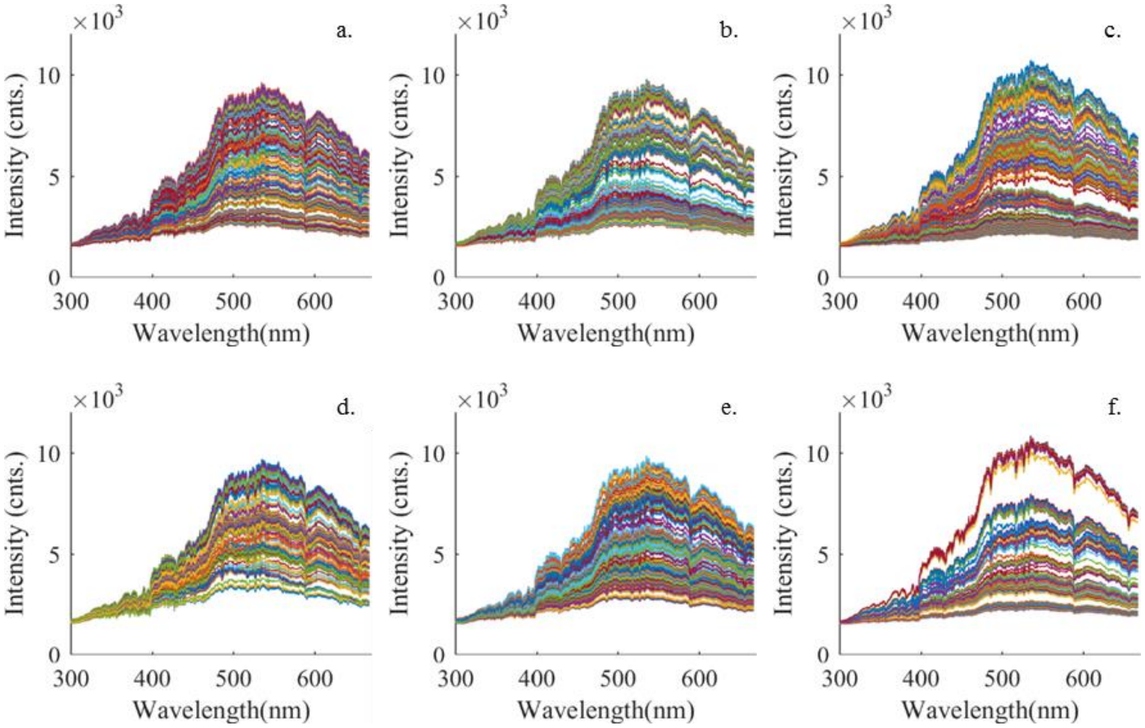


Figure 5-4: Raw ambient light measurements from UV spectrometer collected during reflectance measurement for six targets (a. darkest target, e. lightest target, f. white PTFE target)

In the next part, raw reflectance data with variable integration time are shown (M-1) (Figure 5-5). Reflectance spectra were filtered to skip saturated measurements and low-intensity spectra. Saturation happened just a few times during the data collection at the moments when the target was switched from a dark target to a brighter one. Since the integration time is longer on darker targets, it takes one or more measurements to adjust the integration time with a brighter target. Also, switching from a bright target to a darker one leads to low-intensity spectrums for first few measurements which causes lower sensitivity in obtaining information. Since integration time is adjusted to compensate for varying ambient light, all spectra exhibited similar average intensities between targets.

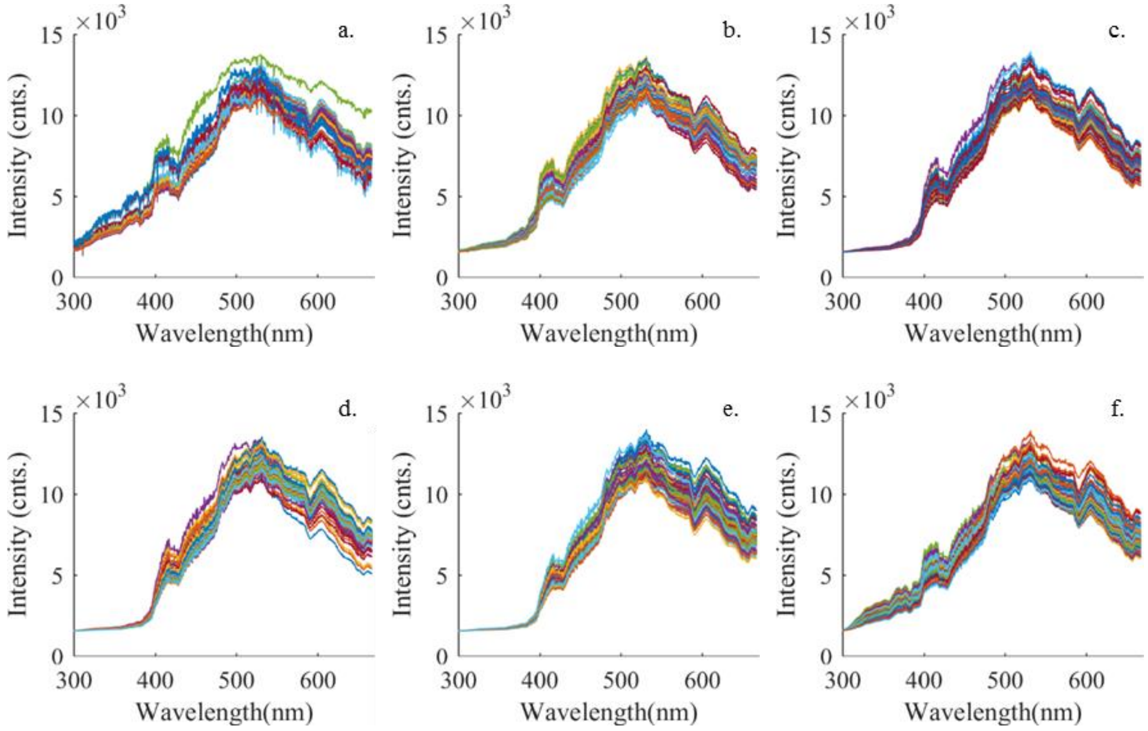


Figure 5-5: Reflectance measurements with variable integration time with M-1 mode of compensation for six targets of: a. C1; b. C2; c. C3; d. C4; e. C5; f. C6.

By dividing each raw reflectance measurement by its corresponding integration time (M-2), data are scattered. It was now easier to visually distinguish dark and light targets, although, identifying light targets of C4, C5, and C6 was still difficult due to similar spectra (Figure 5-6).

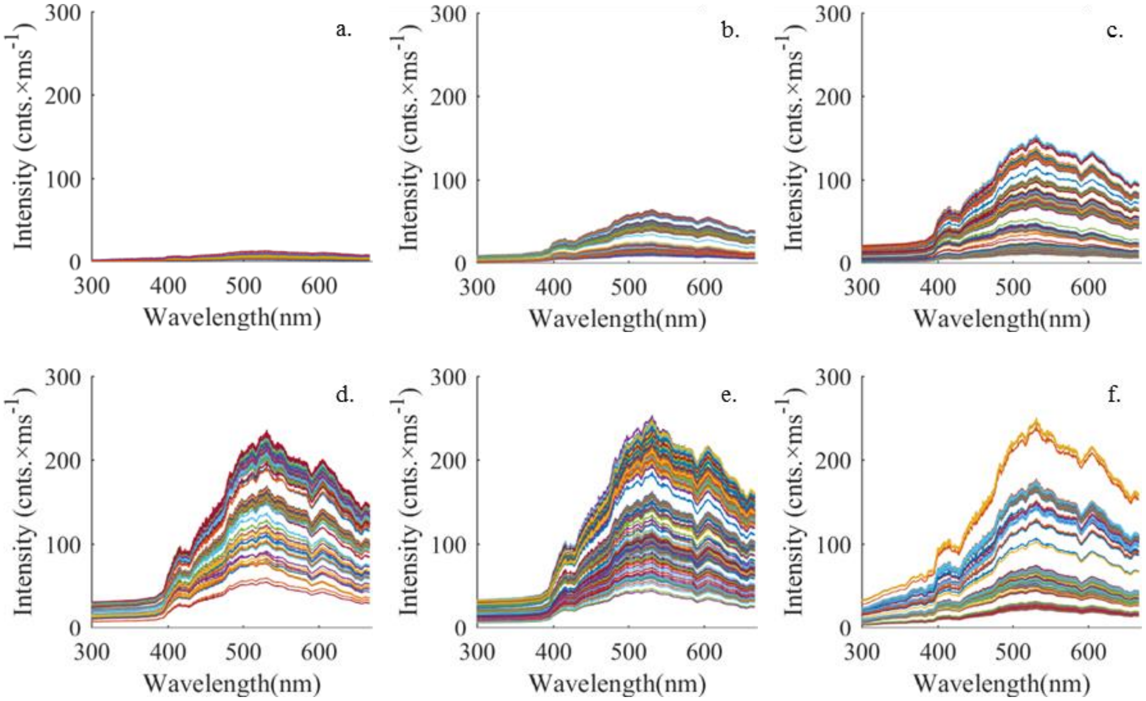


Figure 5-6: Reflectance measurements with variable integration time with M-2 mode of compensation for six targets of: a. C1; b. C2; c. C3; d. C4; e. C5; f. C6.

Next, each raw reflectance measurement was divided by its corresponding integration time and simultaneous ambient light measurement (M-3). According to Figure 5-7, darker targets were more easily distinguished while brighter targets have multiple similar spectra like the previous step and with a noticeable overlap. Target C4 was now more distinct from C5 and C6 than in the previous method.

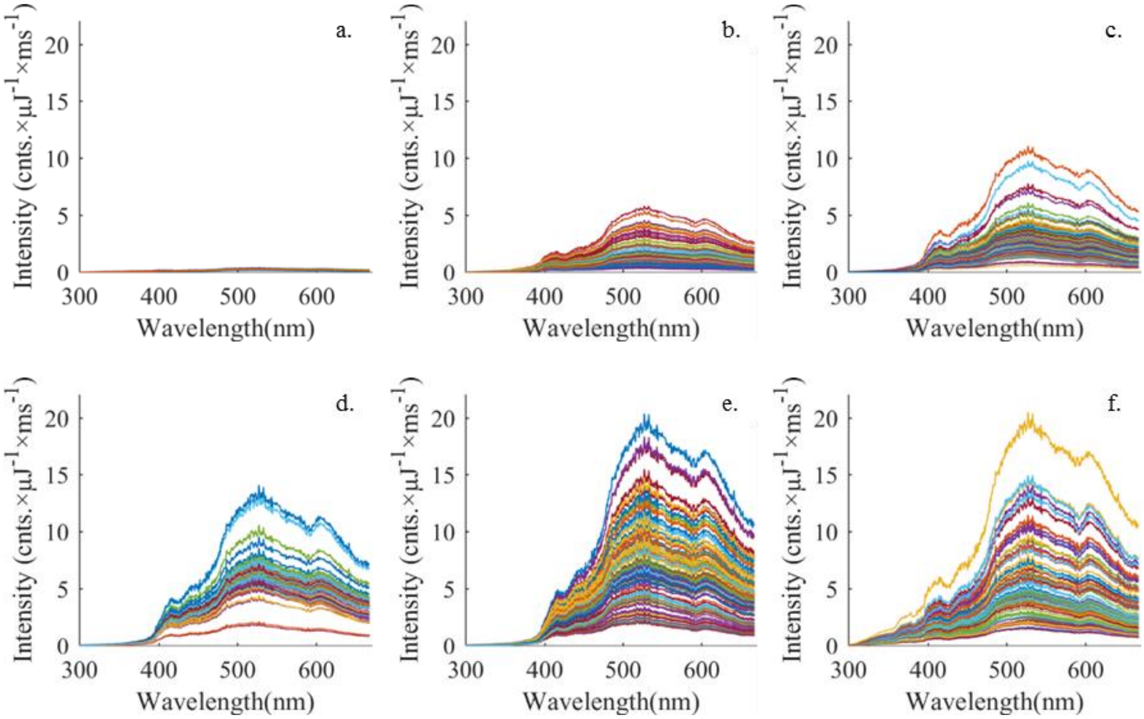


Figure 5-7: Reflectance measurements with variable integration time with M-3 mode of compensation for six targets of: a. C1; b. C2; c. C3; d. C4; e. C5; f. C6.

5.4.3 Preprocessing

Spectral data in different modes of compensation were modeled using machine learning algorithms to evaluate how accurate an algorithm can estimate the target based on existing training dataset. Before applying machine learning algorithms, estimated mean squared prediction error and percent variance explained in the output were used as decision criteria to find the optimal number of components for target recognition. As expected, there were many highly correlated wavelengths that could be combined to compress the dataset and make the subsequent data processing faster and to avoid overfitting. Based on Figure 5-8, a model with around 20 components would result in a low estimated mean squared prediction error and also about 90 percent of the variability in the output can be explained. For components more than 20, amount of error increases due to overfitting. On the other hand, increasing the number of components, increases the amount of variance explained in general. For more than 20 components, however, there is only a slight increase in variance explanation by adding many components. Our models were tested with a few more and less components than 20 to make sure about the optimal number. Based on these preliminary results, 20 was considered as the optimal number of components to feed into learning algorithms.

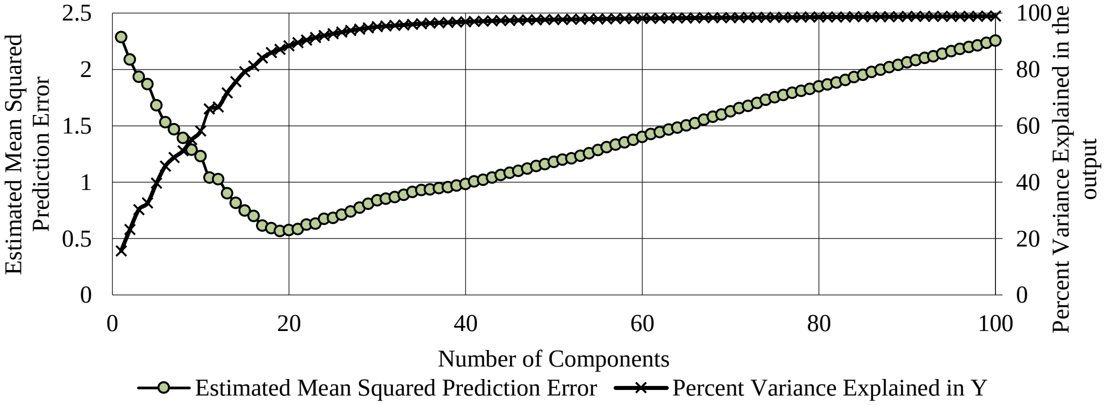


Figure 5-8: Estimated error and the variance explained in the output versus number of components in a model

5.4.4 *Machine Learning Algorithms*

Twenty-two pre-configured machine learning algorithms were used to train models for each combination of compensation mode and for each spectrometer type. Figure 5-9 shows the prediction accuracy of models generated from the UV spectrometer. It can be observed that Quadratic Discriminant, Linear Discriminant, Linear SVM, Quadratic SVM, and Cubic SVM are all accurate models; however, Quadratic Discriminant is slightly more accurate than others for all three compensation modes. Each model was generated ten times with random selection of training and testing dataset and the average prediction accuracy for each model was obtained.

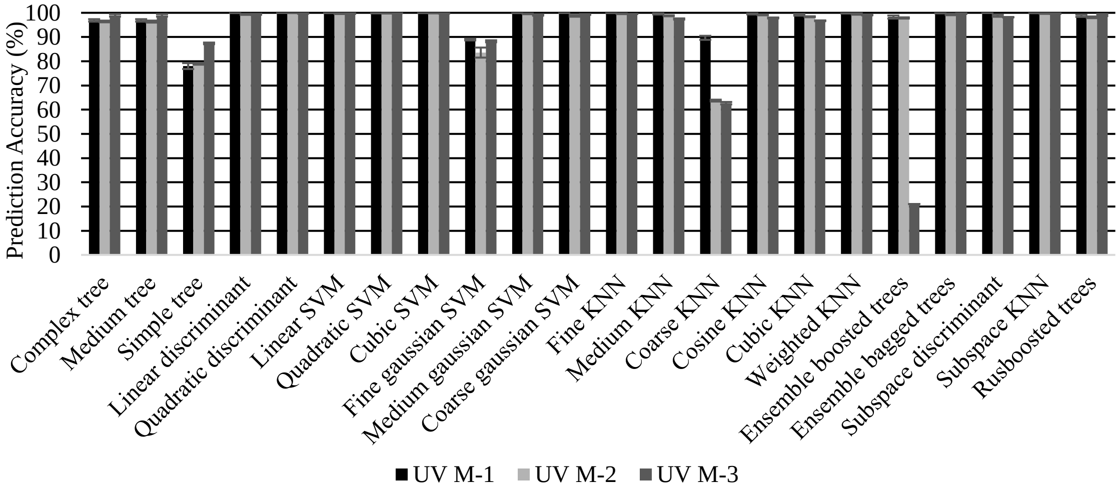


Figure 5-9: Prediction accuracy for 22 machine learning algorithms applied to relative reflectance data collected by the UV spectrometer from reflectance targets and for three compensation modes.

Figure 5-10 shows the models generated from data collected with VIS spectrometer for three compensation modes and again Quadratic Discriminant provided the highest accuracy with a slight difference for three compensation modes.

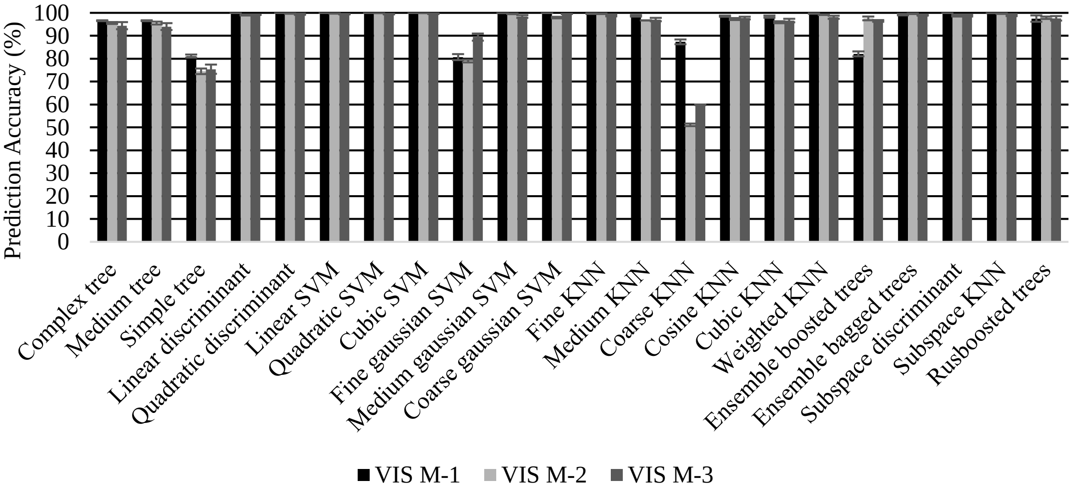


Figure 5-10: Prediction accuracy for 22 machine learning algorithms applied to relative reflectance data collected by the VIS spectrometer from reflectance targets and for three compensation modes.

The prediction accuracy of models generated from data collected by NIR spectrometers is shown in Figure 5-11. The prediction accuracy was lower compared to models from UV and VIS spectrometers data. Quadratic SVM had a slightly higher performance in terms of prediction accuracy.

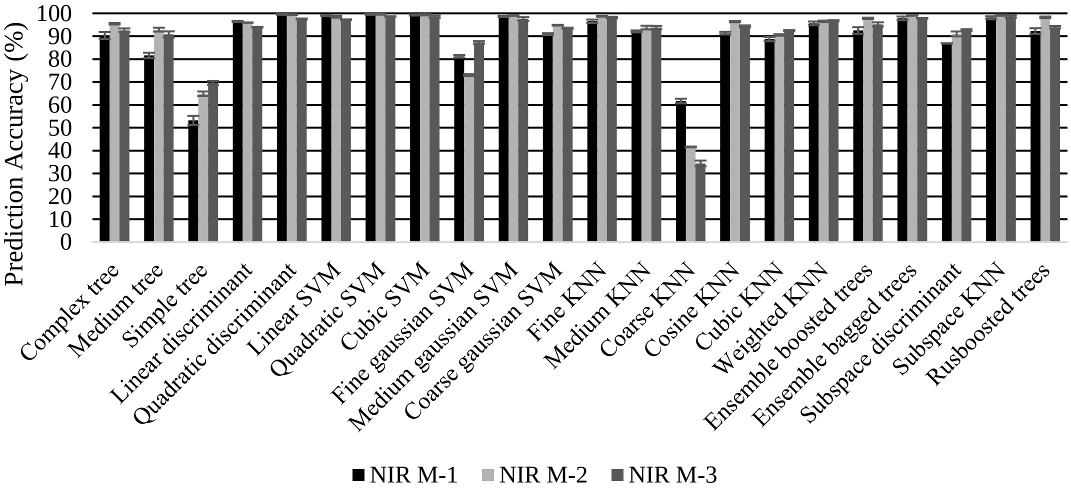


Figure 5-11: Prediction accuracy for 22 machine learning algorithms applied to relative reflectance data collected by the NIR spectrometer from reflectance targets and for three compensation modes.

Table 5-1 contains the average of each model for different compensation modes and spectrometer type. Each combination of spectrometer type and compensation mode was repeated five times to provide replications.

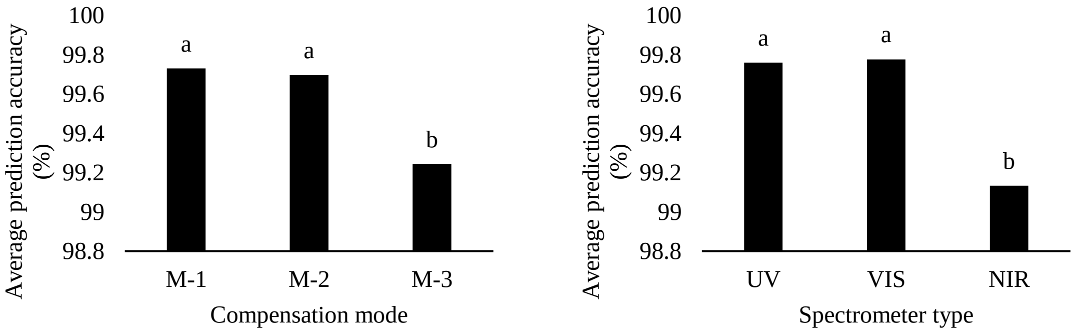
Table 5-1: The prediction accuracy for 22 machine learning algorithms in target recognition and for different compensation modes and spectrometer types

Algorithm	M-1			M-2			M-3		
	UV	VIS	NIR	UV	VIS	NIR	UV	VIS	NIR
Complex tee	96.9	96.6	90.3	96.4	95.6	95.5	98.9	94.4	92.6
Medium tree	96.9	96.6	81.6	96.4	95.5	92.9	98.9	94.0	90.8
Simple tree	78.0	81.1	53.2	78.9	74.5	64.9	87.4	75.4	69.6
Linear discriminant	100	100	96.4	99.3	99.0	95.9	99.4	99.2	93.9
Quadratic discriminant	100	100	99.5	99.9	99.9	99.4	99.7	99.4	97.5
Linear SVM	99.9	100	99.0	99.6	99.8	98.5	99.7	99.4	97.2
Quadratic SVM	99.9	100	99.5	99.8	99.9	99.7	99.9	99.4	98.6
Cubic SVM	99.9	100	99.3	99.9	99.9	99.4	99.7	99.6	98.5
Fine Gaussian SVM	88.9	80.7	81.2	83.6	78.9	73.0	88.2	89.4	87.3
Medium Gaussian SVM	99.9	100	98.6	99.5	99.5	99.1	99.1	98.4	97.5
Coarse Gaussian SVM	100	100	90.9	98.7	97.9	94.8	99.3	99.5	93.5
Fine SVM	99.9	99.8	96.5	99.6	99.6	98.7	99.5	98.9	98.1
Medium KNN	99.6	98.7	92.1	98.8	96.7	93.8	97.4	97.1	93.8
Coarse KNN	89.7	87.3	61.7	63.7	51.1	41.6	62.7	59.8	34.4
Cosine KNN	99.5	98.5	91.3	99.1	97.3	96.4	97.9	97.7	94.3
Cubic KNN	99.2	98.3	88.7	98.4	95.9	90.5	96.7	96.6	92.4
Weighted KNN	99.8	99.8	95.6	99.5	99.2	96.5	99.0	98.1	96.8
Ensemble boosted trees	98.2	86.8	92.5	97.9	97.6	97.8	21.0	96.5	95.2
Ensemble bagged trees	99.9	99.2	97.8	99.3	99.6	99.1	99.6	99.0	97.8
Subspace discriminant	100	100	86.9	98.8	98.8	91.0	98.2	98.8	92.7
Subspace KNN	99.9	100	98.1	99.7	99.6	99.2	99.7	99.0	98.5
RUSboosted trees	98.8	97.5	92.3	98.1	97.9	98.3	99.1	97.6	93.9

In the next step, a statistical analysis was conducted to see if the effect of different compensation modes and the type of spectrometer had significant impact on overall prediction accuracy. Based on Table 5-2, both of these two variables had a significant effect on prediction accuracy and the null hypothesis was rejected because of the low p-value. Also, the results of the *multiple comparison test* in MATLAB (Figure 5-12-a) showed that the difference between M-3 with M-1 and M-2 was significant. But there was no significant difference between M-1 and M-2; however, M-1 provided a slightly higher prediction accuracy. The NIR spectrometer had a lower overall prediction accuracy on different compensation modes and it can be observed from the *multiple comparison test* that there is a significant difference between NIR and both VIS and UV. No significant effect was observed between UV and VIS spectrometers; however, VIS spectrometer data provided a slightly more accurate models than UV for target recognition (Figure 5-12-b).

Table 5-2: Significance testing of compensation mode and spectrometer type on overall accuracy of the predictive model

Source	Sum of square	df	Mean square	F	Prob>F
Compensation mode	4.43	2	2.21	117.31	1.19e-24
Spectrometer type	8.068	2	4.03	213.44	5.13e-33

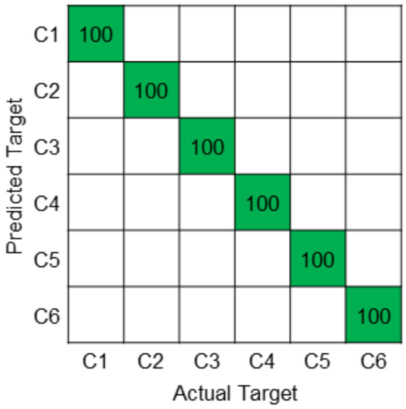


a. b.

Figure 5-12: Multi-comparison significance test between different a. compensation modes b. spectrometer type

Based on the overall data analysis, the highest prediction accuracy was obtained using data collected with the VIS spectrometer and applying M-1 compensation mode and with a model generated using the Quadratic Discriminant algorithm.

The confusion matrix in Figure 5-13 demonstrates more details about the predictive power of the optimal model derived from reflectance measurements with M-1 compensation mode obtained by the VIS spectrometer and using Quadratic Discriminant method. In this matrix, rows represent the actual target and columns represent the predicted target. If a specific prediction was correct on a target, then it is placed on the diagonal of the matrix. The cells not on the diagonal, however, show the inaccuracies in predictions. The number on each cell indicates the percentile of the frequency of a specific prediction. On the optimal model, no inaccuracy has found all predictions matched with the actual target. It is worth noting that this experiment used idealized targets that were uniformly distinct across a wide range of wavelengths. Identical performance should not be expected with observing more “natural” targets. The optimal sensor type and modeling method may also vary based upon the target.



	C1	C2	C3	C4	C5	C6
C1	100					
C2		100				
C3			100			
C4				100		
C5					100	
C6						100

Figure 5-13: Confusion matrix for estimating reflectance target using Quadratic Discriminant algorithm

5.5 Conclusions

Six targets were provided with differences either in colors or material in the grayscale range. Then, these targets were benchmarked using visible and near-infrared spectrometers calibrated with a reference target. Two data acquisition systems (reflectance and ambient light) were used for collecting spectral data at five days at different times of day to cover a large portion of ambient light variability in the spectral dataset. A system was designated for measuring ambient light, and sun radiance and the other series measured the reflectance from targets. A mechanism was applied to update the integration time of each reflectance spectrometer based on ambient light condition. Then, spectral data were compensated for ambient light condition in three different modes; raw reflectance (M-1), reflectance divided by the corresponding integration time (M-2), and reflectance divided by the corresponding integration time and ambient light (M-3). Twenty-two learning algorithms were used to generate models for recognizing reflectance targets for each combination of spectrometer type and compensation mode. Most of the algorithms had a prediction accuracy over 90%. The Quadratic SVM model generated from VIS spectrometer data with M-1 compensation mode provided the maximum prediction accuracy (100%). Based on a statistical analysis, it was found out that both spectrometer type and compensation mode have a significant effect on the prediction accuracy of targets. Also, the difference between NIR spectrometer with UV and VIS was significant unlike the difference between VIS and UV. M-3 was significantly different from M-1 and M-2 while the M-1 and M-2 were not significantly different. It was concluded that by adjusting integration time based on ambient light conditions, machine learning models could provide a sufficiently high accuracy for recognizing targets according to their spectral responses.

CHAPTER 6: OBJECTIVE 5: MOISTURE CONTENT CLASSIFICATION OF SOIL FROM SPECTRAL DATA COLLECTED UNDER AMBIENT LIGHT CONDITIONS USING A PASSIVE LOW-COST REMOTE SENSING SYSTEM AND MACHINE LEARNING

6.1 SUMMARY

Estimating soil/crop parameters such as soil water content using remote sensing under ambient light condition is challenging and often involves a complicated calibration process. The objective of this study is to use a novel hyperspectral data acquisition system, including UV, VIS, and NIR spectrometers, developed from a previous study for estimating the soil moisture level under ambient light condition. To automate compensating against ambient light changes, a technique for updating integration time during data collection was deployed. 21 moisture-controlled sample chosen from 7 moisture levels were measured at different times over two days. The data collection on each sample was 20 minutes. To keep track of effect of water content change during data collection, the data collection period was divided to three periods of 5, 10, and 20 minutes. A preprocessing step was conducted to compress the dataset using PLS regression method. Then, preprocessed data were fed into 22 machine learning algorithms and prediction accuracy of each model with data collected from each spectrometer and each data collection period was obtained. It was found out that linear discriminant on the models generated with a 10-minute period of data collection performed the best.

6.2 INTRODUCTION

During last few decades, agricultural production has significantly increased while the cropland extent has either remained the same or only increased by a small percentage (Gleick, 2003; Ozdogan et al., 2010). Intensive agricultural production is the primary consumer of fresh water (Rosegrant et al., 2008) and irrigation puts more pressure on water resources in a specific area to meet water needs (Cai & Rosegrant, 2002). Instead of applying water uniformly over a field, it is desirable to irrigate site-specifically if the field soil is spatially variable. Variable rate irrigation is an effective method to optimize the water usage during irrigation and applying water at the right amount and in the right place. Hence, tracking soil moisture cross a field spatially and temporally at sufficient resolution would be desired. Soil moisture is also associated with nutrient availability for plants and overall field performance (Khanal et al., 2017). Generating prescription maps for entire fields using spectral remote sensing data has been a popular approach in order to implement variable rate irrigation. There are three main platforms for collecting spectral data: satellites, conventional aircraft, and unmanned aircraft systems (UAS). Regardless of the type of platform, soil moisture is a primary component that affects spectral response (Lobell & Asner, 2002; Rossel et al., 2006). Each of these platforms was deployed in several studies and experiments.

Satellite and conventional aircrafts provide data at a lower resolution for field scales. There are also temporal and cost limitations for data collection from these two platforms (Montes de Oca et al., 2018). On the other hand, UAS are a cost effective method that are evolving rapidly and can provide data at high spatial and temporal resolutions

(Hakala et al., 2018). UAS flight time is a major limitation on large scale fields – especially, for heavy payloads (Gnyp et al., 2016).

Spectral remote sensing data are collected through either hyperspectral or multispectral sensors. Multispectral sensors measure reflected light in a few certain wavebands in a wide spectral range (Bokolonga et al., 2016; Rabatel et al., 2014) while hyperspectral sensors or spectrometers collect data at many wavelengths in a broad spectral range (Lee et al., 2010). Multispectral sensors are common since they create a smaller dataset which facilitates the data processing compared to the larger datasets obtained from spectrometers and very large datasets obtained from hyperspectral cameras. On the other hand, hyperspectral data contain more information which can be used for monitoring different field parameters simultaneously while multispectral data is limited to measure few parameters and with lower spectral details. In a study by Gnyp et al. (2016), tractor-based and UAS-based spectrometer data were compared, and it was concluded that both systems have the potential for monitoring nitrogen status in a winter wheat field. Portable spectrometers are suitable for UAS deployment at relatively low cost. STS spectrometers mounted on a UAS were tested in multiple studies for hyperspectral measurements and compared with conventional field spectrometers. Promising results were shown for further experiments (Burkart et al., 2014; Tsouvaltsidis et al., 2015; Von Bueren et al., 2015). In a study by Zeng et al. (2017), spectral data from a portable spectrometer was fused with multispectral camera images to provide expanded spectral information for at each pixel.

Multiple studies were conducted for estimating soil moisture based on the spectral analysis in which a reference measurement before each reflectance was necessary for field condition unless ambient light is controlled in a lab condition (Hamidisepehr et al., 2017; Kaleita et al., 2005). Ambient light changes during spectral data collection is a challenge that must be addressed before using the data to make management decisions. Transient clouds and changes in sun angle cause changes in measured spectra that do not correspond to changes in the parameter of interest. Using calibration tarps in the field during data collection as a reference is a possible solution for normalizing reflectance data; however, these targets should be included in all measurements for an accurate ambient light tracking.

Recently, several machine learning methods are more commonly used in the literature due to their capability for handling datasets with high dimensionality. Among these algorithms, some of them have more popularity for remote sensing data on agricultural applications. For instance, support vector machines (SVM) have been used for predicting soil water content (Pasolli et al., 2011). It was shown by Wu et al. (2007) and Ahmad et al. (2010) that SVM outperformed artificial neural networks (ANN) in estimating soil water content. Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA), both based on Bayesian discriminant theory, have been used for processing remote sensing data for agricultural targets (Lee et al., 2010) including weed detection on radish (Cho et al., 2002), sugar beet (Jafari et al., 2006), carrots (Piron et al., 2011), and also from a UAS platform (Koot, 2014). Bayesian methods, besides ANN and SVM, are also a prevalent approach for estimating soil moisture from remote sensing data (Ali et al., 2015; Notarnicola et al., 2008; Paloscia et al., 2008). There are other learning algorithms which have been used for drought monitoring such as decision trees (Im et al., 2016; Park et al., 2016). Partial least squares (PLS) and principal component analysis (PCA) or even heuristic methods are also useful to reduce the dimensionality of the dataset by eliminating redundancy and creating independent parameters for estimating a specific parameter (Mulla, 2013; Ye et al., 2017).

Previously, a UAS-deployable spectral data acquisition system was developed with a dynamic system to update integration time of spectrometers during varying ambient light conditions. The system was shown to successfully recognize targets painted with different grayscale targets (Hamidisepehr & Sama, 2018). The main objective of this study was to apply the same method for estimating soil moisture content under varying ambient light conditions. Specific objectives include:

1. Test the system on soil samples at different moisture contents under varying ambient light conditions.
2. Assess prediction accuracy on multiple models generated by different machine learning algorithms for classifying soil moisture content.

6.3 Material and Methods

6.3.1 Sample Preparation

In this study, samples with predetermined moisture level were prepared from silt loam soil. Plastic containers (950 mL) volume with airtight removable lids were used to hold samples with different moisture contents. The soil was air-dried before preparing samples with various moisture levels passed through a 2 mm sieve to avoid disturbance through larger mineral and organic particles. To determine the initial moisture gravimetrically, a sample was put in a convection oven at 105 °C for over 24 hours. Mass of the sample before and after drying was compared to find out the initial soil moisture content. It was intended to provide samples in seven moisture levels: air dry, 5%, 10%, 15%, 20%, 25%, and 30%. Each container was filled with 150 mL of soil, and the surface was kept flat and uniform. The net mass of soil was measured by subtracting of container tare from the entire weight of the sample. In the next step, the amount of water required for each sample to reach a certain moisture content was determined. Water was sprayed onto the soil surface as evenly as possible. The lid of each container was closed for several days to allow the added moisture to redistribute through the sample. If moisture stayed on the very top surface of the soil sample, the container was shaken to mix the top surface layer with a lower level. Samples then were left to equilibrate for several more days to ensure even moisture distribution through the sample. Three replications were prepared for each moisture level for a total of 21 soil samples for this experiment.

6.3.2 Spectral Data Collection

Reflected light from each sample was measured using three portable spectrometers (STS, Ocean Optics, Dunedin, FL) in the ultraviolet (UV), visible (VIS), and near-infrared (NIR) ranges. These spectrometers were coupled with a Raspberry Pi 3 (RPi) (B V1.2, Raspberry Pi Foundation, Cambridge, United Kingdom) to control the measurement process. There were two python scripts running continuously on the RPi: the first one was for communicating with a GPS receiver for updating time on the RPi, and the second script enabled the system for periodic data collection and recording each measurement as a tab-delimited text file with a time-formatted file naming schema. The serial number of the respective spectrometer was recorded in each filename to facilitate tracking individual measurements.

Data collection was implemented on May 10 and 11, 2018 between approximately 10 am and 4 pm on the roof of the Charles E. Barnhart Building in Lexington, KY. Samples at different moisture contents were measured during different ambient light conditions in the morning and in the afternoon. Each sample was placed in the measurement spot for about 20 minutes and approximately 230 measurements were made during each measurement period by each spectrometer. The length of each measurement process varied based on the spectrometer type and ambient light automatically by updating integration time, but was typically less than 10 s. There was a 10-second break after finishing each measurement from all spectrometers before the next measurement was made. The integration time of a spectrometer is an important parameter which needs to be set in a way to provide maximum performance from sensors. If integration time was set too high on a given spectrometer, the resulting spectrum would saturate at the maximum intensity. If integration time was set too low, the resulting spectrum would either saturate at the minimum intensity (dark current) or be insensitive to the parameter of interest. Changes in ambient light cause different spectral responses from a constant target. In order to compensate for the ambient light conditions, a dynamic approach for adjusting integration time near real time was implemented. In this approach, a maximum intensity with no saturation was considered as a threshold. The ratio of threshold intensity and the maximum intensity from the last measurement was used as a gain to obtain the integration time for the next measurement.

During the 20-minute measurement period, the moisture content of each soil sample decreased due to evaporation. Moisture content after the measurement period was determined gravimetrically by putting each sample in an oven for over 24 hours at 105°C. Given the change in moisture content during a measurement period, the entire measurement period was subdivided into three different periods: the first 5 minutes, the first 10 minutes, and the entire 20 minutes after starting to take measurements. Models were trained for all three intervals and compared in terms of prediction accuracy on samples which had not been shown to the learning algorithm during the training process.

6.3.3 Test Stand and Sample Holder

A test stand was deployed to locate a soil sample underneath the spectrometers. The height of the reflectance spectrometers was set at 1 m; however, it can be adjusted for further experiments. Initially, plastic sample containers were intended to use as sample holder; however, it was observed that the edges of a container would cast a shadow on soil surface during data collection. Therefore, a sample holder was designed with a hollowed rectangular pocket in the center for placing the soil sample. The sample holder was milled from a sheet of black Delrin plastic and the outer surface painted flat black to minimize specular reflections from the sample holder surface. The dimensions of pocket were selected based on the field of view of three adjacent spectrometers and the area that they cover from a 1 m height. A drawing of the two data acquisitions systems and the sample is shown in Figure 6-1. Dimensional information on different parts can be found in 172 and 173.

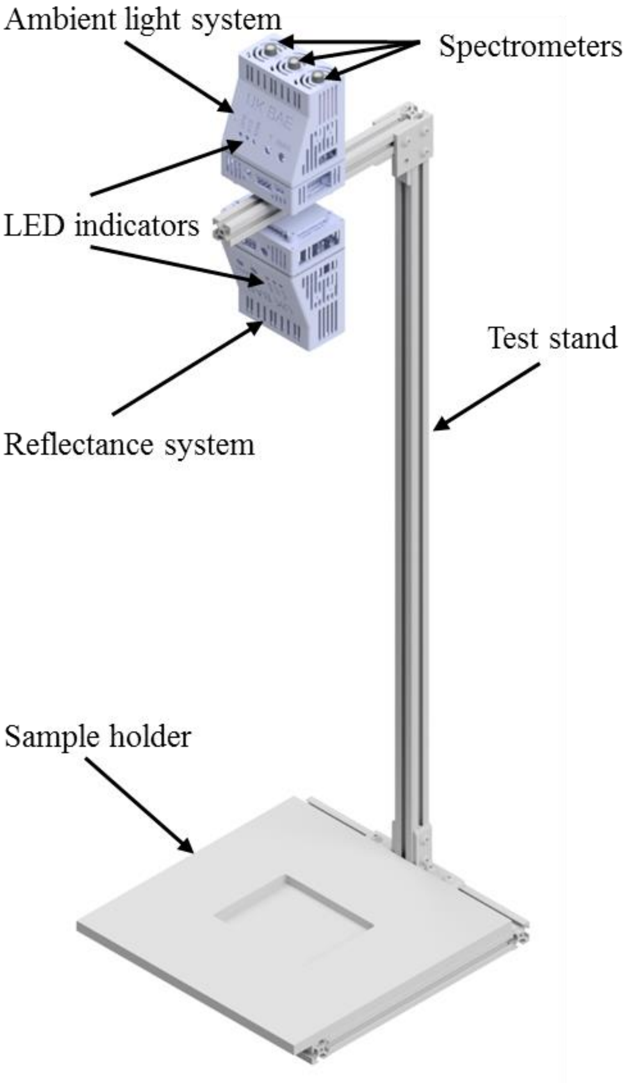


Figure 6-1: CAD drawing of data acquisition systems and the sample holder mounted on the test stand

6.3.4 *Spectral Data Preprocessing*

Raw reflectance was measured at 1024 distinct wavelengths for each type of spectrometer individually. In order to avoid issues with multi-collinearity, highly correlated wavelengths were combined to create a single independent parameter using partial least squares (PLS) regression method in MATLAB (R2017a; The Mathworks; Natick, MA). Reducing the number of input variables also reduced the dimensionality of the dataset – making subsequent model training faster. Two decision making criteria were considered to find the optimal number of wavelengths for generating a prediction model: estimated mean squared prediction error and variance explained in moisture content. The optimal number of components has both a low estimated mean squared error and a high variance explanation of the soil moisture content.

6.3.5 Ambient Light Measurements

A second data acquisition system similar to the reflectance system faced upward to keep track of ambient light changes. The only difference between two sets is that ambient light spectrometers were fitted with optical diffusers and reflectance spectrometers were fitted with collimating lenses for setting the field-of-view. Ambient light spectrometers were calibrated by the manufacturer in compliance with NIST practices. A calibration coefficient was provided for each wavelength to calibrate the ambient light data, and the intensity unit changed from count value to units of energy. Unlike reflectance spectrometers, ambient light spectrometers were set to a constant integration time defined by the manufacturer; 1 s on UV and NIR spectrometers and 180 ms on VIS spectrometer.

6.3.6 Machine Learning Methods

Cross-validation method was defined as the validation method. This method protected the model against overfitting by partitioning the dataset into five folds (or divisions). Each fold is a random selection of training/validation/testing data. For each fold, a model was trained using the out-of-fold observations, and then the model performance was assessed using in-fold-data. Prediction accuracy on each fold was estimated, and the average overall accuracy over all folds was calculated.

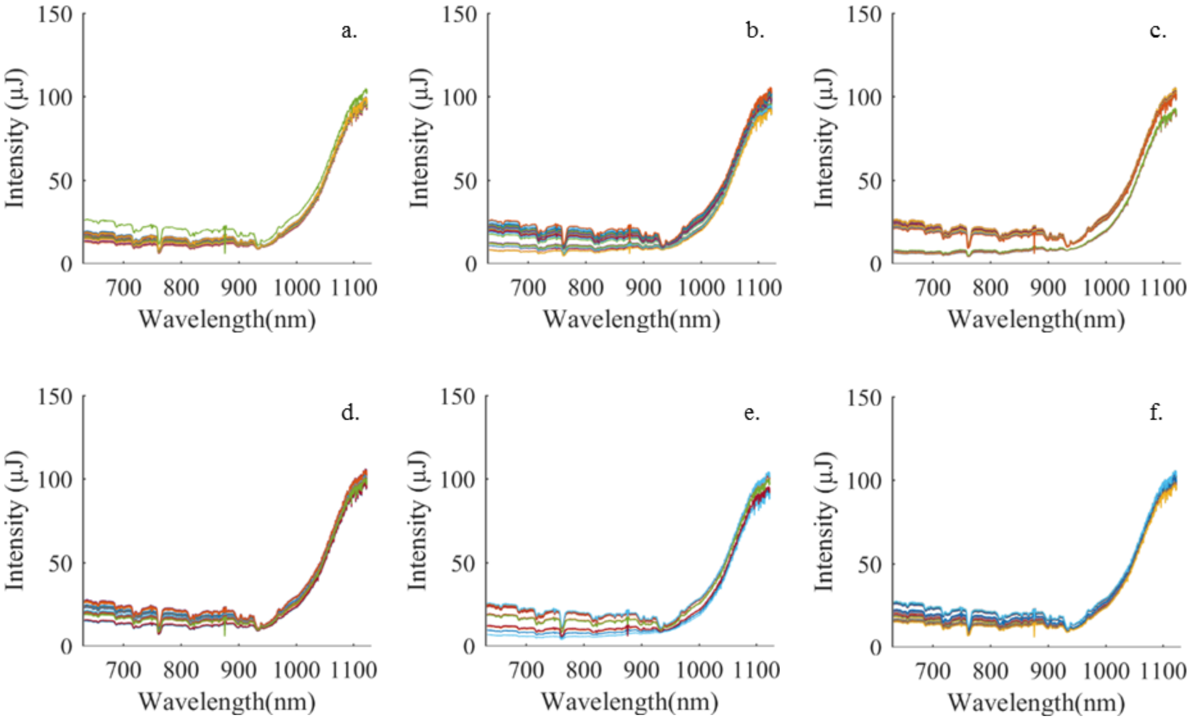
At the time of this study, there were 22 pre-developed learning algorithms available in the *Classification Learner App* in MATLAB (R2017a) including decision trees, support vector machines (SVM), nearest neighbor classifiers, ensemble classifiers, and discriminant analysis (Bayesian method). The same validation method was deployed for all learning algorithms.

At the end of the learning process, 22 models were generated for each spectrometer and for each measurement period. The prediction accuracy of these models was compared to find the optimal spectrometer for soil moisture estimation. The goal was to reduce the number of sensors to minimize the size of the data acquisition system and reduce the data processing complexity. Payload size and mass affect UAS performance. It would be ideal to have only one spectrometer deployed on the data acquisition system if it could provide a soil moisture classification model with sufficiently high prediction accuracy. The learning algorithm and the spectrometer type which provided the highest prediction accuracy compared to other models was chosen as the optimal model for estimating soil moisture.

6.4 Results and Discussion

6.4.1 Ambient light data

Ambient light data was collected alongside soil reflectance measurements to keep track of the solar irradance. This allowed the range of solar radiance to be compared across all soil samples to help ensure that solar irradance was well-distributed across all samples. Figure 6-2 shows ambient light spectrums collected using NIR spectrometer at different ambient light conditions. Each subfigure illustrates the ambient light condition when a specific soil moisture was measured.



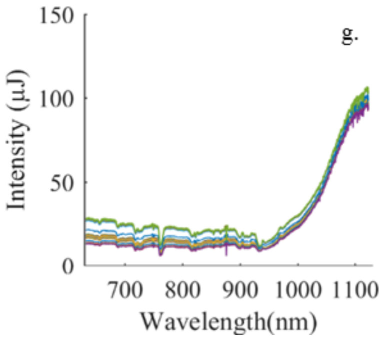


Figure 6-2: Calibrated ambient light measurements simultaneously with soil reflectance measurement with NIR spectrometer with soil moisture content of a. air dried; b. 5%; c. 10%; d. 15%; e. 20%; f. 25%; g. 30%.

6.4.2 Spectral data from soil samples

Full spectra from three types of spectrometers and for seven soil moisture levels were obtained during data collection. Each moisture level was replicated at three different times of the day to include more variability in the ambient light during data collection. Figure 6-3 shows spectra obtained with the NIR spectrometer on seven moisture levels. Spectra obtained from UV and VIS spectrometers can be found in 183. As expected, spectra from each moisture level are similar to each other with a peak intensity close to the defined threshold intensity due to the variable integration time method.

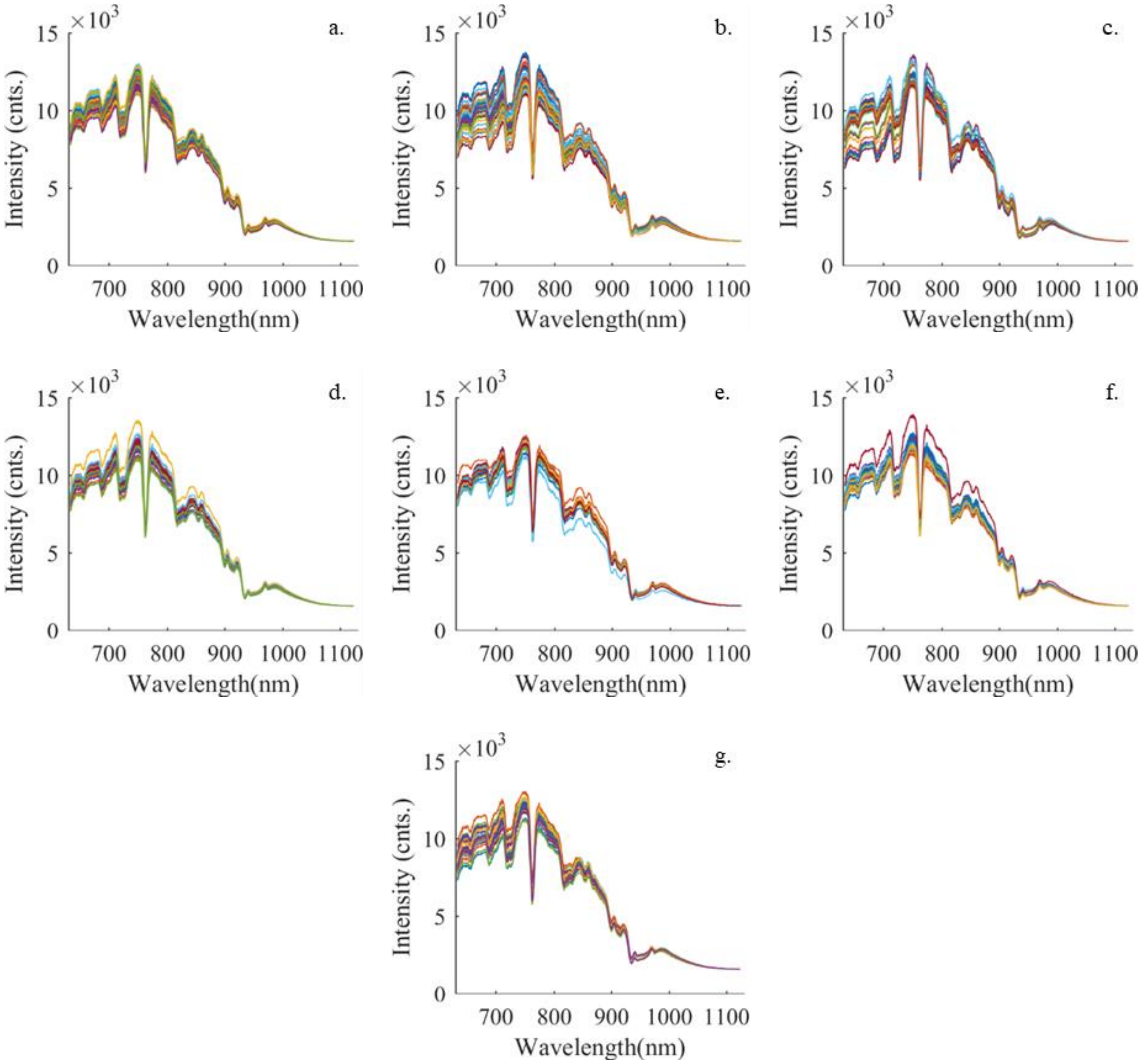


Figure 6-3: Compensated reflected light measurements with NIR spectrometer from soil samples with soil moisture content of a. air dried; b. 5%; c. 10%; d. 15%; e. 20%; f. 25%; g. 30%

6.4.3 Spectral data preprocessing

The PLS regression preprocessing method helped to reduce the dimensionality of the spectral dataset. Figure 6-4 shows the amount of estimated error and also variance explanation in soil moisture at a different number of components. Number of components ranging from seven to twenty were tested as a preliminary evaluation. Fifteen components provided models with the lowest complexity and the highest prediction accuracy; however, there were only slight differences in this range which might differ from one experiment to the other. Fifteen components meet both criteria of high variance explanation and low estimated error. The percent of variance explanation increases at a faster rate for lower number of components, but the rate is gradually decreasing for a higher number of components. On the other hand, the amount of estimated error decreases by increasing the number of components for lower range while the error did not decrease noticeably after 15 components. Hence, 15 components were determined as the optimal number of components for feeding to machine learning algorithms. The optimal number was about the same for different types of spectrometers; although, they were not shown in the figure. A few higher and lower number of components were tested as a preliminary work to make sure 15 was the optimal number components.

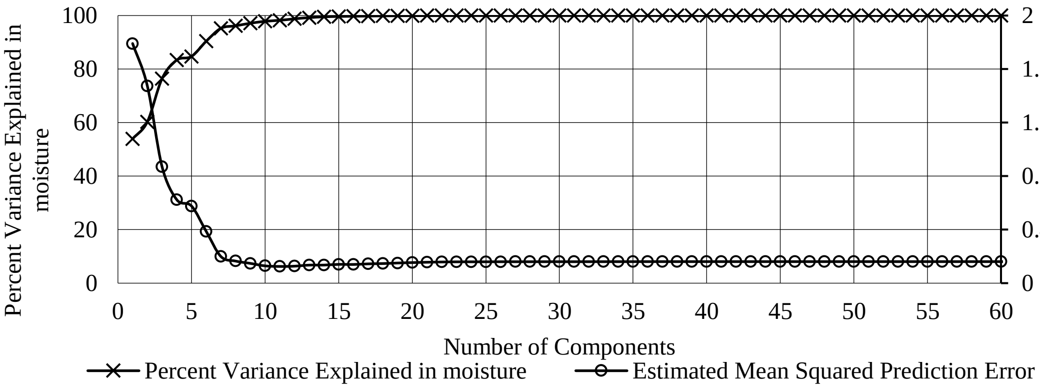


Figure 6-4: Percent variance explained in moisture, and estimated mean squared prediction error for 1 to 60 components.

6.4.4 *Machine learning method*

Twenty-two models with 15 components were trained for each type of spectrometer and three lengths of data collection periods. Seven models which had the highest accuracy on average for all three types of spectrometers are shown in Figure 6-5. Among these models, a linear discriminant analysis model provided the maximum prediction accuracy for both VIS and NIR spectrometers and the 10-minute measurement period of data collection. The same results were obtained for the first 5-minute measurement period of data collection (182).

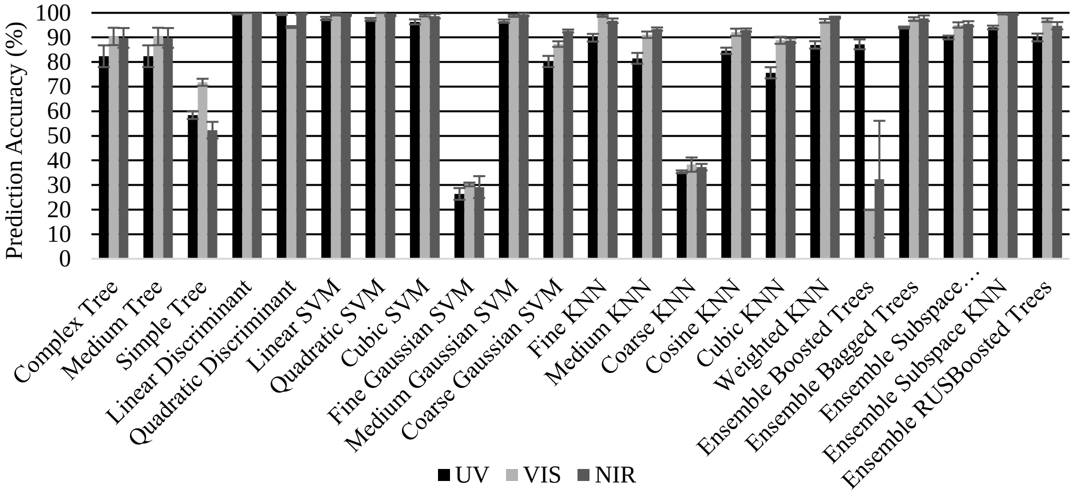


Figure 6-5: Prediction accuracy for the 22 machine learning algorithms applied to relative reflectance data with three types of spectrometers during a 10-minute measurement period.

Linear discriminant analysis algorithm provided perfect prediction accuracies for both VIS and NIR spectrometers. It should be noted that there was considerable overlap in spectral ranges of VIS and NIR spectrometers, which provides some insight as to why they exhibited similar results.

Figure 6-6 shows the prediction accuracy after a 20-minute data collection period using the same learning methods. For the 20-minute data collection, the quadratic discriminant analysis model provided a higher accuracy on average for all three spectrometers while the linear discriminant analysis model had higher performance on the VIS spectrometer data. Since the soil moisture was decreasing during the data collection measurement period, the prediction accuracy during the 20-minute measurement period was less than the 5-minute and 10-minute measurement periods. The amount of variability in soil moisture depends on initial moisture level of a given sample and the ambient air conditions. This difference was generally larger for high moisture levels due to more free water to evaporate. Also, it was observed that the top thin layer of soil, especially at higher moisture levels, was brighter than deeper layers, which lost less moisture during data collection. High overall accuracy in the 20-minute period was obtained from VIS spectrometer compared to other spectrometers. It was expected to obtain lower accuracy in UV spectrometer; however, lower accuracy for the NIR spectrometer observations was a little unexpected. Figure 6-6 demonstrated that the spectral measurement is not limited to the thin top layer and can obtain information about deeper layers. Light in shorter wavelengths contains more energy which can help the light to penetrate more and collect information from deeper layers. Wavelengths in the VIS spectrometer are shorter than the ones in the NIR spectrometer and have the capability to extract information from deeper layers of soil samples (Jackson & Huete, 1991), and it can be considered as a possible explanation on better performance of VIS over NIR spectrometer on Linear Discriminant method during the 20 minute measurement period.

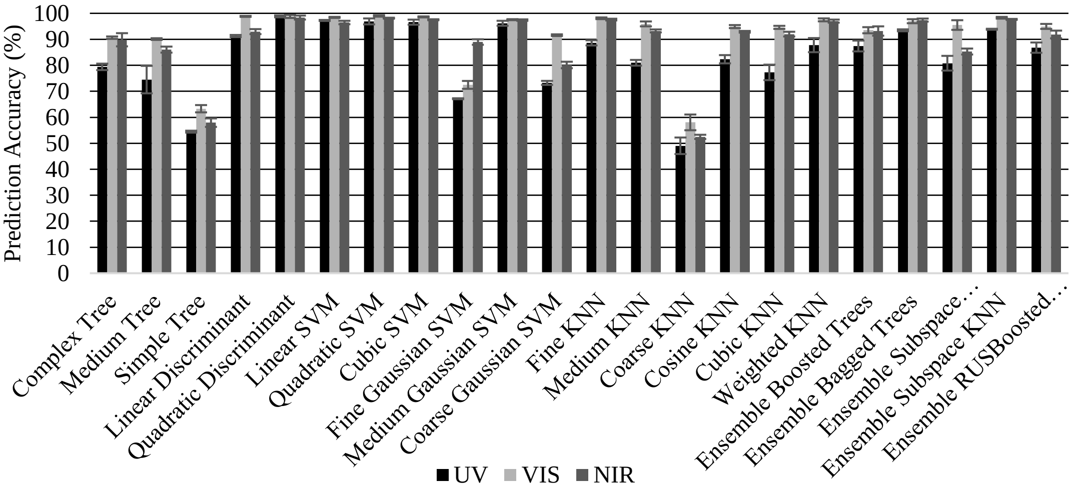
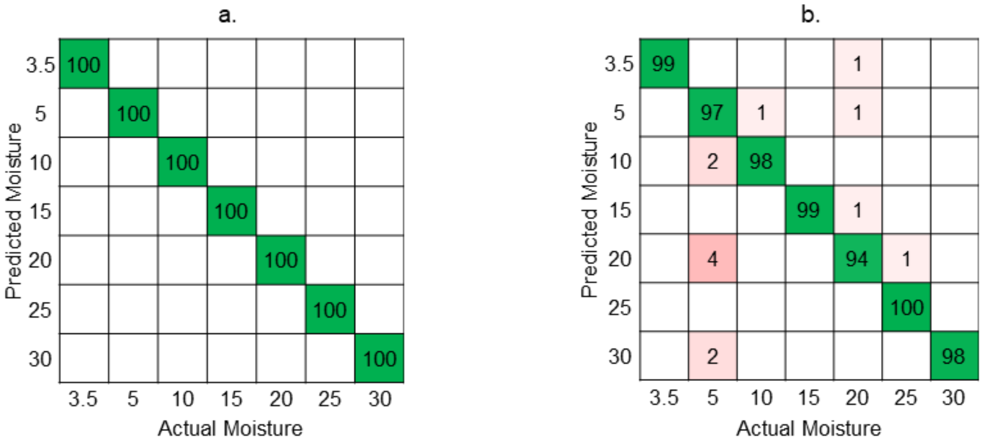


Figure 6-6: Prediction accuracy for the 22 machine learning algorithms applied to relative reflectance data on three types of spectrometers during a 20-minute measurement period.

The confusion matrix is an illustrative method to show the scattering of inaccuracies in soil moisture estimation. Two confusion matrixes are shown in Figure 6-7 on data collected by NIR spectrometer. One came from the model generated with the data collection in the 10-minute measurement period and using linear discriminant analysis (Figure 6-7-a). The other one was from the model with data collection in the 20-minute measurement period using quadratic discriminant analysis (Figure 6-7-b). The 10-minute period of data collection generated a matrix with every sample properly classified. On the other hand, for the 20-minute period, several inaccuracies were observed with no discernable pattern. However, the overall prediction accuracy was still over 98%.



a.

	3.5	5	10	15	20	25	30
3.5	100						
5		100					
10			100				
15				100			
20					100		
25						100	
30							100

b.

	3.5	5	10	15	20	25	30
3.5	99				1		
5		97	1		1		
10		2	98				
15				99	1		
20		4			94	1	
25						100	
30		2					98

Figure 6-7: Soil moisture classification results for models generated from data collected with NIR spectrometer from a) the 10-minute data collection and using linear discriminant analysis and b) the 20-minute data collection with quadratic discriminant analysis.

6.5 Conclusions

Previously, a UAS-deployable spectral data acquisition system was developed with a dynamic system to update the integration time of spectrometers during varying ambient light conditions. The system was shown to successfully recognize varying grayscale targets. In this study, this system was deployed to classify soil moisture content using spectral data. The capability of the system for differentiating between different moisture levels under varying ambient light conditions was tested. A broad spectrum was obtained from each measurement with each spectrometer. The dataset was compressed using PLS regression to eliminate redundant information (correlated wavelengths). After preprocessing, 22 models were trained using data with three different spectrometers to find out about the optimal model and sensor which provided the highest prediction accuracy.

Three periods of data collection were considered post sample deployment (5, 10, and 20-minute) to observe the effect of soil moisture content change during the data collection period. The linear discriminant model combined with the VIS spectrometer and a period up to 10 minutes resulted in 100% classification accuracy on seven soil moisture content levels ranging from <5% to 30% moisture content.

CHAPTER 7: SUMMARY AND CONCLUSIONS

The primary objective of this dissertation was to develop tools and methods for remotely estimating soil water content. In the first study, moisture-controlled soil and wheat stalk residue samples were prepared and measured at varying heights using a reflectance probe connected to visible and near-infrared spectrometers. A computer program was written that used reflectance data to determine the optimal narrowband wavelengths based on user-defined constraints, and the statistical significance of sensor height and moisture content was determined for the “best” pair. Constraints for this study were configured to maximize the slope of the index (i.e., sensitivity to moisture) and either to maximize the R2 or minimize the RMSE of the index function. Results showed that wavelengths centered near 1300 nm and 1500 nm, within the range of 400 to 1700 nm, produced the best index for individual samples. An advantage of this pair of wavelengths is that they can be sensed with a single type of sensor using narrowband optical filters. The 1500 nm band, when measured with an active ground-based sensor, will provide spectral information not available when using passive aerial or satellite-based remote sensing methods due to absorption from atmospheric moisture. When applied to all samples, the index performed well for the soil samples but poorly for the wheat stalk residue samples. Based on these results, it is expected that crop residues, such as wheat stalks, will reduce the accuracy of remotely sensed soil surface moisture measurements. Future work should include heterogeneous surfaces that include both soil and crop residue in varying proportions to determine the composite response.

In the second study, the same spectral data from the first study was used but with a different type of data processing. Relative reflectance spectral data from moisture-controlled silt-loam soil and wheat stalk residue samples was used to test the ability of several machine learning algorithms to classify moisture content from the spectral data. This method contrasted with an index-based method used in a previous study of the same spectral data. Previous work has tended to focus on a pair of wavelengths rather than the full spectrum. It was hypothesized that the machine learning approach would yield better prediction accuracy because of utilizing a larger number of components than index-based method from the spectral data. The appropriate number of components for this dataset was determined to be 20 using PLS regression. The components were fed into 20 different machine learning algorithms, from which cubic SVM and ensemble bagged trees produced the highest combined prediction accuracy for silt-loam soil samples (over 93%) and wheat stalk residue samples (over 86%). This represented a substantial improvement over the index-based method, where only two wavelengths were used to develop a moisture prediction model.

Third study was aimed to develop a spectral sensing platform suitable for UAS deployment and to measure the reflectance from a reference target to assist with the development of a calibration procedure that is functional over a wide range of ambient light conditions. This study included two parts. In the first part of this study, a platform was developed to be deployed on a UAS to measure the reflectance intensity from a target. Two sets of portable STS spectrometers in three ranges of UV, VIS, and NIR were used along with a RPi to form a reflectance system and an ambient light system. In the second part of this study, a method for compensating for ambient light conditions and sensor integration time was developed and tested during the 2017 Great American Eclipse. Results showed a large variability in reflected light intensity due to significant changes in sun radiance. Reflectance values were compensated using ambient light measurements and integration time. Compensated reflectance values exhibited a consistent spectral signature for measurements taken at different ambient light conditions. This method will be useful for future field work where ambient light conditions cannot be controlled and the sensor integration time may need to be adjusted to optimize the sensitivity of the spectrometer.

In the fourth study, the objective goal was to test the data acquisition system from the last study to recognize reflectance targets under ambient light conditions. Six targets were painted with different colors in the grayscale range. Then, these targets were benchmarked using visible and near-infrared spectrometers calibrated with a reference target. Two sets of spectrometers coupled with an embedded data acquisition system were used for collecting spectral data at five days at different ambient light conditions. A set was designated for measuring ambient light, and sun radiance and the other set measured the reflectance from targets. A mechanism was applied to update the integration time of each reflectance spectrometer based on ambient light condition. Then, spectral data were calibrated against ambient light condition and integration time. Twenty-two learning algorithms were used to generate models for recognizing reflectance targets from each other. It was determined that most of the algorithms had a prediction accuracy over 90%. The quadratic discriminant model provided the perfect prediction accuracy. It was concluded that by calibrating the system against ambient light conditions, machine learning models could provide a sufficiently high accuracy for recognizing targets from their spectral responses.

In the last study, the system developed in Chapter 4 and evaluated in Chapter 5 using simple targets was tested for classifying moisture contents of soil samples. Integration time on individual spectrometers was adjusted according to the ambient light condition. By compensating against ambient light changes, reflectance measurements focus on other sample characteristics including soil moisture content. A preprocessing step using PLS regression was implemented on spectral data to reduce the dimensionality of the dataset and avoid overfitting. In the preprocessing step, the number of variables from 1024 wavelengths reduced to 15 independent parameters. Linear discriminant analysis method on data collected from NIR and VIS spectrometers, among 22 learning algorithms, generated a model with a perfect prediction accuracy based on a 10-minute data collection. For longer periods of data collection, several more misclassifications occurred in all models due to changes in soil moisture content during data collection. For a 20-minute data collection, quadratic discriminant analysis resulted in a model with the highest performance (98%) from data collected by NIR spectrometer. It can be concluded that the hyperspectral data acquisition system introduced in this study along with the linear discriminant analysis for processing can be used for soil moisture remote sensing.

CHAPTER 8: FUTURE WORK

Future work based on results obtained in the first study should include heterogeneous samples that include both soil and crop residue in varying proportions to determine the composite response. As new low-cost sensors are developed, the optimization parameters used to determine the "best" wavelengths should be refined based on actual sensor response, rather than ideal assumptions. Also, as a future work for both first and second study to test have similar experiments in field conditions. The results of these two studies were from laboratory prepared samples of individual material types measured under controlled conditions. Field application of this work will require additional considerations including, among other factors, more complex distributions of materials and variability in ambient light. Future work should include testing of this process under field conditions to demonstrate the applicability as a high-throughput method for remotely sensing moisture content of soils and crop residues.

Future work based on results obtained in the third study should include testing the ability to classify different targets at varying ambient light conditions and to automatically adjust the integration time of each reflectance spectrometer based on previous measurements to maximize sensitivity which was conducted in the fourth study. Similarly, testing the same system and method for classifying soil samples was considered as a future work for third and fourth studies which have been successfully accomplished in the fifth study.

The next step to continue this path would be to measure the soil or crops spectrally from a UAS platform and estimate soil moisture based on the reflectance measurements.

Different field parameter estimations should be tested for other common parameters which can be classified based on spectral measurements.

APPENDICES

A. CODES

A.1. Optimal Index Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Optimal_index.m %
% Author: Ali Hamidisepehr (c) 2016 %
% Function: This script reads in a reflectance data file and %
% calculates the performance of all possible indexes. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clc;close all;clear;
num_samples=21; % Number of samples in total excluding reference and
background data
num_same_sample=3;% We might have samples with almost the same moisture
content. We specify the number of them here
num_moisture_levels=7;% number of different moisture levels
num_replication=3; % Number of replications for each sample. (Different
from samples with almost the same moisture)
[x x_txt raw]=xlsread('Dataset_Overall_stalks.xlsx');%Reading the
dataset including all measurements with full specturm
same_HM=zeros(length(x),1);b=zeros(length(x),1);r=zeros(length(x),1);%
initializing a matrix for putting samples with same moistureand height
together
wavelength=x(:,1);% the first column of the excel file produced
includes wavelength values
VIS=0;NIR=0;% initialize two values for vision and NIR range
bandwidth=25;% bandwith or the width of wavelength range centered at
above values
% The codes or names correspond to different moisture. Notice that when
we
% have more than one sample with the same moisture we must put them
close
% together. Say the first three arrays in the matrix represent samples
with
% the same moisture
moist={'S1_';
'S2_';'S3';'S4';'S5';'S6';'S7';'S8';'S9';'S10';'S11';'S12';'S13';'S14';
'S15';'S16';'S17';'S18';'S19';'S20';'S21'};
height=['H1';'H2';'H3'; 'H4'; 'H5']; % Defining different heights
moist2=[1:num_moisture_levels*num_same_sample];% this matrix is
technically same as moist matrix. the only difference is this one
includes numerical values
Moist_values=[0 5 10 15 20 25 30];% moisture levels used in soil
samples
average_all=zeros(length(wavelength),1);%this variables is for putting
averages of all moisture levels togther
```

```
%In this script we plot our data for each given height and all moisture
%levels. The main loops changes height level
for  h=1:length(height)
    figure % for each height one figure plots all moisture levels
    hold on
%This loops take into account only samples with different moisture so
that
%we combine samples with the same moisture
for  m=1:num_same_sample:length(moist)

% This loop goes through all columns one by one to detect columns with
the
% same height and moisture
for v=1:length(x_txt)
    k=m; % Variable k is used to take all the same moisture samples
into account through using a while loop
    while k<num_same_sample+m
    t=strfind(x_txt(v),(moist{k}));%For each column, check if there is
the desired moisture in the label of the column
    p=strfind(x_txt(v),height(h,:));%For each column, check if there is
the desired height in the label of the column
    if ~isempty(t{1}) && ~isempty(p{1})% If the column is the deired
one with desired moisture and height
       same_HM=[same_HM x(:,v)]; % Put all the columns with same
moisture and height next to the each other

    end
    k=k+1;
    end
end
same_HM(:,1)=[];% Since we intialized with a column of zeros, we remove
that column from the last result
average=mean(same_HM,2);% Take an average between all samples and
replications with the same moisture and height
average_all=[average_all mean(same_HM,2)];%combining heights and
putting "average" values next to each other


plot(x(:,1),average,'DisplayName',['Sample '
num2str(moist2(m:m+num_same_sample-1))])% plot the average data vs
wavelength
xlabel('Wavelength(nm)');ylabel('%Reflectance');
same_HM=zeros(length(x),1);%Reset the matrix same_HM to the initialized
zeros again at the end of each iteration
end
for m=length(moist):length(moist)+2 % for data related to background
and refrence we repeat what we did before in the loops
    for v=1:length(x_txt)

    t=strfind(x_txt(v),'BACK');% For each column, check if there is
Background in the label of the column
```

```
    g=strfind(x_txt(v),'REF');% For each column, check if there is
Reference in the label of the column
    p=strfind(x_txt(v),height(h,:));%For each column, check if there is
the desired height in the label of the column
    if ~isempty(t{1}) && ~isempty(p{1})% If the column is the deired
one(the result of strfind is a cell so that we should use its numerical
value by using {})
       b=[b x(:,v)];% Put all the columns with 'Background' and the
same height next to the each other
    elseif ~isempty(g{1}) && ~isempty(p{1})
                r=[r x(:,v)]; % Put all the columns with 'Reference'
and the same height next to the each other
       end

    end
end
b(:,1)=[];% Since we intialized with a column of zeros, we remove that
column from the last result
background=mean(b,2);% Take an average between all samples with
'Background' and the same height
r(:,1)=[];% Since we intialized with a column of zeros, we remove that
column from the last result
reference=mean(r,2);% Take an average between all samples with
'Reference' and the same height
plot(x(:,1),background,'DisplayName','Background ')
plot(x(:,1),reference,'DisplayName','Reference ')
hold off
legend('-DynamicLegend','Location','Best')
title(['Reflectance versus Wavelength for Varying Soil Moisture with
Height ',num2str(h)])
b=zeros(length(x),1);r=zeros(length(x),1); %Reset matrixes 'b' and 'r'
to the initialized zeros again
end
C=1;
VIS(1)=[];NIR(1)=[];
average_all(:,1)=[];%we initialized this with zero and now it should be
eliminated
%By this for loop we go through all pairs of wavelengths one by one to
%check which pair provides the best prediction for moisture content
based
%on measured Reflectance using Rsquare, RMSE and slope of the linear
%regression of moisture and index value
for i=23:bandwidth:length(wavelength)
    for j=1:bandwidth:length(wavelength)

        if j>i % the second wavelength must be greater the first one to
avoid redundancy
        VIS=0;NIR=0;
        %set the values for wavelength 1 & 2
        wave_range2=wavelength(j);
        wave_range1=wavelength(i);
        for zz=1:length(Moist_values)*length(height)
        Range_VIS=average_all((wavelength>=(wave_range1-bandwidth) &
wavelength<=(wave_range1+bandwidth)),zz);% select the desired range of
first wavelength and Extract average reflectance values associated with
them in desired column
        VIS=[VIS mean(Range_VIS)];%put average values of the first
range of wavelength next to each other so that for each given moisture
and height
        Range_NIR=average_all((wavelength>=(wave_range2-bandwidth) &
wavelength<=(wave_range2+bandwidth)),zz);% select the desired range of
second wavelength and Extract average reflectance values associated
with them in desired column
        NIR=[NIR mean(Range_NIR)];%put average values of the second
range of wavelength next to each other so that for each given moisture
and height
        end
        %we initialized these with zero and now it should be eliminated
        VIS(1)=[];
        NIR(1)=[];
        %in this loop we seperate differnt heights for Vision and NIR
        %ranges and put them in rows
        for gg=1:length(height)
        VIS_Height(gg,:)=VIS(((gg-
1)*num_moisture_levels)+1:gg*num_moisture_levels);
        NIR_Height(gg,:)=NIR(((gg-
1)*num_moisture_levels)+1:gg*num_moisture_levels);
        NDWI(gg,:)=(VIS_Height(gg,:)-
NIR_Height(gg,:))./(VIS_Height(gg,:)+NIR_Height(gg,:));%caluculate the
Index elment_wisely
        end
%based on statistical analysis Height doesn't affect on the results
%significantly then we could combine heights
one_height_NDWI=mean(NDWI);
%fit a model so that dependant axis is moist values and independant
axis is
%NDWI
mdl=fitlm(Moist_values,one_height_NDWI,'linear','RobustOpts','on');
rsquare=mdl.Rsquared.Ordinary;%obtain the Rsqure value for each index
rmse=mdl.RMSE;%obtain the RMSE value for each index
coefficients=mdl.Coefficients.Estimate;
slope=coefficients(2);%%obtain the slope for each index

%creating a row of matrix for each index including two wavelengths and
%three decision criteria
index_all(C,:)=[wavelength(i) wavelength(j) rsquare rmse slope];
C=C+1;

    end
    end
end
figure
scatter(index_all(:,3),index_all(:,5))
xlabel('R_Squared');ylabel('Slope');
```

```
figure
scatter(index_all(:,4),index_all(:,5))
xlabel('RMSE');ylabel('Slope');
```

A.2. Creating a Comprehensive Data File from All Text Files

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Dataset_Overall_stalks.m %
% Author: Ali Hamidisepehr (c) 2016 %
% Function: This script reads all reflectance data text files and %
% put them in one single Excel file. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Dataset_Overall_stalks()
clear;clc;close all;
folder = pwd;% get the directory
Listing = struct2cell(dir(folder)); % list all files in the current
directory
FileNames = Listing(1,:)';% put all the file names in a column
h=zeros(2505,1);% intializing a variable that shows the number of
wavelengths
label='Wavelength';% intializing a variable that is going to used later
as label in an excel file
data_lines=16;% while extracting text file into a table, actual data
will start from line 16 for data obtained with the spectrometer
% this for loop goes through all files in the current directory to
check if
% each one is reflection data of soil samples or not
for i=1:length(Listing)
    z=[0];
    k = strfind(char(FileNames(i)),'Reflection'); %distinguish files
which contain reflection data and ignore irrelevant data
    if isempty(k)
        %do nothing
    else
        label=[label, (FileNames(i))];%put filenames including
reflection data next to eachother to be used as heading in the excel
file
        [Data from]=Extract_txt([folder,'\',char(FileNames(i))]);
%Extract wavelength and Reflection data from text files

        %this for loop goes through all reflectnce data and put all
values for a specfic files into one column
        for b=data_lines:length(from)
        z=[z;str2num(from{b})];
       end
       z(1)=[];
       h=[h,z];% this matrix creates a big matrix which includes all
data in all text files at the end of the loop
     end

end
wavelength=[0];
%this for loop creates a single column for wavelength values
for b=data_lines:length(from)
```

```
 wavelength=[wavelength;str2num(Data{b})];
 end
wavelength(1)=[];
h(:,1)=wavelength;%add wavelength values as the first column in the
table
delete('Dataset_Overall_stalks.xlsx');
xlswrite('Dataset_Overall_stalks.xlsx',label)%writing the heading part
of the excel file
xlswrite('Dataset_Overall_stalks.xlsx',h,1,'A2')%writing all the data
of the excel file
end
```

A.3. Finding Optimal Number of Components for a Prediction Model from Lab Data

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Optimal_num_components.m % Author: Ali Hamidisepehr (c) 2016 %
% Function: This script calculates the estimated error and variance
% explained for a model with a given number of components. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc; close all;
X=xlsread('mldata_Stalks_Residue_Dataset.xlsx');%reflectance values for
different samples
y=xlsread('mloutputStalks_Residue_Dataset.xlsx');% moisture values
conrresponded to each sample
num_components=20;%the desired number of components after compression
%% Calculating Estimated Mean Squared Error
% Applying PLS method for reducing the dimensionality of dataset
[Xl,Yl,Xs,Ys,beta,pctVar,PLSmsep] =
plsregress(X,y,num_components,'CV',num_components);
plot(0:num_components,PLSmsep(2,:),'b-o');
xlabel('Number of components');
ylabel('Estimated Mean Squared Prediction Error');
variables_output_StalkResidue_ready_ml=[Xs,y];% formatting dataset for
feeding into machine learning algorithms

%% Calculating Percent Variance Explained
[Xloadings,Yloadings,Xscores,Yscores,betaPLS10,PLSPctVar] =
plsregress(X,y,num_components);
plot(1:num_components,cumsum(100*PLSPctVar(2,:)),'-bo');
xlabel('Number of PLS components');
ylabel('Percent Variance Explained in Y');
ylim([90 100]);
[Xl,Yl,Xs,Ys,beta,pctVar,mse,stats] = plsregress(X,y,num_components);
plot(1:length(stats.W),stats.W,'-');
xlabel('Variable');
ylabel('PLS Weight');
legend({'1st Component' '2nd Component' '3rd
Component'},'location','NW');
```

A.4. RPi Program

A.4.1. Reading GPS Packets and Updating RPi Clock

```
##############################################################
# Title: RPi_GPS.py %
# Author: Ali Hamidisepehr (c) 2017 %
# Function: This script receives time data continuously through
# GPS receiver and set the RPi clock every few seconds.
##############################################################

import gps
import os
import time
import RPi.GPIO as gpio
from itertools import cycle
gpio.setmode(gpio.BCM)
gpio.setwarnings(False)
gpio.setup(21,gpio.OUT)
# Listen on port 2947 (gpsd) of localhost
session = gps.gps("localhost", "2947")
session.stream(gps.WATCH_ENABLE | gps.WATCH_NEWSTYLE)
led_flip=cycle(range(2))
i = 0
while True:
    try:
        report = session.next()
        # Wait for a 'TPV' report and display the current time
        # To see all report data, uncomment the line below
        # print report
        if report['class'] == 'TPV':

            if hasattr(report, 'time'):
                timestamp = report.time
                print(timestamp)
                if led_flip.next():
                        gpio.output(21,False)
                else:
                        gpio.output(21,True)



                if i == 0:
                    GPStime = timestamp.split("T")[1].split(".")[0]
                    GPSyear = timestamp.split("-")[0]
                    GPSmonth = timestamp.split("-")[1]
                    GPSdate = timestamp.split("-")[2].split("T")[0]
                    if GPSmonth == "01":
                        GPSmonthtext = "Jan"
                    elif GPSmonth == "02":
                        GPSmonthtext = "Feb"
                    elif GPSmonth == "03":
                        GPSmonthtext = "Mar"
                    elif GPSmonth == "04":
                        GPSmonthtext = "Apr"
                    elif GPSmonth == "05":
                        GPSmonthtext = "May"
                    elif GPSmonth == "06":
                        GPSmonthtext = "Jun"
                    elif GPSmonth == "07":
                        GPSmonthtext = "Jul"
                    elif GPSmonth == "08":
                        GPSmonthtext = "Aug"
                    elif GPSmonth == "09":
                        GPSmonthtext = "Sep"
                    elif GPSmonth == "10":
                        GPSmonthtext = "Oct"
                    elif GPSmonth == "11":
                        GPSmonthtext = "Nov"
                    else:
                        GPSmonthtext = "Dec"
                    DateString = "sudo date -s '" + GPSmonthtext + " "+
GPSdate + " " + GPStime + " UTC " + GPSyear + "'"
                    print(DateString)
                    os.system(DateString)
                i = i + 1
                if i > 5:
                    i = 0
    except KeyError:
        pass
    except KeyboardInterrupt:
        quit()
    except StopIteration:
        session = None
        print("GPSD has terminated")
```

A.4.2. *Main Program*

```
#############################################################
# Title: RPi_main_script.py %
# Author: Ali Hamidisepehr (c) 2017 %
# Function: This script controls individual spectrometers and
# adjust integration time based on ambient light automatically.
#############################################################


import urllib.request #importing the library for opening the php files
corresponded to spectrometers settings
import time #importing the library for setting the time parameters like
delay
import re #importing the regular expression operations library for
finding a pattern in a string
import glob,os #Miscellaneous operating system interfaces library for
shutting down the system. glob will be used for fixing the file format.
from time import gmtime,strftime #importing time variables for reading
the time and date
import RPi.GPIO as gpio #importing GPIO library for using GPIO pins on
RPi
import numpy as np
gpio.setmode(gpio.BCM) # Define numbering mode on GPIO
gpio.setwarnings(False) # Disable warnings
gpio.setup(4,gpio.IN,gpio.PUD_UP)# Defining pin 4 as an input and
pulled up switch (internally connected to 3.3V). PWM signal which was
already converted to a digital output will be used on this pin
gpio.setup(12,gpio.IN,gpio.PUD_UP)# Defining pin 12 as an input and
pulled up switch (internally connected to 3.3V). It will be used as a
push button on ground and UAS set
gpio.setup(25,gpio.IN,gpio.PUD_UP) #Defining pin 25 as a switch to
differentiate between radiance(ground) set and reflectance(UAS) set.
gpio.setup(23,gpio.OUT) # using pin 23 as an output in LED to indicate
if the spectrometers are taking measurements
gpio.setup(16,gpio.OUT) # using pin 16 as an output in LED to indicate
if the switch is set on for radiance or reflectance measurement
gpio.output(16,False)
gpio.output(23,False)
gpio.add_event_detect(12,gpio.FALLING) # Checks on pin 12 and detect
any falling signal received there. It checks if the momentary push
button connected to this pin is pressed
gpio.add_event_detect(4,gpio.FALLING) # Checks on pin 4 and detect any
falling signal received there. It checks if the converted signal from
PWM to Digital is High.
#gpio.add_event_detect(25,gpio.FALLING)
LocalAddress='http://127.0.0.1/cgi-bin/'

###################################################################
with urllib.request.urlopen('http://127.0.0.1/cgi-
bin/getserial.php?channel=0') as response111:#opening the php file
```

```
corresponded to getting the serial number of the spectrometer on
channel 0
            html111 = response111.read() #reading information obtained
from the opening the previous php file
            z=re.search('S.....',str(html111)) #search for a serial
number format from the string created from the readings in the last
command
            serialnum=z.group() #storing the serial number of the
spectrometer on channel 0

#### the same process for finding serial number on channels 1 & 2
with urllib.request.urlopen('http://127.0.0.1/cgi-
bin/getserial.php?channel=1') as response222:
            html222 = response222.read()
            z2=re.search('S.....',str(html222))
            serialnum2=z2.group()

with urllib.request.urlopen('http://127.0.0.1/cgi-
bin/getserial.php?channel=2') as response333:
            html333 = response333.read()
            z3=re.search('S.....',str(html333))
            serialnum3=z3.group()
###################################################################
DataFolder='/home/ocean/Data'
## set the location for storing spectral data on three channels
with
urllib.request.urlopen(LocalAddress+'setsavelocation.php?location='+Dat
aFolder+'&channel=0') as response6:
            html6 = response6.read()

with
urllib.request.urlopen(LocalAddress+'setsavelocation.php?location='+Dat
aFolder+'&channel=1') as response16:
            html16 = response16.read()

with
urllib.request.urlopen(LocalAddress+'setsavelocation.php?location='+Dat
aFolder+'&channel=2') as response22:
            html22 = response22.read()


## Evaluating integration time, Boxcar, and the number of scans to be
averaged on each measurement for three channels
if serialnum=='S04413': #Ground UV for sun radiance
    int_time=1000000
    boxcar=5
    average=5
elif serialnum=='S08285': #Ground VIS for sun radiance#It was
recommended by manufacturer to be 350000
    int_time=180000
    boxcar=5
    average=10
elif serialnum=='S07821': #Ground NIR for sun radiance
```

```
    int_time=1000000
    boxcar=5
    average=5
elif serialnum=='S07846': #UAV NIR for reflectance
    int_time=80000
    boxcar=5
    average=5
elif serialnum=='S05821': #UAV VIS for reflectance
    int_time=35000
    boxcar=5
    average=5
elif serialnum=='S03927': #UAV UV for reflectance
    int_time=80000
    boxcar=5
    average=5
sequenceinterval=int((int_time*average/1000)+50)
#print(serialnum);print(sequenceinterval,int_time,average)
if serialnum2=='S04413': #Ground UV for sun radiance
    int_time2=1000000
    boxcar2=5
    average2=5
elif serialnum2=='S08285': #Ground VIS for sun radiance
    int_time2=180000
    boxcar2=5
    average2=10
elif serialnum2=='S07821': #Ground NIR for sun radiance
    int_time2=1000000
    boxcar2=5
    average2=5
elif serialnum2=='S07846': #UAV NIR for reflectance
    int_time2=80000
    boxcar2=5
    average2=5
elif serialnum2=='S05821': #UAV VIS for reflectance
    int_time2=35000
    boxcar2=5
    average2=5
elif serialnum2=='S03927': #UAV UV for reflectance
    int_time2=80000
    boxcar2=5
    average2=5
sequenceinterval2=int((int_time2*average2/1000)+50)
#print(serialnum2)
#print(serialnum2);print(sequenceinterval2,int_time2,average2)
if serialnum3=='S04413': #Ground UV for sun radiance
    int_time3=1000000
    boxcar3=5
    average3=5
elif serialnum3=='S08285': #Ground VIS for sun radiance
    int_time3=180000
    boxcar3=5
    average3=10
elif serialnum3=='S07821': #Ground NIR for sun radiance
```

```
    int_time3=1000000
    boxcar3=5
    average3=4
elif serialnum3=='S07846': #UAV NIR for reflectance
    int_time3=80000
    boxcar3=5
    average3=5
elif serialnum3=='S05821': #UAV VIS for reflectance
    int_time3=35000
    boxcar3=5
    average3=5
elif serialnum3=='S03927': #UAV UV for reflectance
    int_time3=80000
    boxcar3=5
    average3=5
sequenceinterval3=int((int_time3*average3/1000)+50)
num_replic=3
if int_time>100000:
    coeff=1.1
else:
    coeff=3
delay=int((max(sequenceinterval,sequenceinterval2,sequenceinterval3)*(n
um_replic)*coeff/1000)) #amount of delay in seconds based on the
longest measurement
x2=12000

## Adjustments on each spectrometer corresponded to each specific
channel
    ## Channel 0
with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time)+'&channel=0') as response: # setting integration time on channel0
        html = response.read()
        print(html)
with
urllib.request.urlopen(LocalAddress+'setaverage.php?scans='+str(average
)+'&channel=0') as response2: #setting number of scans for each
measurement
        html2 = response2.read()

with
urllib.request.urlopen(LocalAddress+'setboxcar.php?width='+str(boxcar)+
'&channel=0') as response3: #setting boxcar
        html3 = response3.read()

with
urllib.request.urlopen(LocalAddress+'setmaxacquisitions.php?acquisition
s='+str(num_replic)+'&channel=0') as response5: #setting the maximum
number of acquistion in case taking a series of measurements
        html5 = response5.read()
```

```
with
urllib.request.urlopen(LocalAddress+'setsavemode.php?savemode=multi&cha
nnel=0') as response7: # if spectral data is stored in one file or
multiple files
        html7 = response7.read()

with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval)+'&channel=0') as response8: #the interval
between each measurement in case taking a series of measurements
        html8 = response8.read()


    ## Channel 1

with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time2)+'&channel=1') as response10:
        html10 = response10.read()

with
urllib.request.urlopen(LocalAddress+'setaverage.php?scans='+str(average
2)+'&channel=1') as response11:
        html11 = response11.read()

with
urllib.request.urlopen(LocalAddress+'setboxcar.php?width='+str(boxcar2)
+'&channel=1') as response12:
        html12 = response12.read()

with
urllib.request.urlopen(LocalAddress+'setmaxacquisitions.php?acquisition
s='+str(num_replic)+'&channel=1') as response13:
        html13 = response13.read()


with
urllib.request.urlopen(LocalAddress+'setsavemode.php?savemode=multi&cha
nnel=1') as response14:
        html14 = response14.read()

with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval2)+'&channel=1') as response15:
        html15 = response15.read()
        print(html15)


    ## Channel 2
with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time3)+'&channel=2') as response16:
        html16 = response16.read()
```

```
with
urllib.request.urlopen(LocalAddress+'setaverage.php?scans='+str(average
3)+'&channel=2') as response17:
        html17 = response17.read()

with
urllib.request.urlopen(LocalAddress+'setboxcar.php?width='+str(boxcar3)
+'&channel=2') as response18:
        html18 = response18.read()

with
urllib.request.urlopen(LocalAddress+'setmaxacquisitions.php?acquisition
s='+str(num_replic)+'&channel=2') as response19:
        html19 = response19.read()


with
urllib.request.urlopen(LocalAddress+'setsavemode.php?savemode=multi&cha
nnel=2') as response20:
        html20 = response20.read()

with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval3)+'&channel=2') as response21:
        html21 = response21.read()



## The main While loop checks every few seconds to see if it is set for
radiance(Ground) measurement or reflectance (UAS) measurement. If it is
set on Radiance measurements, then it takes
##measurement every few seconds automatically  and in a cycle. In this
mode when pin 4 is driven, the system shuts down. And if it is set on
Reflectance, then it waits for a high
## signal converted from autopilot on pin 4 or a push button on pin 12
to be pressed.
##

while True:
    if gpio.input(25): # if switch connected to pin is on then we are
measuring the radiance and LED light connected to pin 16 will turn on
        gpio.output(16,True)

    while gpio.input(25)==1:

            ## Sutting down the system
            station='Ground_'
            if gpio.event_detected(4): # when the switch is set on
radiance and pin 4 is driven the system will shut down
                os.system('sudo shutdown -h now')
            ## Using time in file naming scheme for all three channels
and spectrometers. Once the data from each spectrometer is collected a
LED on pin 23 will start blinking


number_files_before_measurement=len(os.listdir(DataFolder))## This
value is used later to know if new files are being created


            pretime=strftime("%d%m%Y%H%M%S",time.localtime()) #Reading
the time and date from the RPi clock

txtfilename=DataFolder+'/'+station+'_'+pretime+serialnum+'R01".txt"'

b=LocalAddress+'setprefix.php?prefix='+station+'_'+pretime+serialnum+'R
&channel=0' # using time in filenaming

            with urllib.request.urlopen(b) as response30:
                html30 = response30.read()





            with
urllib.request.urlopen(LocalAddress+'startsequence.php?channel=0') as
response31: # Start taking measurement
                html31 = response31.read()
                gpio.output(23,True)
                time.sleep(1) ## it takes about 1 second for RPi to
create new files since starting measurement. By excluding this delay
the number of files before and after measurement will be the same


            if gpio.event_detected(4): # when the switch is set on
radiance and pin 4 is driven the system will shut down
                os.system('sudo shutdown -h now')



#number_files_before_measurement=len(os.listdir(DataFolder))
            pretime=strftime("%d%m%Y%H%M%S",time.localtime())

txtfilename2=DataFolder+'/'+station+'_'+pretime+serialnum2+'R01".txt"'

b2=LocalAddress+'setprefix.php?prefix='+station+'_'+pretime+serialnum2+
'R&channel=1'
            with urllib.request.urlopen(b2) as response32:
                html32 = response32.read()
            with
urllib.request.urlopen(LocalAddress+'startsequence.php?channel=1') as
response33:
                html33 = response33.read()

                time.sleep(1)

            if gpio.event_detected(4): # when the switch is set on
radiance and pin 4 is driven the system will shut down
                os.system('sudo shutdown -h now')

            pretime=strftime("%d%m%Y%H%M%S",time.localtime())

txtfilename3=DataFolder+'/'+station+'_'+pretime+serialnum3+'R01".txt"'

b3=LocalAddress+'setprefix.php?prefix='+station+'_'+pretime+serialnum3+
'R&channel=2'
            with urllib.request.urlopen(b3) as response34:
                html34 = response34.read()

            with
urllib.request.urlopen(LocalAddress+'startsequence.php?channel=2') as
response35:
                html35 = response35.read()
                time.sleep(1)

            if gpio.event_detected(4): # when the switch is set on
radiance and pin 4 is driven the system will shut down
                os.system('sudo shutdown -h now')

            ## In this for loop file extensions will be fixed. The SDK
creates text files with extra "" that before and after txt.
            ## Files with this goofy form cannot be transfered to a USB
drive. The file extensions will be fixed in this for loop.
            number_files_after_measurement=len(os.listdir(DataFolder))
            while not
number_files_after_measurement==number_files_before_measurement+9:
                time.sleep(1)

number_files_after_measurement=len(os.listdir(DataFolder))
            number_files_after_measurement=len(os.listdir(DataFolder))
            print(str(number_files_before_measurement)+'before')
            print(str(number_files_after_measurement)+'after')

            ## Reading the last measurement from reflectance and put
values in anew list
            if serialnum=='S07846' or serialnum=='S05821' or
serialnum=='S03927':
                mylist=[]
                n=0
                with open(txtfilename,'r') as f:
                    for line in f:
                        try:
                                mylist.append(float(line.split('\t')[-
1]))
                        except:
                                print('Skipping line %s...'%n)
                        n +=1


                filtered=[e for e in mylist if (np.mean(mylist)-
2*np.std(mylist)<e<np.mean(mylist)+2*np.std(mylist))] #Removing
outliers

                peak1=max(filtered)#Finding the maximum intensity

                int_time=int(int_time*x2/peak1)#Calculating the new int
time based on last measurement
                ##Setting the maximum integration time threshhold
                if int_time>10000000:
                    int_time=10000000
                with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time)+'&channel=0') as responsec0: # setting integration time on
channel0
                    htmlg = responsec0.read()
                sequenceinterval=int((int_time*average/1000)+50)
                with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval)+'&channel=0') as response8: #the interval
between each measurement in case taking a series of measurements
                    html8 = response8.read()
                print(int_time)


                mylist=[]
                n=0
                with open(txtfilename2,'r') as f:
                    for line in f:
                        try:
                                mylist.append(float(line.split('\t')[-
1]))
                        except:
                                print('Skipping line %s...'%n)
                        n +=1


                filtered=[e for e in mylist if (np.mean(mylist)-
2*np.std(mylist)<e<np.mean(mylist)+2*np.std(mylist))]
                peak2=max(filtered)
                int_time2=int(int_time2*x2/peak2)
                if int_time2>10000000:
                    int_time2=10000000

                with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time2)+'&channel=1') as responsec1: # setting integration time on
channel0
                    htmlg2 = responsec1.read()
                sequenceinterval2=int((int_time2*average2/1000)+50)
                with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval2)+'&channel=1') as response15:
                    html15 = response15.read()

                print(int_time2)
                mylist=[]
                n=0
                with open(txtfilename3,'r') as f:
                    for line in f:
                        try:
                                mylist.append(float(line.split('\t')[-
1]))
                        except:
                                print('Skipping line %s...'%n)
                        n +=1

                filtered=[e for e in mylist if (np.mean(mylist)-
2*np.std(mylist)<e<np.mean(mylist)+2*np.std(mylist))]

                peak3=max(filtered)
                int_time3=int(int_time3*x2/peak3)
                if int_time3>10000000:
                    int_time3=10000000

                with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time3)+'&channel=2') as response: # setting integration time on
channel0
                    htmlg3 = response.read()
                sequenceinterval3=int((int_time3*average3/1000)+50)
                with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval3)+'&channel=2') as response21:
                    html21 = response21.read()
                print(int_time3)
            gpio.output(23,False)
            time.sleep(10)
            for filename in glob.iglob(os.path.join(DataFolder,
'*".txt"')):
                    os.rename(filename, filename[:-6]+'.txt')



    ## When the switch is set on, reflectance measurement waits for
pulses which come from PWM to Digital converter or a manual push
button.
    while gpio.input(25)==0: # The LED connected to pin 25 will be
turned off when the switch is set on Reflectance measurement
```

```
        gpio.output(16,False)



        station='UAS_'
        if gpio.event_detected(12) or gpio.event_detected(4): ##Waiting
for pulses on pin 4 for push button or 12 comes from PWM to Digital
converter
                gpio.remove_event_detect(4) #after detecting a pulse on
the pins once, block further noise pulses comes from push button after
pressing the button just for once


number_files_before_measurement=len(os.listdir(DataFolder))## This
value is used later to know if new files are being created
                ## Using time in file naming scheme for all three
channels and spectrometers. Once the data from each spectrometer is
collected a LED on pin 23 will start blinking
                pretime=strftime("%d%m%Y%H%M%S",time.localtime())
#Reading the time and date from the RPi clock

txtfilename=DataFolder+'/'+station+'_'+pretime+serialnum+'R01".txt"'

b4=LocalAddress+'setprefix.php?prefix='+station+'_'+pretime+serialnum+'
R&channel=0'
                with urllib.request.urlopen(b4) as response300:
                    html300 = response300.read()

                with
urllib.request.urlopen(LocalAddress+'startsequence.php?channel=0') as
response310:
                    html310 = response310.read()
                    gpio.output(23,True)
                    time.sleep(1)
                    print("spectrometer on channel 1 is done taking
measurement")


                pretime=strftime("%d%m%Y%H%M%S",time.localtime())

txtfilename2=DataFolder+'/'+station+'_'+pretime+serialnum2+'R01".txt"'

b5=LocalAddress+'setprefix.php?prefix='+station+'_'+pretime+serialnum2+
'R&channel=1'
                with urllib.request.urlopen(b5) as response320:
                    html320 = response320.read()

                with
urllib.request.urlopen(LocalAddress+'startsequence.php?channel=1') as
response330:
                    html330 = response330.read()

                    time.sleep(1)
                    print("spectrometer on channel 2 is done taking
measurement")


#number_files_before_measurement=len(os.listdir(DataFolder))## This
value is used later to know if new files are being created
                pretime=strftime("%d%m%Y%H%M%S",time.localtime())

txtfilename3=DataFolder+'/'+station+'_'+pretime+serialnum3+'R01".txt"'

b6=LocalAddress+'setprefix.php?prefix='+station+'_'+pretime+serialnum3+
'R&channel=2'
                with urllib.request.urlopen(b6) as response340:
                    html340 = response340.read()

                with
urllib.request.urlopen(LocalAddress+'startsequence.php?channel=2') as
response350:
                    html350 = response350.read()

                    time.sleep(1)
                    print("spectrometer on channel 3 is done taking
measurement")

number_files_after_measurement=len(os.listdir(DataFolder))

                ## In this for loop file extensions will be fixed. The
SDK creates text files with extra "" that before and after txt.
                ## Files with this goofy form cannot be transfered to a
USB drive. The file extensions will be fixed in this for loop.


                while not
number_files_after_measurement==number_files_before_measurement+9:
                    time.sleep(1)

number_files_after_measurement=len(os.listdir(DataFolder))

number_files_after_measurement=len(os.listdir(DataFolder))
                print(str(number_files_before_measurement)+'before')
                print(str(number_files_after_measurement)+'after')



                if serialnum=='S07846' or serialnum=='S05821' or
serialnum=='S03927':
                    mylist=[]
                    n=0
                    with open(txtfilename,'r') as f:
                        for line in f:
                            try:

mylist.append(float(line.split('\t')[-1]))
```

```
                            except:
                                    print('Skipping line %s...'%n)
                            n +=1


                    filtered=[e for e in mylist if (np.mean(mylist)-
2*np.std(mylist)<e<np.mean(mylist)+2*np.std(mylist))]

                    peak1=max(filtered)

                    int_time=int(int_time*x2/peak1)
                    if int_time>10000000:
                        int_time=10000000

                    with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time)+'&channel=0') as responsec0: # resetting integration time on
channel0
                        htmlg = responsec0.read()
                    sequenceinterval=int((int_time*average/1000)+50)
                    with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval)+'&channel=0') as response8: #the interval
between each measurement in case taking a series of measurements
                        html8 = response8.read()
                    print(int_time)


                    mylist=[]
                    n=0
                    with open(txtfilename2,'r') as f:
                        for line in f:
                            try:

mylist.append(float(line.split('\t')[-1]))
                            except:
                                    print('Skipping line %s...'%n)
                            n +=1


                    filtered=[e for e in mylist if (np.mean(mylist)-
2*np.std(mylist)<e<np.mean(mylist)+2*np.std(mylist))]
                    peak2=max(filtered)
                    int_time2=int(int_time2*x2/peak2)
                    if int_time2>10000000:
                        int_time2=10000000

                    with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time2)+'&channel=1') as responsec1: # setting integration time on
channel0
                        htmlg2 = responsec1.read()
                    sequenceinterval2=int((int_time2*average2/1000)+50)
                    with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval2)+'&channel=1') as response15:
                        html15 = response15.read()

                    print(int_time2)
                    mylist=[]
                    n=0
                    with open(txtfilename3,'r') as f:
                        for line in f:
                            try:

mylist.append(float(line.split('\t')[-1]))
                            except:
                                    print('Skipping line %s...'%n)
                            n +=1

                    filtered=[e for e in mylist if (np.mean(mylist)-
2*np.std(mylist)<e<np.mean(mylist)+2*np.std(mylist))]

                    peak3=max(filtered)
                    int_time3=int(int_time3*x2/peak3)
                    if int_time3>10000000:
                        int_time3=10000000

                    with
urllib.request.urlopen(LocalAddress+'setintegration.php?time='+str(int_
time3)+'&channel=2') as response: # setting integration time on
channel0
                        htmlg3 = response.read()
                    sequenceinterval3=int((int_time3*average3/1000)+50)
                    with
urllib.request.urlopen(LocalAddress+'setsequenceinterval.php?interval='
+str(sequenceinterval3)+'&channel=2') as response21:
                        html21 = response21.read()
                    print(int_time3)

                #Checking file names to fix errors
                for filename in glob.iglob(os.path.join(DataFolder,
'*".txt"')):
                        os.rename(filename, filename[:-6]+'.txt')
                gpio.output(23,False)

                gpio.add_event_detect(4,gpio.FALLING) # Redefine an
event to detect a falling edge on pin 4 for push button

gpio.cleanup()
```

A.5. Eclipse Data Processing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Eclipse_Calibration.m % Author: Ali Hamidisepehr (c) 2016 %
% Function: This script organizes ambient light and reflectance data
% during eclipse and includes the system calibration process. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;clc;close all;
folder = pwd;% get the directory
Listing = struct2cell(dir(folder)); % list all files in the current
directory
FileNames = Listing(1,:)';% put all the file names in a column
x=1;
%Reflectance_VIS='S05821';%Radiance_VIS='S08285';
%Reflectance_NIR='S07846';%Radiance_NIR='S07821';
%Reflectance_UV='S03927';%Radiance_UV='S04413';
%IT=55;
spectrometer_code=input('1.UV 2.VIS 3.NIR');
%Applying calibration coefficients for ambient light spectrometers
load calibration_files.mat;
gains4413=S04413(:,2);
gains7821=S07821(:,2);
gains8285=S08285(:,2);
if spectrometer_code==1
    Gain=gains4413;
    figpath=([folder '\Figures for paper\UV\']);
    Rad_spec='S04413';
    Ref_spec='S03927';
    IT=70;
elseif spectrometer_code==2
    Gain=gains8285*(1000/180);
    figpath=([folder '\Figures for paper\VIS\']);
    Rad_spec='S08285';
    Ref_spec='S05821';
    IT=35;
elseif spectrometer_code==3
    Gain=gains7821;
   figpath=([folder '\Figures for paper\NIR\']);
    Rad_spec='S07821';
    Ref_spec='S07846';
    IT=55;
else
    fprintf('wrong code');
end

%this loop creates a table of measurements sorted temporally for
ambient
%light measurements
for p=1:length(Listing)
        k = strfind(char(FileNames(p)),Rad_spec); %distinguish files
which contain reflection data and ignore irrelevant data
```

```
            if isempty(k)
            %do nothing
            else

            code=1;
            filename=char(FileNames(p));
            timestamp=filename(9:end-13);

formtime=datetime(timestamp,'InputFormat','ddMMyyyyHHmmss');

formtimeweather(x,1)=datetime(timestamp,'InputFormat','ddMMyyyyHHmmss')
;

hours=hour(formtime);minutes=minute(formtime);seconds=second(formtime);
            totaltime=(hours*3600)+(minutes*60)+seconds;
            comptab(x,:)={filename totaltime code};
            x=x+1;

            end
end
rep=unique([comptab{:,2}])';%finding unique times of measurement
init=calibrationimport(char(comptab(1,1)));
unique_time_weather=unique(formtimeweather);
%Extracting ambient light data from text files and averaging
measurements
%taken at the same time
for indmeasurement=1:size(rep,1)
    data=zeros(length(init),1);
for n=1:length(comptab)
   if comptab{n,2}==rep(indmeasurement)
   filename=comptab(n,1);
   raw=calibrationimport(char(comptab(n,1)));
   data=[data raw(:,2)];

   end
end
wavelengths=raw(:,1);
data(:,1)=[];
data_ave=mean(data,2);
comptabrep(indmeasurement,:)={filename rep(indmeasurement) code};
comptabrep1(indmeasurement,:)={filename rep(indmeasurement) code};
final_data(:,indmeasurement)=data_ave;
end

%this loop creates a table of measurements sorted temporally for
%reflectance measurements
y=1;
for p=1:length(Listing)
        k = strfind(char(FileNames(p)),Ref_spec); %distinguish files
which contain reflection data and ignore irrelevant data

            if isempty(k)
            %do nothing
            else

            code=2;
            filename=char(FileNames(p));
            timestamp=filename(9:end-13);

formtime=datetime(timestamp,'InputFormat','ddMMyyyyHHmmss');

hours=hour(formtime);minutes=minute(formtime);seconds=second(formtime);
            totaltime=(hours*3600)+(minutes*60)+seconds;
            comptab(x,:)={filename totaltime code};
            comptab2(y,:)={filename totaltime code};
            x=x+1;
            y=y+1;
            end
end
rep2=unique([comptab2{:,2}])';
%Extracting reflectance data from text files and averaging measurements
%taken at the same time
for indmeasurement2=1:size(rep2,1)
    data=zeros(length(init),1);
for n=1:length(comptab2)
   if comptab2{n,2}==rep2(indmeasurement2)
   %else
   filename=comptab2(n,1);
   raw=calibrationimport(char(comptab2(n,1)));
   data=[data raw(:,2)];
   end
end
data(:,1)=[];
data_ave=mean(data,2);
%filename=comptab(n,1);
comptabrep(indmeasurement2+indmeasurement,:)={filename
rep2(indmeasurement2) code};
comptabrep2(indmeasurement2,:)={filename rep2(indmeasurement2) code};
final_data2(:,indmeasurement2)=data_ave;
end


[values idx]=sort([comptab{:,2}],'ascend');
comptab=comptab(idx,:);
[values_rep idx_rep]=sort([comptabrep{:,2}],'ascend');
comptabrep=comptabrep(idx_rep,:);
% interpolating reflectance data at times there is an ambient light
% measurement but not a reflectance measurement
for j=1:length(comptabrep2)
    for i=1:length(comptabrep1)-1
    if comptabrep2{j,2}==comptabrep1{i,2}
        inter_rad(:,j)=final_data(:,i);
    elseif comptabrep2{j,2}>comptabrep1{i,2} &&
comptabrep2{j,2}<comptabrep1{i+1,2}
```

```
        diff1=comptabrep2{j,2}-comptabrep1{i,2};
        diff2=comptabrep1{i+1,2}-comptabrep2{j,2};

inter_rad(:,j)=(final_data(:,i)*(diff1/(diff1+diff2)))+(final_data(:,i+
1)*(diff2/(diff1+diff2)));
    elseif comptabrep2{j,2}<=comptabrep1{1,2}
        inter_rad(:,j)=final_data(:,1);
    elseif comptabrep2{j,2}>=comptabrep1{end,2}
        inter_rad(:,j)=final_data(:,end);
    end
    end
end
Figure_Paper_Format;
%figure
hold on
%% Raw Radiance Figure
for p=1:size(final_data,2)
    plot(wavelengths,final_data(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.)');
ylim([0 15000])
ax=gca;
ax.YAxis.Exponent=3;
export_fig([figpath 'Radiance Before Calibration'],'-png', '-r300')
%Exporting the paper format figure

%figure
%% Interpolated radiance data to make synchronization with reflectance
data
Figure_Paper_Format;
hold on
for p=1:size(final_data2,2)
    plot(wavelengths,inter_rad(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.)');
%figure
%% calibrated radiance data using calibration coefficients provided by
the manufacturer
Figure_Paper_Format;
hold on
for u=1:size(inter_rad,2)
    corrected_rad(:,u)=(inter_rad(:,u).*Gain);
    plot(wavelengths,corrected_rad(:,u))
end
xlabel('Wavelength(nm)');ylabel('Intensity (µJ)');
ylim([0 50])
export_fig([figpath 'Radiance After Calibration'],'-png', '-r300')
%Exporting the paper format figure
%% calculating calibrated reflectance based on integration time and
calibrated radiance
paired_spectrums=final_data2./(corrected_rad*IT);%*IT
```

```
%% Raw reflectance figure
%figure
Figure_Paper_Format;
hold on
for p=1:size(final_data2,2)
    plot(wavelengths,final_data2(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.)');
ylim([0 20000])
ax=gca;
ax.YAxis.Exponent=3;
export_fig([figpath 'Reflectance Beofre Calibration'],'-png', '-
r300','-nocrop') %Exporting the paper format figure
%% Calibrated reflectance figure
%figure
Figure_Paper_Format;
hold on
for p=1:size(final_data2,2)
    plot(wavelengths,paired_spectrums(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.×µJ^{-1}×ms^{-1})');
ylim([0 30])
export_fig([figpath 'Reflectance After Calibration'],'-png', '-r300')
%Exporting the paper format figure
std_noncalib=std(final_data2,0,2);
std_calib=std(paired_spectrums,0,2);

%finding the wavelength with maximum variability in reflectance
intensity
%on each spectrometer before calibration and having histograms to
compare
%this variability before and after calibration
bb=final_data2(417,:);%VIS 417;UV number 746;NIR number 71
cc=paired_spectrums(417,:);
%figure
Figure_Paper_Format;
h=histogram(bb,10);
h.Normalization='probability';
%title('Wavelength: 675nm')
xlabel('Intensity (cnts.)');ylabel('Probability');
ylim([0 1])
xlim([1000 16000])
export_fig([figpath 'Histogram Before'],'-png', '-r300') %Exporting the
paper format figure
%figure
Figure_Paper_Format;
h2=histogram(cc,10);
h2.Normalization='probability';
xlabel('Intensity (cnts.×µJ^{-1}×ms^{-1})');ylabel('Probability');
ylim([0 1])
xlim([2 27])%xlim([1 20])%NIR%xlim([1 12])%UV %xlim([2 27])%VIS
```

```
export_fig([figpath 'Histogram After'],'-png', '-r300') %Exporting the
paper format figure
```

A.6. Target Recognition Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title: Target_Recognition.m % Author: Ali Hamidisepehr (c) 2017 %
% Function: This script organizes ambient light and reflectance data
% and prepare dataset for feeding into machine learning algorithms in
order
% to recognize targets. %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;clc;close all;
folder = pwd;% get the directory
Listing = struct2cell(dir(folder)); % list all files in the current
directory
FileNames = Listing(1,:)';% put all the file names in a column
x=1;
ns=1;
s=1;
Target_number=0;
%Reflectance_VIS='S05821';%Radiance_VIS='S08285';
%Reflectance_NIR='S07846';%Radiance_NIR='S07821';
%Reflectance_UV='S03927';%Radiance_UV='S04413';
%IT=55;

%Applying calibration coefficients to raw radiance data
spectrometer_code=input('1.UV 2.VIS 3.NIR');
%Applying calibration coefficients for ambient light spectrometers
load calibration_files.mat;
gains4413=S04413(:,2);
gains7821=S07821(:,2);
gains8285=S08285(:,2);


if spectrometer_code==1
    Gain=gains4413;
    figpath=([folder '\Figures for paper\UV\']);
    Rad_spec='S04413';
    Ref_spec='S03927';

elseif spectrometer_code==2
    Gain=gains8285*(1000/180);
    figpath=([folder '\Figures for paper\VIS\']);
    Rad_spec='S08285';
    Ref_spec='S05821';

elseif spectrometer_code==3
    Gain=gains7821;
   figpath=([folder '\Figures for paper\NIR\']);
    Rad_spec='S07821';
    Ref_spec='S07846';

else
```

```
    fprintf('wrong code');
end

%this loop creates a table of measurements sorted temporally for
ambient
%light measurements
for p=1:length(Listing)
        k = strfind(char(FileNames(p)),Rad_spec); %distinguish files
which contain reflection data and ignore irrelevant data

            if isempty(k)
            %do nothing
            else
            code=1;
            filename=char(FileNames(p));%turn fule name to character
            timestamp=filename(9:end-13);%seperate the timestamp part
of the file name

formtime=datetime(timestamp,'InputFormat','ddMMyyyyHHmmss');%define the
timestamp format

hours=hour(formtime);minutes=minute(formtime);seconds=second(formtime);
%
            totaltime=(hours*3600)+(minutes*60)+seconds;%combining
hours, minutes, and seconds to seconds
            comptab(x,:)={filename totaltime code};%Create a table each
row include file name, timestamp and the spectrometer station code
            x=x+1;

            end
end
rep=unique([comptab{:,2}])';% find just the unique timestamps. it is
because each measurement has three replication
init=calibrationimport(char(comptab(1,1)));%importing calibration
coeffecients and integration time

%Extracting ambient light data from text files and averaging
measurements
%taken at the same time
for indmeasurement=1:size(rep,1)
    data=zeros(length(init),1);
for n=1:length(comptab)
   if comptab{n,2}==rep(indmeasurement)
   %else
   filename=comptab(n,1);
   raw=calibrationimport(char(comptab(n,1)));
   data=[data raw(:,2)];
   end
end
wavelengths=raw(:,1);
data(:,1)=[];
data_ave=mean(data,2);
```

```
comptabrep(indmeasurement,:)={filename rep(indmeasurement) code};
comptabrep1(indmeasurement,:)={filename rep(indmeasurement) code};
final_data(:,indmeasurement)=data_ave; %final data include all the
measurements for radiance data
end

%this loop creates a table of measurements sorted temporally for
%reflectance measurements
y=1;
for p=1:length(Listing)
        k = strfind(char(FileNames(p)),Ref_spec); %distinguish files
which contain reflection data and ignore irrelevant data

            if isempty(k)
            %do nothing
            else
            % x=x+1;
            code=2;
            filename=char(FileNames(p));
            timestamp=filename(9:end-13);

formtime=datetime(timestamp,'InputFormat','ddMMyyyyHHmmss');

hours=hour(formtime);minutes=minute(formtime);seconds=second(formtime);
            totaltime=(hours*3600)+(minutes*60)+seconds;
            comptab(x,:)={filename totaltime code};
            comptab2(y,:)={filename totaltime code};
            x=x+1;
            y=y+1;
            end
end
rep2=unique([comptab2{:,2}])';
%Extracting reflectance data from text files and averaging measurements
%taken at the same time
check_saturation=zeros(length(wavelengths),1);
for indmeasurement2=1:size(rep2,1)
    data=zeros(length(init),1);
for n=1:length(comptab2)
   if comptab2{n,2}==rep2(indmeasurement2)
   %else
   filename=comptab2(n,1);
   raw=calibrationimport(char(comptab2(n,1)));

%IT(indmeasurement2)=(importIntegrationtime(char(comptab2(n,1))))/1000;
   data=[data raw(:,2)];
   end
end

data(:,1)=[];
data_ave=mean(data,2);
comptabrep(indmeasurement2+indmeasurement,:)={filename
rep2(indmeasurement2) code};
```

```
comptabrep2(indmeasurement2,:)={filename rep2(indmeasurement2) code};
final_data2(:,indmeasurement2)=data_ave;% including all the reflectance
data


%removing the outliers
if max(isoutlier(final_data2(:,indmeasurement2),'movmean',15))==1;
       [val
ind]=max(isoutlier(final_data2(:,indmeasurement2),'movmean',15));

final_data2(ind,indmeasurement2)=final_data2(ind+1,indmeasurement2);

end
%filtering saturated data and also low-intensity data
if max(final_data2(:,indmeasurement2))<14000 &&
max(final_data2(:,indmeasurement2))>11000
    check_saturation=[check_saturation final_data2(:,indmeasurement2)];
    non_saturated_table(ns,:)={filename rep2(indmeasurement2) code};

IT(ns)=(importIntegrationtime(char(non_saturated_table{ns,1})))/1000;
    ns=ns+1;
else
    saturated_table(s,:)={filename rep2(indmeasurement2) code};
    s=s+1;
end
end
check_saturation(:,1)=[];

[values idx]=sort([comptab{:,2}],'ascend');
comptab=comptab(idx,:);
[values_rep idx_rep]=sort([comptabrep{:,2}],'ascend');
comptabrep=comptabrep(idx_rep,:);

%Synchronizing the radiance and reflectance data by interpolating
missing
%ambient light data
for j=1:length(non_saturated_table)
    for i=1:length(comptabrep1)-1
    if non_saturated_table{j,2}==comptabrep1{i,2}
        inter_rad(:,j)=final_data(:,i);
    elseif non_saturated_table{j,2}>comptabrep1{i,2} &&
non_saturated_table{j,2}<comptabrep1{i+1,2}

        diff1=non_saturated_table{j,2}-comptabrep1{i,2};
        diff2=comptabrep1{i+1,2}-non_saturated_table{j,2};

inter_rad(:,j)=(final_data(:,i)*(diff1/(diff1+diff2)))+(final_data(:,i+
1)*(diff2/(diff1+diff2)));
    elseif non_saturated_table{j,2}<=comptabrep1{1,2}
        inter_rad(:,j)=final_data(:,1);
    elseif non_saturated_table{j,2}>=comptabrep1{end,2}
```

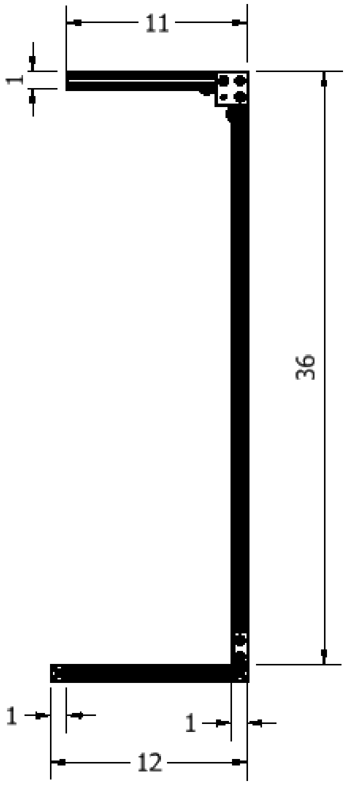
```
        inter_rad(:,j)=final_data(:,end);
    end
    end
end

Figure_Paper_Format(spectrometer_code)
%figure
hold on
%% Raw Radiance Figure
for p=1:size(final_data,2)
    plot(wavelengths,final_data(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.)');
ylim([0 12000])
ax=gca;
ax.YAxis.Exponent=3;
export_fig([figpath 'Radiance Before Calibration'],'-png', '-r300')
%Exporting the paper format figure
Figure_Paper_Format(spectrometer_code)
%figure
hold on
%% Interpolated radiance data to make synchronization with reflectance
data
for p=1:size(inter_rad,2)
    plot(wavelengths,inter_rad(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.)');

Figure_Paper_Format(spectrometer_code)
%figure
hold on
%% calibrated radiance data using calibration coefficients provided by
the manufacturer
for u=1:size(inter_rad,2)
    corrected_rad(:,u)=(inter_rad(:,u).*Gain);
    plot(wavelengths,corrected_rad(:,u))
end
xlabel('Wavelength(nm)');ylabel('Intensity (µJ)');
export_fig([figpath 'Radiance After Calibration'],'-png', '-r300')
%Exporting the paper format figure
%ylim([0 50])
IT_matrix=ones(1024,1)*IT;
%paired_spectrums=final_data2./(corrected_rad.*IT_matrix);%.*IT_matrix
paired_spectrums=(check_saturation-
1500)./(corrected_rad.*IT_matrix);%.*IT_matrix
%paired_spectrums=check_saturation./(corrected_rad.*IT_matrix);%.*IT_ma
trix
Figure_Paper_Format(spectrometer_code)
%figure
hold on
%% Raw reflectance figure
for p=1:size(final_data2,2)
```

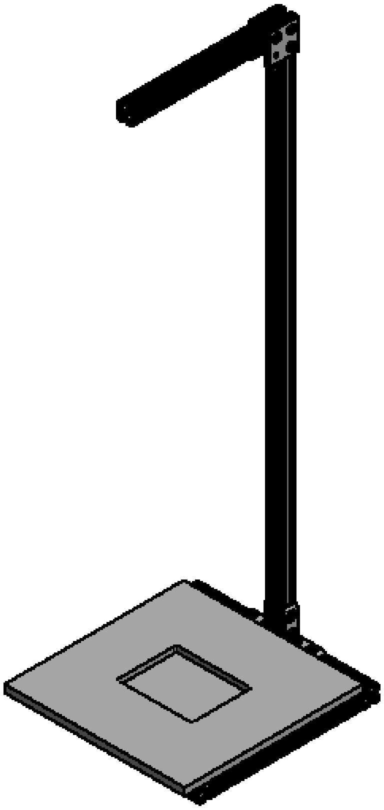
```
    plot(wavelengths,final_data2(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.)');

Figure_Paper_Format(spectrometer_code)
%figure
hold on
%% Removing saturated data from reflectance measurements
for p=1:size(check_saturation,2)
    plot(wavelengths,check_saturation(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.)');
ylim([0 15000])
ax=gca;
ax.YAxis.Exponent=3;
export_fig([figpath 'Reflectance Before Calibration'],'-png', '-r300')
%Exporting the paper format figure
Figure_Paper_Format(spectrometer_code)
%figure
hold on
%% Calibrated reflectance figure
for p=1:size(paired_spectrums,2)
    plot(wavelengths,paired_spectrums(:,p))
end
xlabel('Wavelength(nm)');ylabel('Intensity (cnts.×µJ^{-1}×ms^{-1})');
ylim([0 22]);
export_fig([figpath 'Reflectance After Calibration'],'-png', '-r300')
%Exporting the paper format figure
std_noncalib=std(final_data2,0,2);
std_calib=std(paired_spectrums,0,2);
%% Different calibration methods are considered and for each the
% corresponding dataset is formatted for feeding into machine learning
% algorithms

% Calibration with ambient light data and integration time
Machine_learning_input=(paired_spectrums)';
Machine_learning_output=ones(size(Machine_learning_input,1),1)*Target_n
umber;
Machine_learning_table=[Machine_learning_input
Machine_learning_output];

% No extra Calibration
Machine_learning_input_raw=(check_saturation)';
Machine_learning_output_raw=ones(size(Machine_learning_input_raw,1),1)*
Target_number;
Machine_learning_table_raw=[Machine_learning_input_raw
Machine_learning_output_raw];

% Calibration with ambient light data
paired_spectrums_NoIT=(check_saturation)./(corrected_rad);
Machine_learning_input_NoIT=(paired_spectrums_NoIT)';
```

```
Machine_learning_output_NoIT=ones(size(Machine_learning_input_NoIT,1),1
)*Target_number;
Machine_learning_table_NoIT=[Machine_learning_input_NoIT
Machine_learning_output_NoIT];

% Calibration with integration time
paired_spectrums_IT=(check_saturation)./(IT_matrix);%.*IT_matrix
Machine_learning_input_IT=(paired_spectrums_IT)';
Machine_learning_output_IT=ones(size(Machine_learning_input_IT,1),1)*Ta
rget_number;
Machine_learning_table_IT=[Machine_learning_input_IT
Machine_learning_output_IT];
```

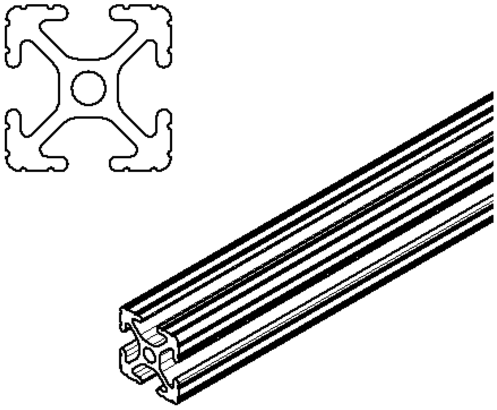
B. DRAWINGS

B.1. Test stand and T-slotted framing (units in inches)

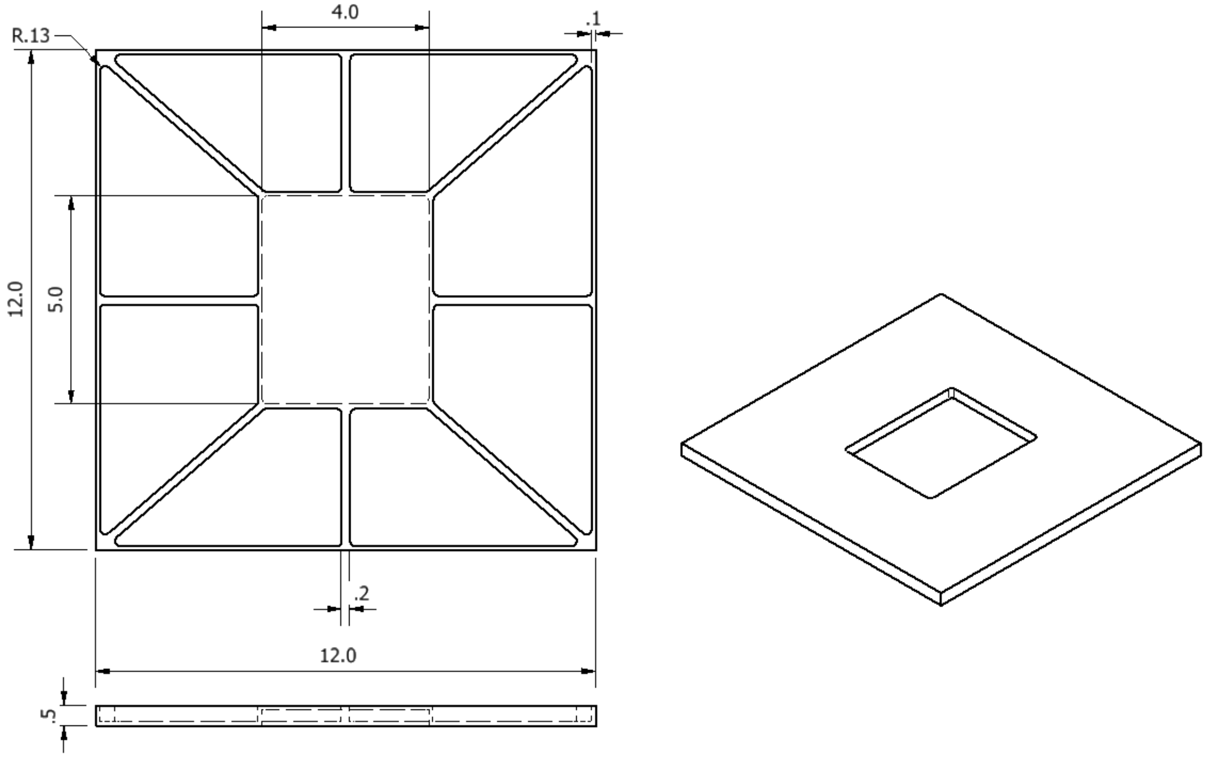




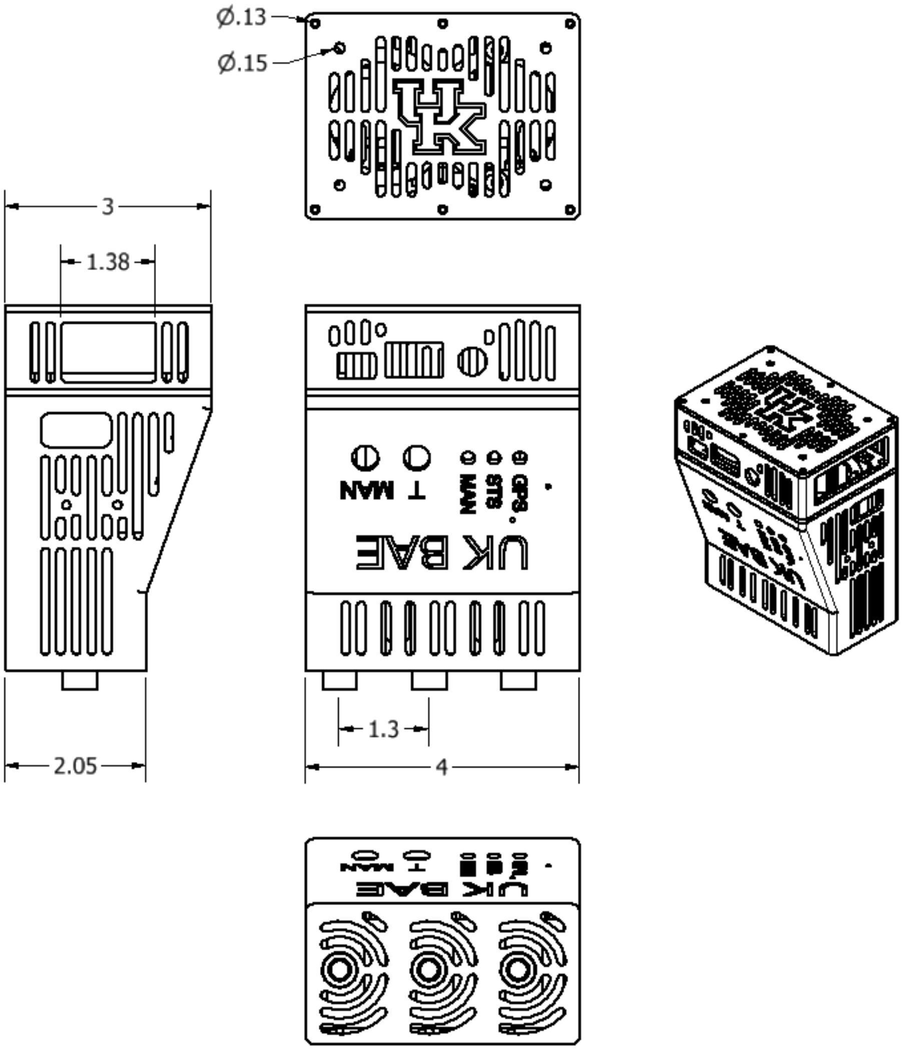
T-Slotted Framing
Single Rail, Silver, 1" High x 1" Wide, Solid

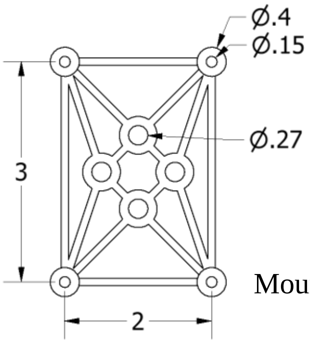


B.2. Sample holder CAD drawing (units in inches)



B.3. 3-D printed enclosure and the mounting pattern for mounting the system on the stand (units in inches)



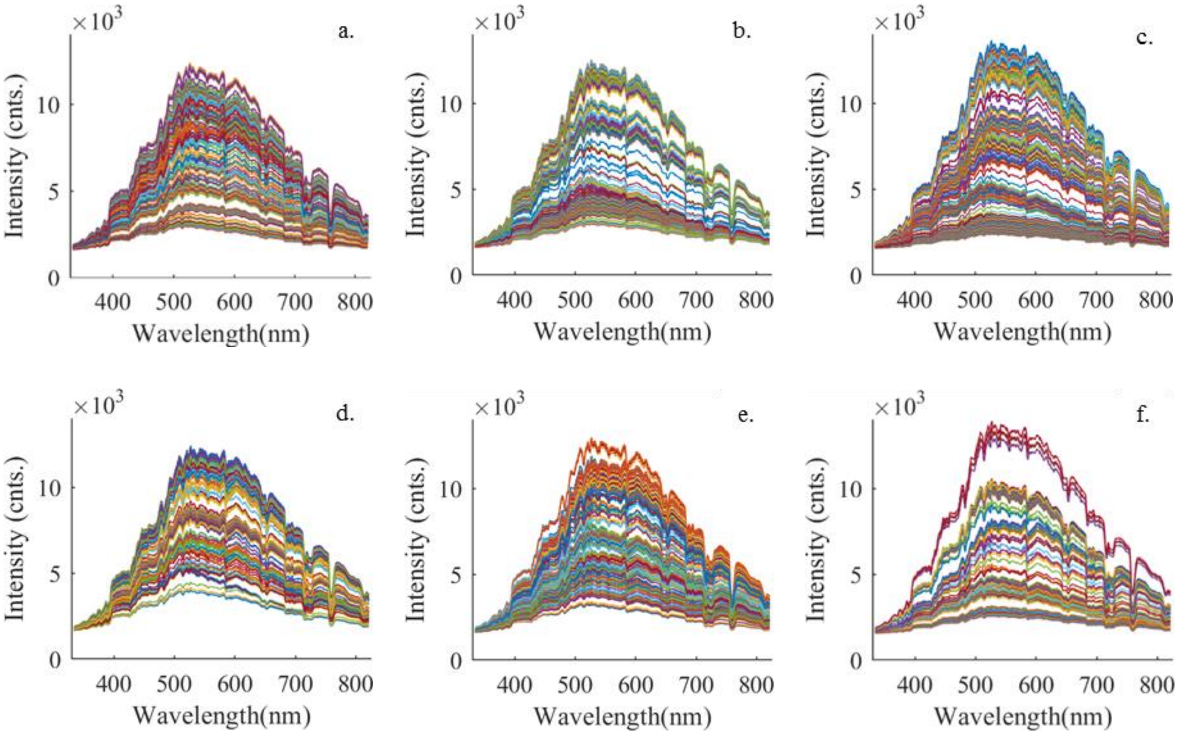


Mounting pattern

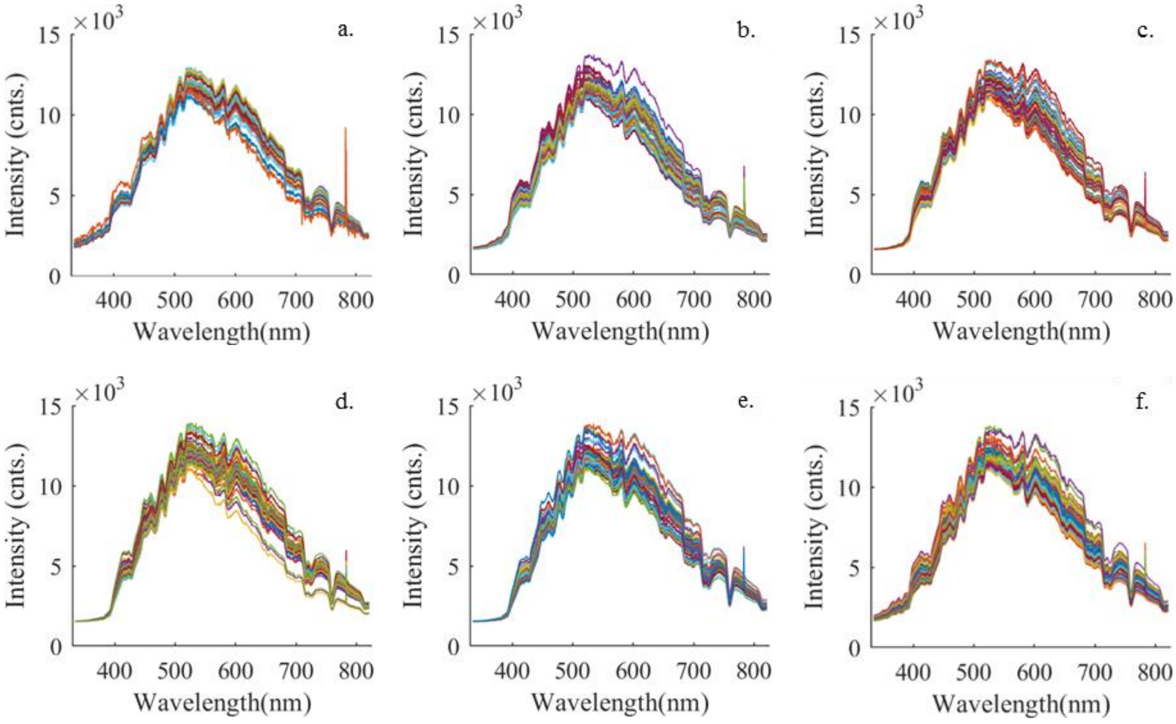
C. FIGURES

C.1. VIS and NIR data for target recognition

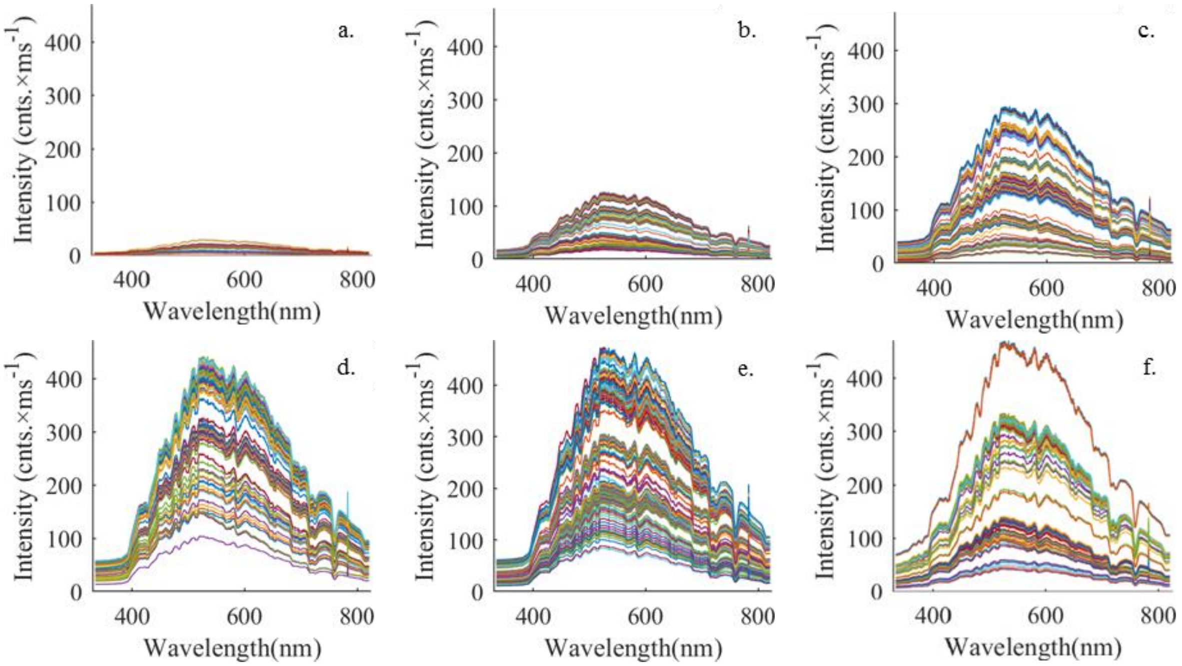
C.1.1. Raw ambient light measurements from VIS spectrometer



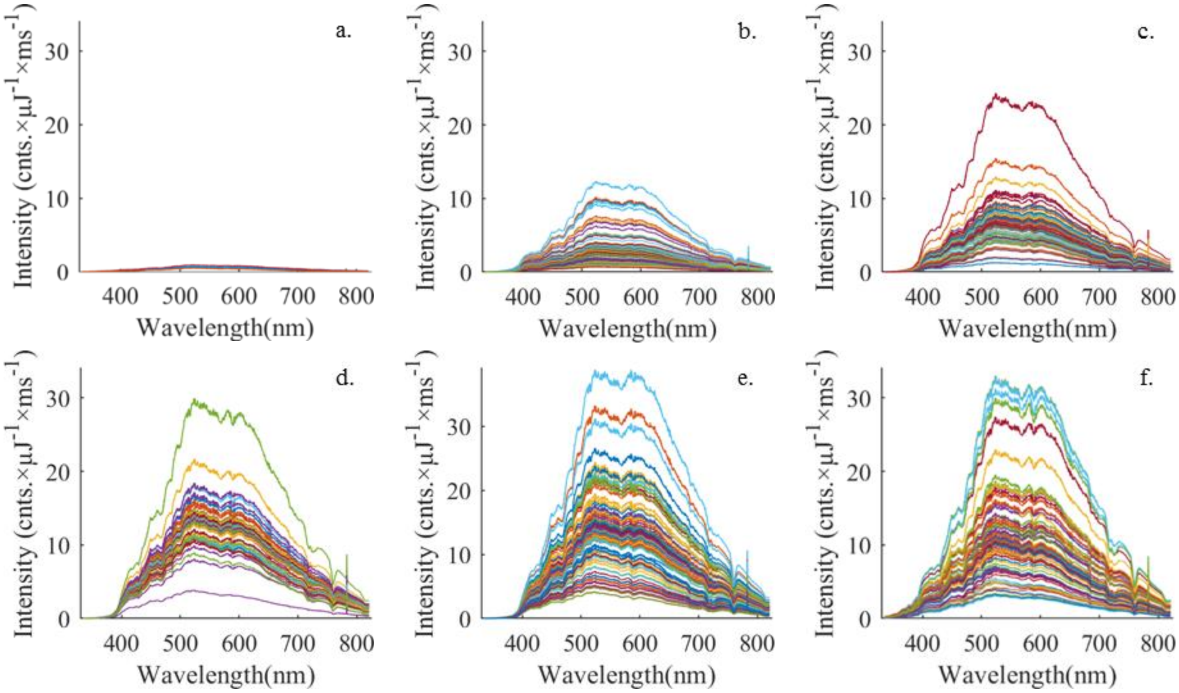
C.1.2. Reflectance measurements from VIS spectrometer with M-1 mode of compensation



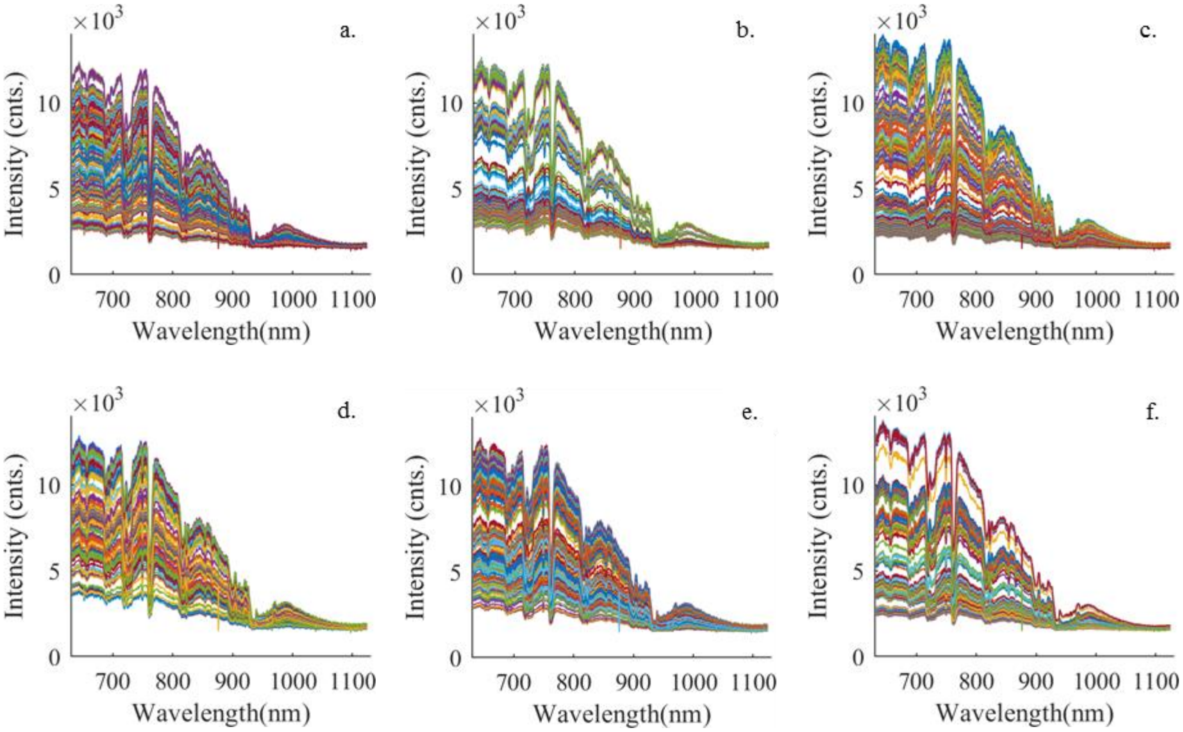
C.1.3. Reflectance measurements from VIS spectrometer with M-2 mode of compensation



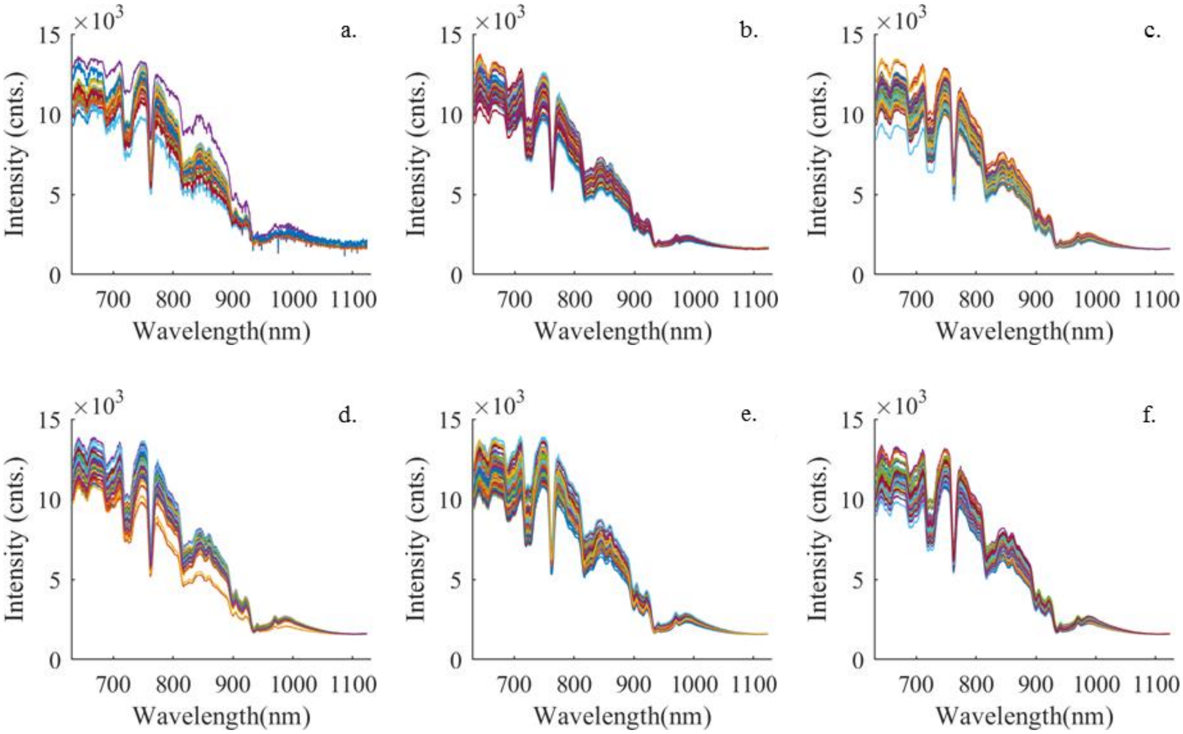
C.1.4. Reflectance measurements from VIS spectrometer with M-3 mode of compensation.



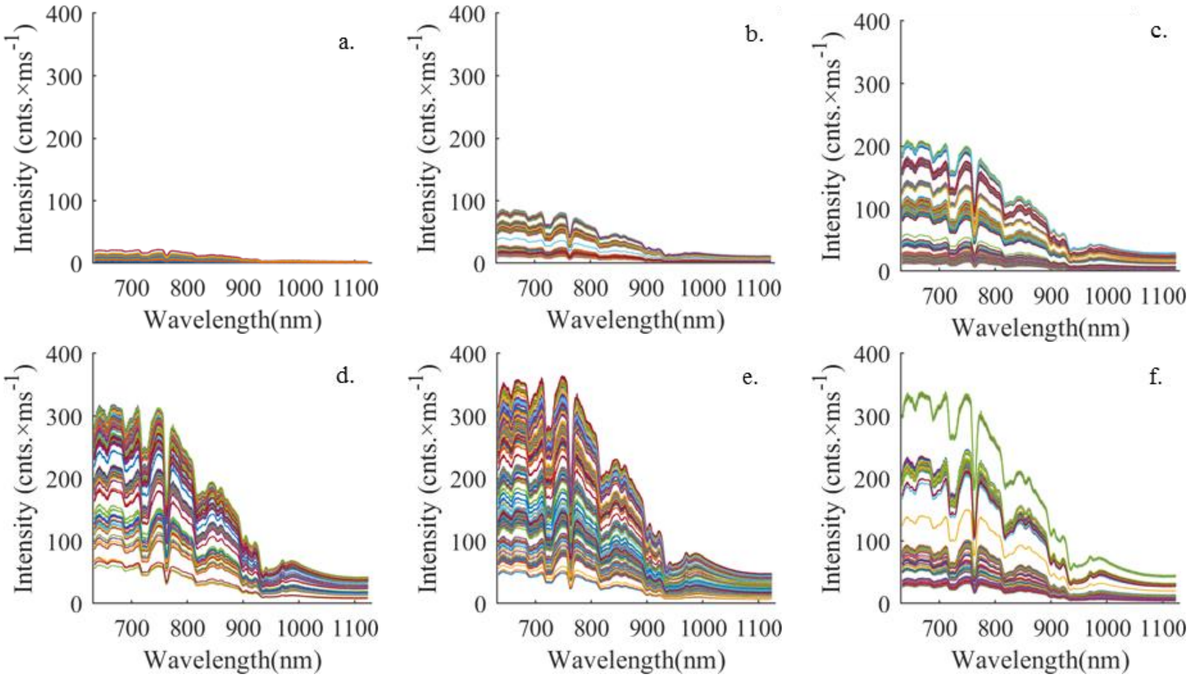
C.1.5. Raw ambient light measurements from NIR spectrometer



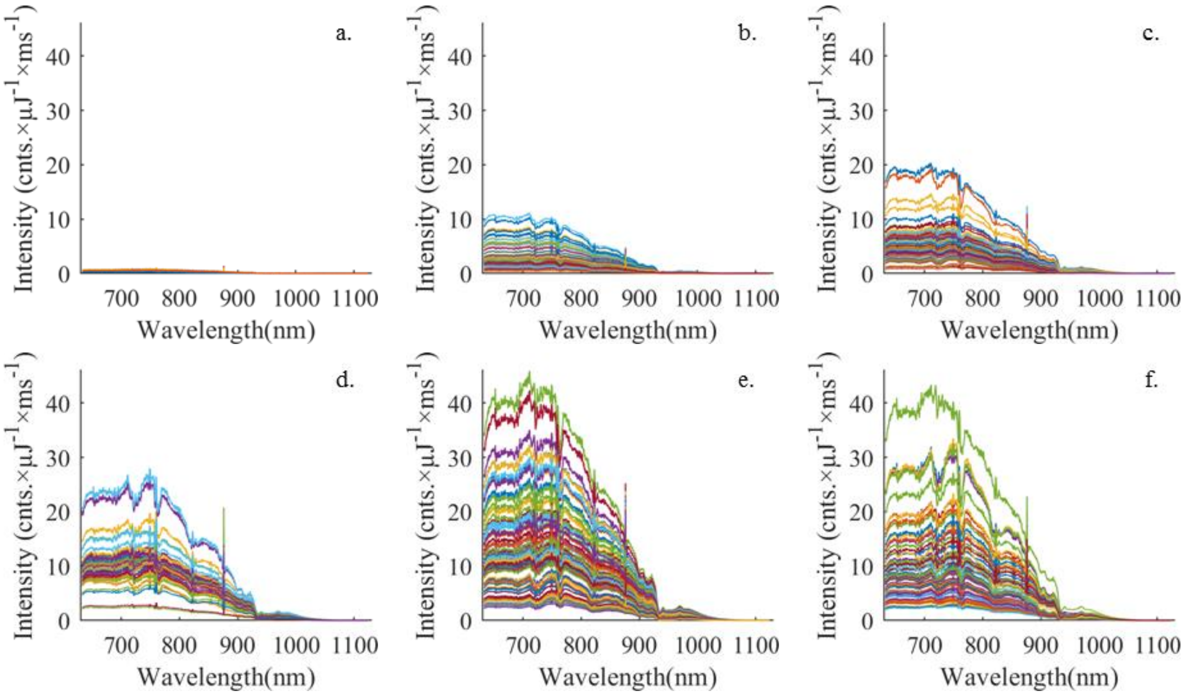
C.1.6. Reflectance measurements from NIR spectrometer with M-1 mode of compensation



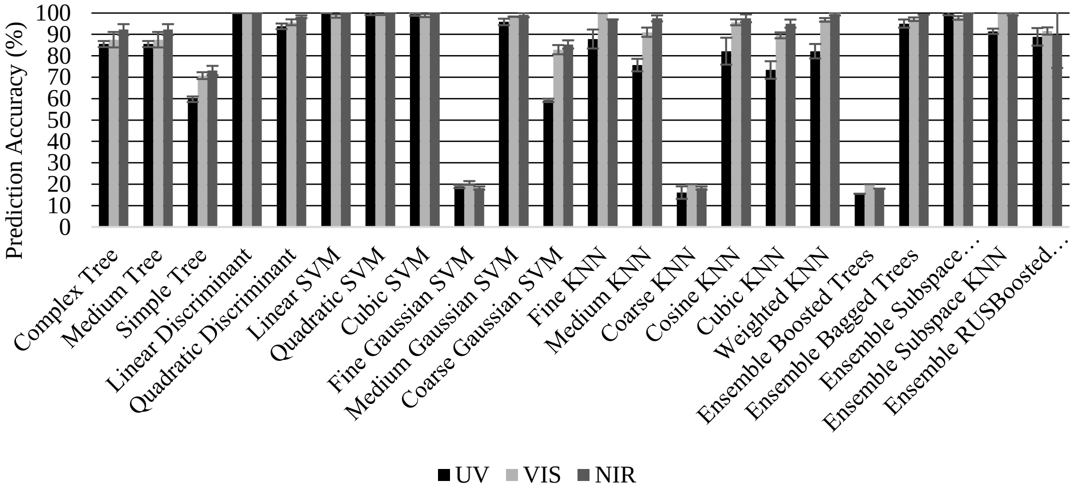
C.1.7. Reflectance measurements from VIS spectrometer with M-2 mode of compensation



C.1.8. Reflectance measurements from VIS spectrometer with M-3 mode of compensation

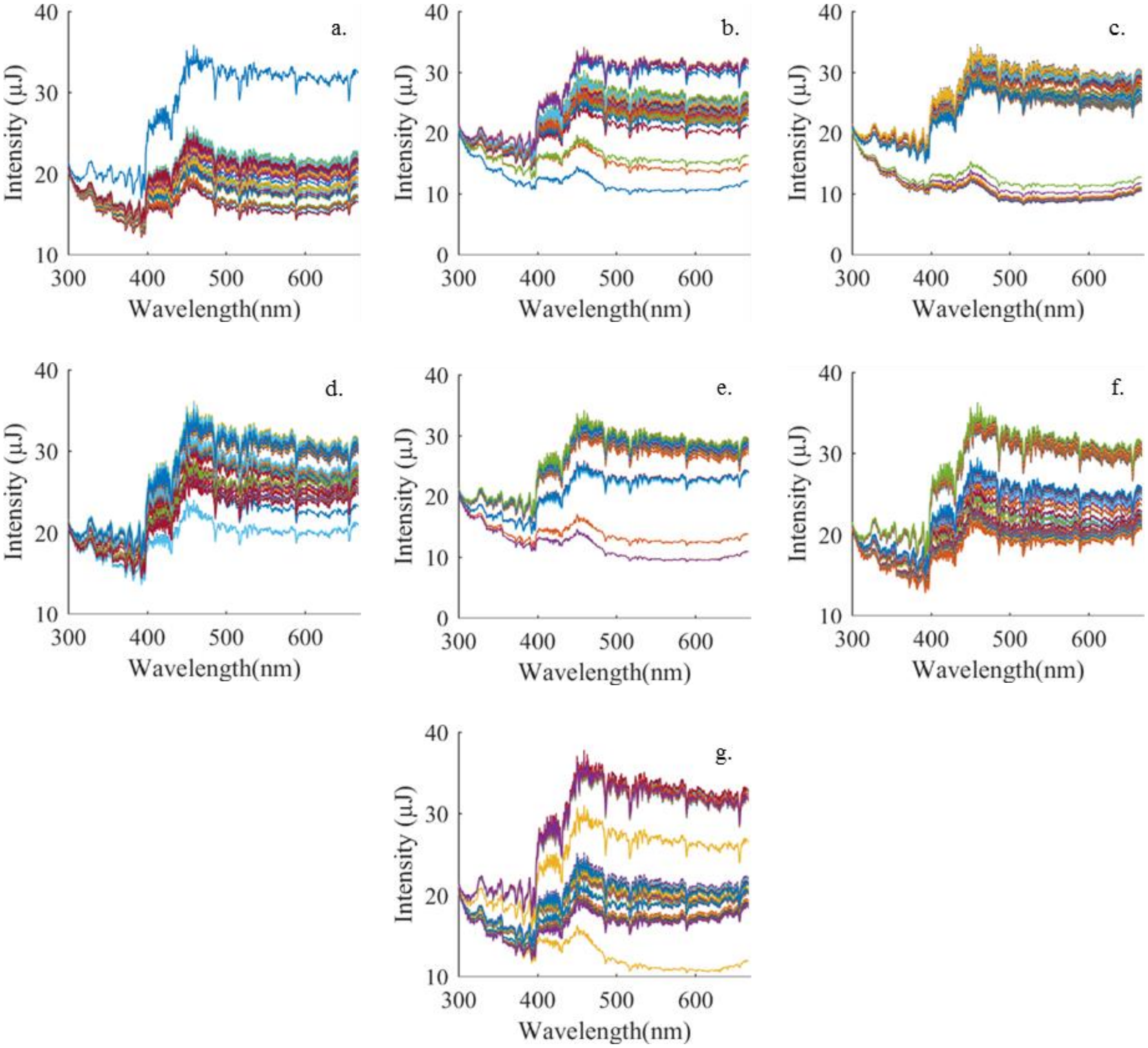


C.2. Prediction accuracy for machine learning algorithms applied to data on three types of spectrometers during a 5-minute measurement period

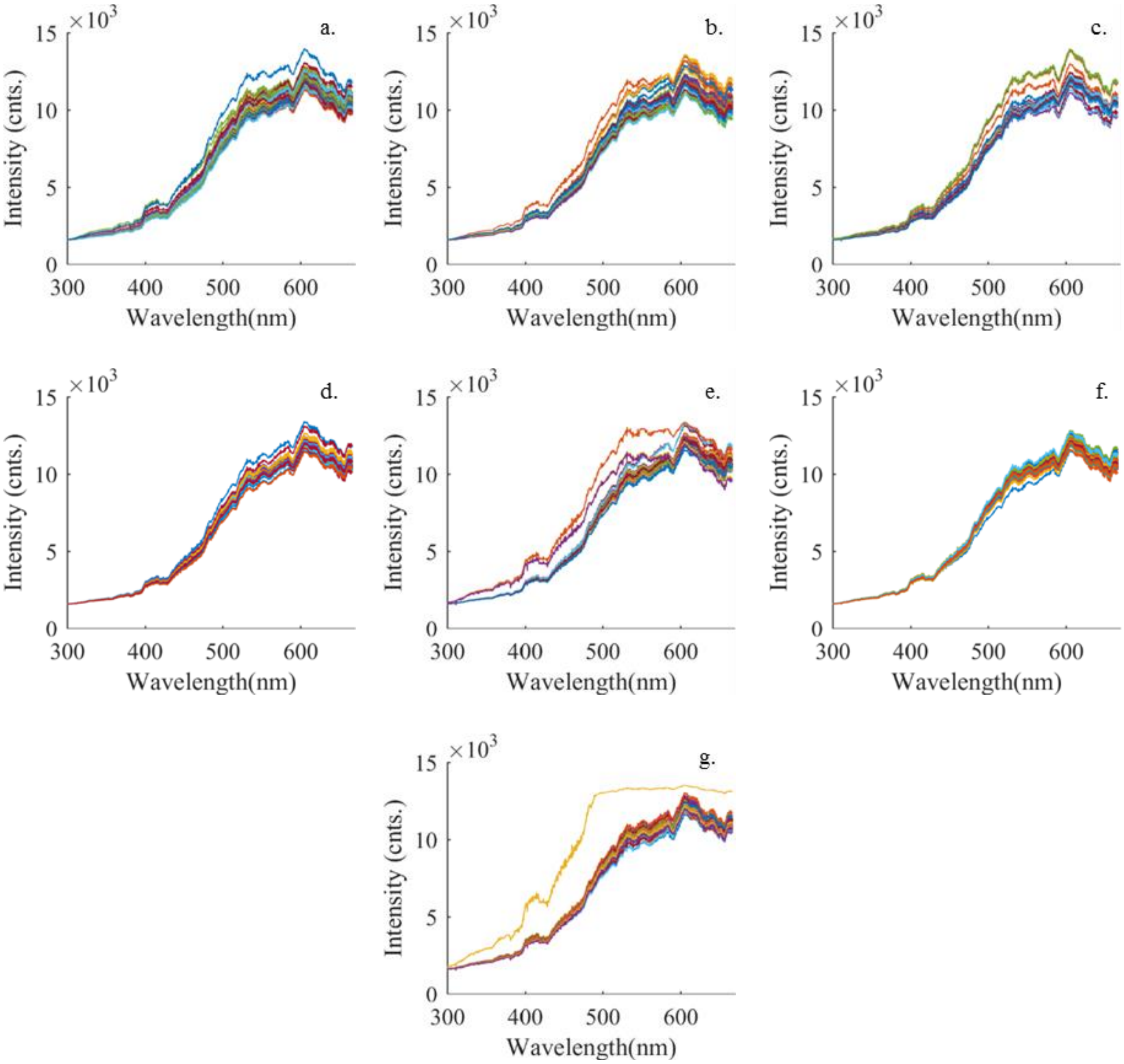


C.3. UV and VIS spectrometers data on soil samples

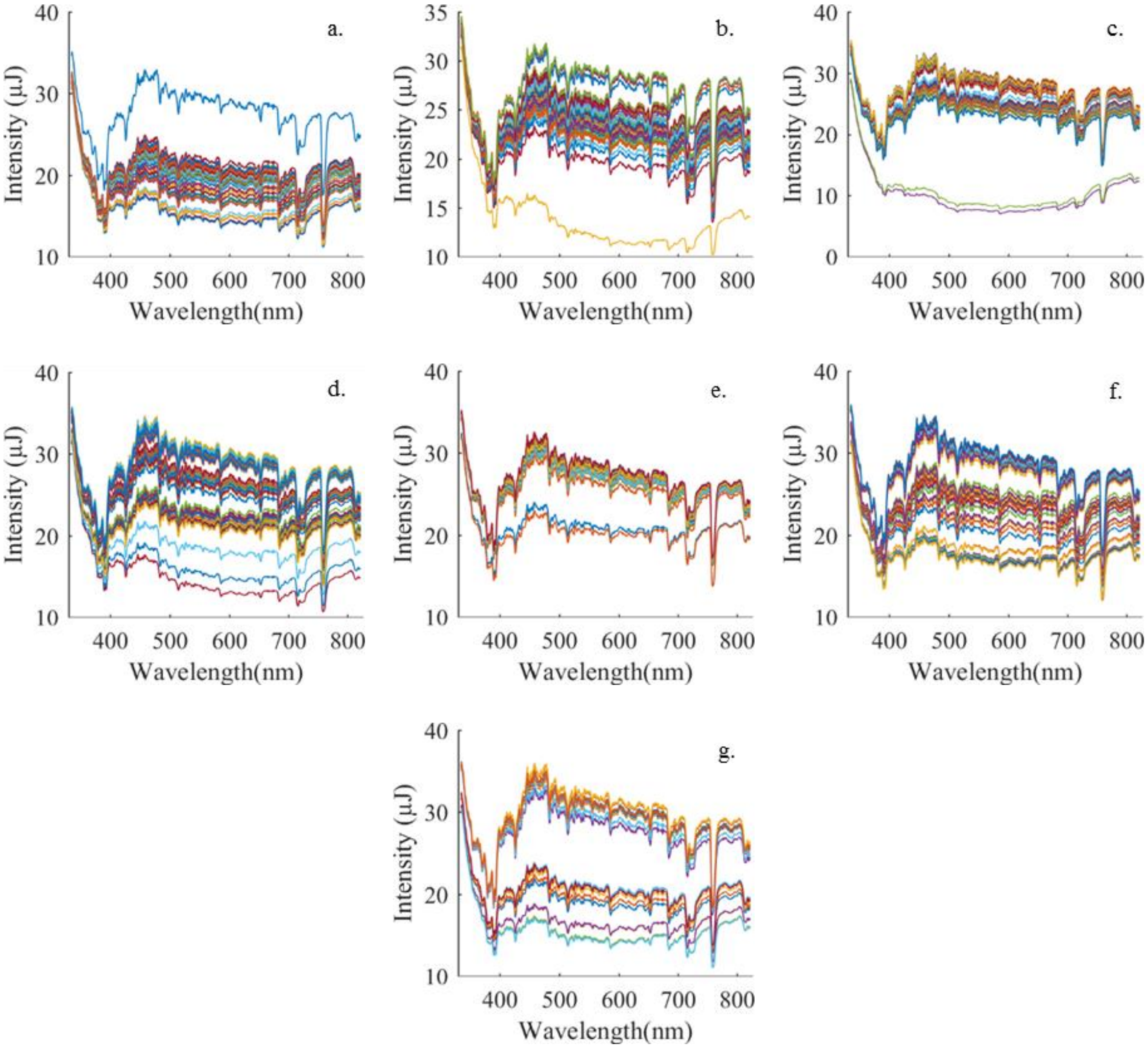
C.3.1. Calibrated ambient light measurements from UV spectrometer



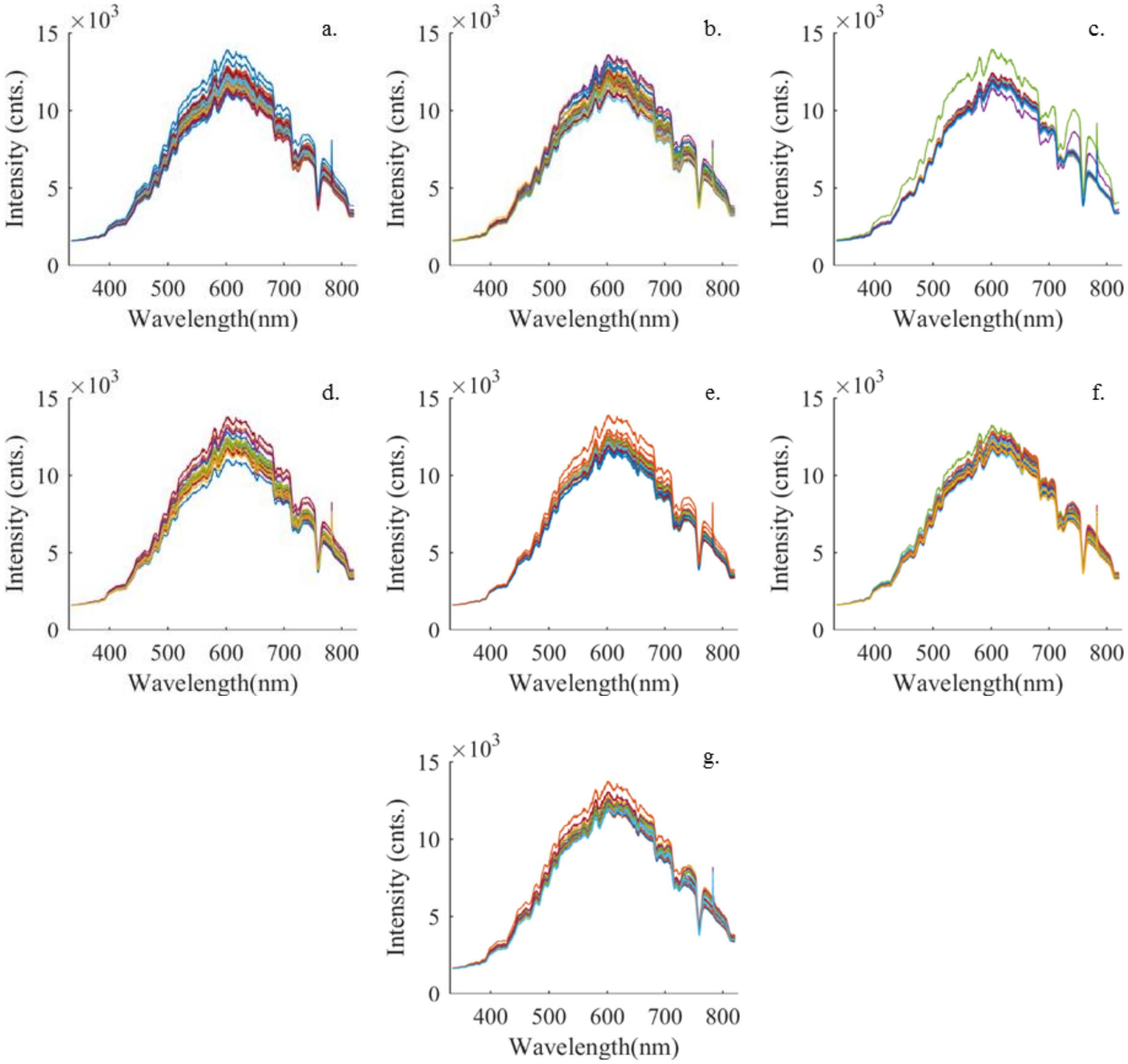
C.3.2. Reflectance measurements from UV spectrometer from soil samples



C.3.3. Calibrated ambient light measurements from VIS spectrometer.



C.3.4. Reflectance measurements from VIS spectrometer from soil samples



BIBLIOGRAPHY

Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., & Sousa, J. (2017). Hyperspectral Imaging: A Review on UAV-Based Sensors, Data Processing and Applications for Agriculture and Forestry. *Remote Sensing, 9*(11), 1110.

Ahmad, S., Kalra, A., & Stephen, H. (2010). Estimating soil moisture using remote sensing data: A machine learning approach. *Advances in Water Resources, 33*(1), 69-80.

Ali, I., Greifeneder, F., Stamenkovic, J., Neumann, M., & Notarnicola, C. (2015). Review of machine learning approaches for biomass and soil moisture retrievals from remote sensing data. *Remote Sensing, 7*(12), 16398-16421.

Atzberger, C. (2013). Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs. *Remote Sensing, 5*(2), 949-981.

Bajwa, S., & Tian, L. (2005). Soil fertility characterization in agricultural fields using hyperspectral remote sensing. *Transactions of the ASAE, 48*(6), 2399-2406.

Behmann, J., Mahlein, A.-K., Rumpf, T., Römer, C., & Plümer, L. (2015). A review of advanced machine learning methods for the detection of biotic stress in precision crop protection. *Precision agriculture, 16*(3), 239-260.

Bendig, J., Bolten, A., Bennertz, S., Broscheit, J., Eichfuss, S., & Bareth, G. (2014). Estimating biomass of barley using crop surface models (CSMs) derived from UAV-based RGB imaging. *Remote Sensing, 6*(11), 10395-10412.

Benedetti, R., & Rossini, P. (1993). On the use of NDVI profiles as a tool for agricultural statistics: the case study of wheat yield estimate and forecast in Emilia Romagna. *Remote sensing of Environment, 45*(3), 311-326.

Bernardes, T., Moreira, M. A., Adami, M., Giarolla, A., & Rudorff, B. F. T. (2012). Monitoring biennial bearing effect on coffee yield using MODIS remote sensing imagery. *Remote Sensing, 4*(9), 2492-2509.

Berni, J., Zarco-Tejada, P., Suárez, L., González-Dugo, V., & Fereres, E. (2009). Remote sensing of vegetation from UAV platforms using lightweight multispectral and thermal imaging sensors. *Int. Arch. Photogramm. Remote Sens. Spatial Inform. Sci, 38*(6).

Bokolonga, E., Hauhana, M., Rollings, N., Aitchison, D., Assaf, M. H., Das, S. R., . . . Petriu, E. M. (2016). *A compact multispectral image capture unit for deployment on drones.* Paper presented at the Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International.

Buck, S., Nemati, M., & Sunding, D. (2016). *The Welfare Consequences of the 2015 California Drought Mandate: Evidence from New Results on Monthly Water*

Demand. Paper presented at the 2016 Annual Meeting, July 31-August 2, 2016, Boston, Massachusetts.

Burkart, A., Cogliati, S., Schickling, A., & Rascher, U. (2014). A novel UAV-based ultra-light weight spectrometer for field spectroscopy. *IEEE sensors journal, 14*(1), 62-67. doi:10.1109/jsen.2013.2279720

Cai, X., & Rosegrant, M. W. (2002). Global water demand and supply projections: Part 1. A modeling approach. *Water International, 27*(2), 159-169.

Candiago, S., Remondino, F., De Giglio, M., Dubbini, M., & Gattelli, M. (2015). Evaluating multispectral images and vegetation indices for precision farming applications from UAV images. *Remote Sensing, 7*(4), 4026-4047. doi:10.3390/rs70404026

Cardinale, B. J., Duffy, J. E., Gonzalez, A., Hooper, D. U., Perrings, C., Venail, P., . . . Wardle, D. A. (2012). Biodiversity loss and its impact on humanity. *Nature, 486*(7401), 59-67.

Carlson, T. N., Dodd, J. K., Benjamin, S. G., & Cooper, J. N. (1981). Satellite estimation of the surface energy balance, moisture availability and thermal inertia. *Journal of Applied Meteorology, 20*(1), 67-87.

Carlson, T. N., & Ripley, D. A. (1997). On the relation between NDVI, fractional vegetation cover, and leaf area index. *Remote sensing of Environment, 62*(3), 241-252.

Cassman, K. G. (1999). Ecological intensification of cereal production systems: yield potential, soil quality, and precision agriculture. *Proceedings of the National Academy of Sciences, 96*(11), 5952-5959.

Cho, S., Lee, D., & Jeong, J. (2002). AE—Automation and Emerging Technologies: Weed–plant Discrimination by Machine Vision and Artificial Neural Network. *Biosystems engineering, 83*(3), 275-280.

Chrétien, L., Théau, J., & Ménard, P. (2015). Wildlife multispecies remote sensing using visible and thermal infrared imagery acquired from an unmanned aerial vehicle (UAV). *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40*(1), 241. doi:10.5194/isprsarchives-xl-1-w4-241-2015

Cocks, T., Jenssen, R., Stewart, A., Wilson, I., & Shields, T. (1998). *The HyMapTM airborne hyperspectral sensor: the system, calibration and performance.* Paper presented at the Proceedings of the 1st EARSeL workshop on Imaging Spectroscopy.

Damm, A., Erler, A., Hillen, W., Meroni, M., Schaepman, M. E., Verhoef, W., & Rascher, U. (2011). Modeling the impact of spectral sensor configurations on the FLD retrieval accuracy of sun-induced chlorophyll fluorescence. *Remote sensing of Environment, 115*(8), 1882-1892.

Díaz-Varela, R. A., de la Rosa, R., León, L., & Zarco-Tejada, P. J. (2015). High-resolution airborne UAV imagery to assess Olive tree crown parameters using 3D photo reconstruction: Application in breeding trials. *Remote Sensing, 7*(4), 4213-4232. doi:10.3390/rs70404213

Doraiswamy, P. C., Sinclair, T. R., Hollinger, S., Akhmedov, B., Stern, A., & Prueger, J. (2005). Application of MODIS derived parameters for regional crop yield assessment. *Remote sensing of Environment, 97*(2), 192-202.

Eismann, M. T. (2012). Hyperspectral Remote Sensing (pp. 417-450): SPIE Bellingham. doi:10.1117/3.899758

Gao, B.-C. (1996). NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. *Remote sensing of Environment, 58*(3), 257-266.

Gao, B.-C., Montes, M. J., & Davis, C. O. (2002). *A curve-fitting technique to improve wavelength calibrations of imaging spectrometer data.* Paper presented at the Proc. 11th Annu. JPL Airborne Earth Sci. Workshop.

Gao, B.-C., Montes, M. J., & Davis, C. O. (2004). Refinement of wavelength calibrations of hyperspectral imaging data using a spectrum-matching technique. *Remote sensing of Environment, 90*(4), 424-433.

Geipel, J., Link, J., & Claupein, W. (2014). Combined spectral and spatial modeling of corn yield based on aerial images and crop surface models acquired with an unmanned aircraft system. *Remote Sensing, 6*(11), 10335-10355. doi:10.3390/rs61110335

Giada, S., De Groeve, T., Ehrlich, D., & Soille, P. (2003). Information extraction from very high resolution satellite imagery over Lukole refugee camp, Tanzania. *International Journal of Remote Sensing, 24*(22), 4251-4266.

Gleick, P. H. (2003). Global freshwater resources: soft-path solutions for the 21st century. *Science, 302*(5650), 1524-1528.

Gnyp, M., Panitzki, M., Reusch, S., Jasper, J., Bolten, A., & Bareth, G. (2016). *Comparison between tractor-based and UAV-based spectrometer measurements in winter wheat.* Paper presented at the 13th International Conference on Precision Agriculture. International Society of Precision Agriculture, St. Louis, Missouri.

Gu, Y., Brown, J. F., Verdin, J. P., & Wardlow, B. (2007). A five-year analysis of MODIS NDVI and NDWI for grassland drought assessment over the central Great Plains of the United States. *Geophysical Research Letters, 34*(6).

Guanter, L., Richter, R., & Moreno, J. (2006). Spectral calibration of hyperspectral imagery using atmospheric absorption features. *Applied optics, 45*(10), 2360-2370.

Hakala, T., Markelin, L., Honkavaara, E., Scott, B., Theocharous, T., Nevalainen, O., . . . Greenwell, C. (2018). Direct Reflectance Measurements from Drones: Sensor

Absolute Radiometric Calibration and System Tests for Forest Reflectance Characterization. *Sensors (Basel, Switzerland), 18*(5).

Hamidisepehr, A., & Sama, M. P. (2018). *A low-cost method for collecting hyperspectral measurements from a small unmanned aircraft system.* Paper presented at the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping III.

Hamidisepehr, A., Sama, M. P., Turner, A. P., & Wendroth, O. O. (2017). A Method for Reflectance Index Wavelength Selection from Moisture-Controlled Soil and Crop Residue Samples. *Transactions of the ASABE, 60*(5), 1479-1487. doi:10.13031/trans.12172

Hatfield, J. L. (2015). Environmental impact of water use in agriculture. *Agronomy Journal, 107*(4), 1554-1556.

HOBO_DataLogger. Silicon Pyranometer Smart Sensor (S-LIB-M003) Manual. Retrieved from http://www.onsetcomp.com/files/manual_pdfs/6708-F%20MAN-S-LIB.pdf

Holland, K., Schepers, J., Shanahan, J., Horst, G., & Mulla, D. (2004). *Plant canopy sensor with modulated polychromatic light source.* Paper presented at the Proceedings of the 7th International Conference on Precision Agriculture and Other Precision Resources Management, Hyatt Regency, Minneapolis, MN, USA, 25-28 July, 2004.

Huang, J., Chen, D., & Cosh, M. (2009). Sub-pixel reflectance unmixing in estimating vegetation water content and dry biomass of corn and soybeans cropland using normalized difference water index (NDWI) from satellites. *International Journal of Remote Sensing, 30*(8), 2075-2104.

Huete, A. R. (1988). A soil-adjusted vegetation index (SAVI). *Remote sensing of Environment, 25*(3), 295-309.

Hung, C., Xu, Z., & Sukkarieh, S. (2014). Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav. *Remote Sensing, 6*(12), 12037-12054. doi:10.3390/rs61212037

Hunt, E. R., Cavigelli, M., Daughtry, C. S., Mcmurtrey, J. E., & Walthall, C. L. (2005). Evaluation of digital photography from model aircraft for remote sensing of crop biomass and nitrogen status. *Precision agriculture, 6*(4), 359-378.

Im, J., Park, S., Rhee, J., Baik, J., & Choi, M. (2016). Downscaling of AMSR-E soil moisture with MODIS products using machine learning approaches. *Environmental Earth Sciences, 75*(15), 1120.

Jackson, R. D., & Huete, A. R. (1991). Interpreting vegetation indices. *Preventive veterinary medicine, 11*(3-4), 185-200.

Jafari, A., Mohtasebi, S. S., Jahromi, H. E., & Omid, M. (2006). Weed detection in sugar beet fields using machine vision. *Int. J. Agric. Biol, 8*(5), 602-605.

Jensen, J., Qiu, F., & Ji, M. (1999). Predictive modelling of coniferous forest age using statistical and artificial neural network approaches applied to remote sensor data. *International Journal of Remote Sensing, 20*(14), 2805-2822.

Kaleita, A. L., Tian, L. F., & Hirschi, M. C. (2005). Relationship between soil moisture content and soil surface reflectance. *Transactions of the ASAE, 48*(5), 1979-1986.

Kalisperakis, I., Stentoumis, C., Grammatikopoulos, L., & Karantzalos, K. (2015). Leaf area index estimation in vineyards from UAV hyperspectral data, 2D image mosaics and 3D canopy surface models. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40*(1), 299.

Kang, S., Shi, P., Pan, Y., Liang, Z., Hu, X., & Zhang, J. (2000). Soil water distribution, uniformity and water-use efficiency under alternate furrow irrigation in arid areas. *Irrigation Science, 19*(4), 181-190. doi:10.1007/s002710000019

Kelcey, J., & Lucieer, A. (2012). Sensor correction of a 6-band multispectral imaging sensor for UAV remote sensing. *Remote Sensing, 4*(5), 1462-1493.

Khanal, S., Fulton, J., & Shearer, S. (2017). An overview of current and potential applications of thermal remote sensing in precision agriculture. *Computers and electronics in agriculture, 139*, 22-32.

Kim, Y. H., Im, J., Ha, H. K., Choi, J.-K., & Ha, S. (2014). Machine learning approaches to coastal water quality monitoring using GOCI satellite data. *GIScience & Remote Sensing, 51*(2), 158-174.

Koot, T. M. (2014). *Weed detection with unmanned aerial vehicles in agricultural systems.* (Master's Thesis), Wageningen University and Research Centre.

Lee, W., Alchanatis, V., Yang, C., Hirafuji, M., Moshou, D., & Li, C. (2010). Sensing technologies for precision specialty crop production. *Computers and electronics in agriculture, 74*(1), 2-33.

Lelong, C. C., Burger, P., Jubelin, G., Roux, B., Labbé, S., & Baret, F. (2008). Assessment of unmanned aerial vehicles imagery for quantitative monitoring of wheat crop in small plots. *Sensors, 8*(5), 3557-3585.

Lobell, D. B., & Asner, G. P. (2002). Moisture effects on soil reflectance. *Soil Science Society of America Journal, 66*(3), 722-727.

Matese, A., Toscano, P., Di Gennaro, S. F., Genesio, L., Vaccari, F. P., Primicerio, J., . . . Gioli, B. (2015). Intercomparison of uav, aircraft and satellite remote sensing platforms for precision viticulture. *Remote Sensing, 7*(3), 2971-2990. doi:10.3390/rs70302971

Matveeva, F., Mousavi, S. A. S., Zhang, X., Seigler, T., & Hoagg, J. B. (2016). *On the effects of changing reference command as humans learn to control dynamic systems.* Paper presented at the Decision and Control (CDC), 2016 IEEE 55th Conference on.

Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing, 42*(8), 1778-1790.

Montes de Oca, A., Arreola, L., Flores, A., Sanchez, J., & Flores, G. (2018). *Low-cost multispectral imaging system for crop monitoring*.

Moreno, R., Corona, F., Lendasse, A., Graña, M., & Galvão, L. S. (2014). Extreme learning machines for soybean classification in remote sensing hyperspectral images. *Neurocomputing, 128*, 207-216.

Mousavi, S. A. S., Zhang, X., Seigler, T., & Hoagg, J. B. (2016). *Characteristics that make dynamic systems difficult for a human to control.* Paper presented at the American Control Conference (ACC), 2016.

Mulla, D. J. (2013). Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems engineering, 114*(4), 358-371.

Mullen, R. W., Freeman, K. W., Raun, W. R., Johnson, G. V., Stone, M. L., & Solie, J. B. (2003). Identifying an in-season response index and the potential to increase wheat yield with nitrogen. *Agronomy Journal, 95*(2), 347-351.

Nay, J. J., Burchfield, E., & Gilligan, J. (2016). A Machine Learning Approach to Forecasting Remotely Sensed Vegetation Health. *arXiv preprint arXiv:1602.06335*.

Nay, J. J., Burchfield, E., & Gilliganc, J. (2016). A Machine Learning Approach to Forecasting Remotely Sensed Vegetation Health.

Nemani, R. R., & Running, S. W. (1989). Estimation of regional surface resistance to evapotranspiration from NDVI and thermal-IR AVHRR data. *Journal of Applied Meteorology, 28*(4), 276-284.

Notarnicola, C., Angiulli, M., & Posa, F. (2008). Soil moisture retrieval from remotely sensed data: Neural network approach versus Bayesian method. *IEEE Transactions on Geoscience and Remote Sensing, 46*(2), 547-557.

O'Shaughnessy, S. A., Evett, S. R., & Colaizzi, P. D. (2015). Dynamic prescription maps for site-specific variable rate irrigation of cotton. *Agricultural Water Management, 159,* 123-138. doi:10.1016/j.agwat.2015.06.001

Ozdogan, M., Yang, Y., Allez, G., & Cervantes, C. (2010). Remote sensing of irrigated agriculture: Opportunities and challenges. *Remote Sensing, 2*(9), 2274-2304.

Pádua, L., Vanko, J., Hruška, J., Adão, T., Sousa, J. J., Peres, E., & Morais, R. (2017). UAS, sensors, and data processing in agroforestry: a review towards practical applications. *International Journal of Remote Sensing, 38*(8-10), 2349-2391. doi:10.1080/01431161.2017.1297548

Paloscia, S., Pampaloni, P., Pettinato, S., & Santi, E. (2008). A comparison of algorithms for retrieving soil moisture from ENVISAT/ASAR images. *IEEE Transactions on Geoscience and Remote Sensing, 46*(10), 3274-3284.

Paredes, J. A., González, J., Saito, C., & Flores, A. (2017). *Multispectral imaging system with UAV integration capabilities for crop analysis.* Paper presented at the Geoscience and Remote Sensing (GRSS-CHILE), 2017 First IEEE International Symposium of.

Park, S., Im, J., Jang, E., & Rhee, J. (2016). Drought assessment and monitoring through blending of multi-sensor indices using machine learning approaches for different climate regions. *Agricultural and forest meteorology, 216*, 157-169.

Pasolli, L., Notarnicola, C., & Bruzzone, L. (2011). Estimating soil moisture with the support vector regression technique. *IEEE Geoscience and remote sensing letters, 8*(6), 1080-1084.

Peñuelas, J., Pinol, J., Ogaya, R., & Filella, I. (1997). Estimation of plant water concentration by the reflectance water index WI (R900/R970). *International Journal of Remote Sensing, 18*(13), 2869-2875.

Perry, C. (2007). Efficient irrigation; inefficient communication; flawed recommendations. *Irrigation and drainage, 56*(4), 367-378. doi:10.1002/ird.323

Piron, A., van der Heijden, F., & Destain, M.-F. (2011). Weed detection in 3D images. *Precision agriculture, 12*(5), 607-622.

Rabatel, G., Gorretta, N., & Labbe, S. (2014). Getting simultaneous red and near-infrared band data from a single digital camera for plant monitoring applications: Theoretical and practical study. *Biosystems engineering, 117*, 2-14.

Rango, A., Laliberte, A., Herrick, J. E., Winters, C., Havstad, K., Steele, C., & Browning, D. (2009). Unmanned aerial vehicle-based remote sensing for rangeland assessment, monitoring, and management. *Journal of Applied Remote Sensing, 3*(1), 033542.

Raun, W. R., Solie, J. B., Johnson, G. V., Stone, M. L., Mullen, R. W., Freeman, K. W., . . . Lukina, E. V. (2002). Improving nitrogen use efficiency in cereal grain production with optical sensing and variable rate application. *Agronomy Journal, 94*(4), 815-820.

Rosegrant, M. W., Msangi, S., Ringler, C., Sulser, T. B., Zhu, T., & Cline, S. A. (2008). *International model for policy analysis of agricultural commodities and trade (IMPACT): Model description*: International Food Policy Research Institute Washington, DC.

Rossel, R. V., Walvoort, D., McBratney, A., Janik, L. J., & Skjemstad, J. (2006). Visible, near infrared, mid infrared or combined diffuse reflectance spectroscopy for simultaneous assessment of various soil properties. *Geoderma, 131*(1-2), 59-75.

Rudd, J. D., Roberson, G. T., & Classen, J. J. (2017). *Application of satellite, unmanned aircraft system, and ground-based sensor data for precision agriculture: a review*. Paper presented at the 2017 ASABE Annual International Meeting.

Rumpf, T., Mahlein, A.-K., Steiner, U., Oerke, E.-C., Dehne, H.-W., & Plümer, L. (2010). Early detection and classification of plant diseases with Support Vector

Machines based on hyperspectral reflectance. *Computers and electronics in agriculture, 74*(1), 91-99.

Serpico, S. B., & Bruzzone, L. (2001). A new search algorithm for feature selection in hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing, 39*(7), 1360-1367.

Serpico, S. B., & Moser, G. (2007). Extraction of spectral channels from hyperspectral images for classification purposes. *IEEE Transactions on Geoscience and Remote Sensing, 45*(2), 484-495.

Shanahan, J. F., Schepers, J. S., Francis, D. D., Varvel, G. E., Wilhelm, W. W., Tringe, J. M., . . . Major, D. J. (2001). Use of remote-sensing imagery to estimate corn grain yield. *Agronomy Journal, 93*(3), 583-589.

Solari, F., Shanahan, J., Ferguson, R., Schepers, J., & Gitelson, A. (2008). Active sensor reflectance measurements of corn nitrogen status and yield potential. *Agronomy Journal, 100*(3), 571-579.

Svensgaard, J., Roitsch, T., & Christensen, S. (2014). Development of a mobile multispectral imaging platform for precise field phenotyping. *Agronomy, 4*(3), 322-336.

Thenkabail, P., Gumma, M., Teluguntla, P., & Mohammed, I. (2014). Hyperspectral Remote Sensing of Vegetation and Agricultural Crops. *Photogrammetric Engineering & Remote Sensing (PE&RS), 80*(8), 697-723.

Trombetti, M., Riaño, D., Rubio, M., Cheng, Y., & Ustin, S. (2008). Multi-temporal vegetation canopy water content retrieval and interpretation using artificial neural networks for the continental USA. *Remote sensing of Environment, 112*(1), 203-215.

Tsouvaltsidis, C., Al Salem, N. Z., Benari, G., Vrekalic, D., & Quine, B. (2015). Remote Spectral Imaging Using a Low Cost UAV System. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40*(1), 25. doi:10.5194/isprsarchives-xl-1-w4-25-2015

USDA. (2015). Irrigation & Water Use.

Verger, A., Baret, F., & Weiss, M. (2008). Performances of neural networks for deriving LAI estimates from existing CYCLOPES and MODIS products. *Remote sensing of Environment, 112*(6), 2789-2803.

Von Bueren, S., Burkart, A., Hueni, A., Rascher, U., Tuohy, M., & Yule, I. (2015). Deploying four optical UAV-based sensors over grassland: challenges and limitations. *Biogeosciences, 12*(1), 163-175.

Walthall, C., Dulaney, W., Anderson, M., Norman, J., Fang, H., & Liang, S. (2004). A comparison of empirical and neural network approaches for estimating corn and soybean leaf area index from Landsat ETM+ imagery. *Remote sensing of Environment, 92*(4), 465-474.

Wu, W., Wang, X., Xie, D., & Liu, H. (2007). *Soil water content forecasting by support vector machine in purple hilly region.* Paper presented at the International Conference on Computer and Computing Technologies in Agriculture.

Xiang, H., & Tian, L. (2011). Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosystems engineering, 108*(2), 174-190. doi:10.1016/j.biosystemseng.2010.11.010

Yari, A., Madramootoo, C. A., Woods, S. A., & Adamchuk, V. I. (2017). Performance evaluation of constant versus variable rate irrigation. *Irrigation and drainage, 66*(4), 501-509. doi:10.1002/ird.2131

Ye, H., Li, W., & Abedini, A. (2017). An improved heuristic for no-wait flow shop to minimize makespan. *Journal of Manufacturing Systems, 44*, 273-279.

Ye, H., Li, W., Abedini, A., & Nault, B. (2017). An effective and efficient heuristic for no-wait flow shop production to minimize total completion time. *Computers & Industrial Engineering, 108*, 57-69.

Yule, I., Hedley, C., & Bradbury, S. (2008). *Variable-rate irrigation.* Paper presented at the 12th Annual Symposium on Precision Agriculture Research & Application in Australasia. Sydney.

Zeng, C., King, D. J., Richardson, M., & Shan, B. (2017). Fusion of Multispectral Imagery and Spectrometer Data in UAV Remote Sensing. *Remote Sensing, 9*(7), 696.

Zhang, C., & Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: a review. *Precision agriculture, 13*(6), 693-712.

VITA

Ali Hamidisepehr

EDUCATION

Doctor of Philosophy, Biosystems and Agricultural Engineering, University of Kentucky, 2015-2018. Dissertation: Classifying Soil Moisture using Reflectance-Based Remote Sensing. Advisor: Michael P. Sama, Ph.D., P.E. GPA: 4
Master of Science, Agricultural Engineering, University of Tehran, Iran, 2012-2014. Thesis: Design and Development of a Hot Bed Spray Dryer. Advisor: Gholamreza Chegini, Ph.D. GPA: 18.40/20 (Rank 1st)
Bachelor of Science, Agricultural Engineering, University of Tehran, Iran 2008-2012. Thesis: Wind Energy Applications in Agriculture. Advisor: Seyed Reza HasanBeigi, Ph.D. GPA: 17.16/20 (Rank 1st)

PUBLICATIONS

1. **Ali Hamidiesepehr,** Michael P. Sama. 2018. A Low-cost Method for Collecting Hyperspectral Measurements from a Small Unmanned Aircraft System. In Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping III (Vol. 10664, p. 106640H). International Society for Optics and Photonics.

2. **Ali Hamidiesepehr,** Michael P. Sama, Ole O. Wendroth, Aaron P. Turner. 2017. A Method for Reflectance Index Wavelength Selection from Moisture Controlled Soil and Crop Residue Samples. Transactions of the ASABE, 60(5), 1479-1487.

3. **Ali Hamidiesepehr,** Michael P. Sama. 2017. Moisture Content Classification of Soil and Stalk Residue Samples from Spectral Data using Machine Learning. Transactions of the ASABE (In Press).

4. **Ali Hamidiesepehr,** Michael P. Sama. 2018. Instrumenting Low-Cost Spectral Remote Sensing aboard a Small Unmanned Aircraft System and a Method for Ambient Light Compensation. Journal of Applied Remote Sensing (In Press).

5. Dwayne R. Edwards, Somsubhra Chattopadhyay, Yao Yu, **Ali Hamidisepehr**. 2017. An Assessment of Climate Change Impacts on Future Water Availability and Droughts in the Kentucky River Basin. Environmental Processes of Springer, 4(3), 477-507.

6. Mohsen Fakher Dizaji, **Ali HamidiSepehr**, Gholamreza Chegini, Javad Khazaei, & Ali Mansuri. 2015. Influence of Hot Bed Spray Dryer Parameters on Physical

Properties of Peppermint (Mentha piperita L.) Tea Powder. International Journal of Food Engineering, 11(1), 115-125.

7. Gholamreza Chegini, **Ali HamidiSepehr**, Mohsen Fakher Dizaji, and Seyed Vahid Mirnezami. 2014. Study of Physical and Chemical Properties of Spray Drying Whey Powder. International Journal of Recycling of Organic Waste in Agriculture 3, no. 2: 1-7.
8. Gholamreza Chegini, **Ali Hamidisepehr**, Mohsen Fakher Dizaji, Behzad Bashiri. 2013. Measurement of Pumped-air by Spray Dryer Atomizer. Journal of Agricultural Engineering Research 14, no. 3 (2013): 29-36 (in Persian).

PRESENTATIONS

1. **Ali Hamidisepehr**, Michael P. Sama 2018. Classifying Reflectance Targets from Hyperspectral Data Collected under Ambient Light Conditions using a Passive Low-Cost Remote Sensing System. ASABE Annual International Meeting. Detroit, MI.
2. **Ali Hamidisepehr,** Michael P. Sama 2018. A Low-cost Method for Collecting Hyperspectral Measurements from a Small Unmanned Aircraft System. SPIE Commercial + Scientific Sensing and Imaging. Orlando, FL.
3. **Ali Hamidisepehr,** Michael P. Sama 2018. A Low-Cost Method for Collecting Hyperspectral Measurements and Using Eclipse Data for System Calibration. ASABE Agricultural Equipment Technology Conference. Louisville, KY.
4. **Ali Hamidisepehr**, Michael P. Sama 2017. A Comparison between Reflectance Index and Machine Learning Algorithms for Moisture Estimation from Spectral Data. ASABE Annual International Meeting. Spokane, WA.
5. Christopher Good, Michael Sama, Saket Dasika, **Ali Hamidisepehr**, Ricky Mason 2017. A Low-Cost System for Instrumenting LiDAR on an Unmanned Aircraft System. ASABE Annual International Meeting. Spokane, WA.
6. **Ali Hamidisepehr,** Michael P. Sama 2017. A Method for Reflectance Index Wavelength Selection from Moisture Controlled Soil and Crop Residue Samples. ASABE Agricultural Equipment Technology Conference. Louisville, KY.

RESEARCH INTERESTS

- Spectroscopy
- Hyperspectral/Multispectral Analysis
- Machine Learning Applications
- Precision Agriculture and Automation

- Remote Sensing in Agriculture
- Spectral Analysis on Field Data
- UAS Applications
- Field-robotics and Control Systems

TECHNICAL SKILLS

- Programming (MATLAB, Python)
- Machine learning (MATLAB)
- Microcontroller programming and applications (Raspberry Pi and Danfoss Plus One)
- 3D modelling (SolidWorks, CATIA, Inventor, AutoCAD)
- Automation (PLC S7 Siemens and mini PLC LOGO!)
- Experience in control systems
- Remote sensing and spectral analysis
- Unmanned aircraft system (UAS) applications in agriculture
- Instrumentation for engineering applications
- Network management by Microsoft products (Windows Server and corresponding devices)

PROFESSIONAL EXPERIENCES

- **Graduate Research Assistant.** University of Kentucky. Department of Biosystems and Agricultural Engineering. Lexington, KY. Supervisor: Michael Sama. (2015-present)

- **Graduate Teaching Assistant.** University of Kentucky. Department of Biosystems and Agricultural Engineering. Lexington, KY. Course Title: “Bio Systems Modeling”. (Spring 2018)

- **Graduate Teaching Assistant.** University of Kentucky. Department of Biosystems and Agricultural Engineering. Lexington, KY. Course Title: “Control of Off-road Vehicles. (Spring 2017)

- **Graduate Student.** University of Kentucky. Collaboration Leading Operational UAS Development for Meteorology and Atmospheric Physics (CLOUD-MAP)- Concentrate on airborne soil hydrology, developing custom multispectral remote sensing instruments that observe moisture differences in crops and soils from a UAS platform, 2015-2018

- **Member.** University of Kentucky Wildcat Pulling Team- ASABE International ¼ Scale Tractor Student Design Competition, 2017 & 2018

- **Exhibitor**. University of Kentucky Engineering Day. Lexington, KY. 2018
- **Exhibitor**. South-eastern ASABE Student Rally Conference Tours at University of Kentucky. Lexington, KY. 2018
- **Graduate Research Assistant.** University of Tehran. Department of Biosystems and Agricultural Engineering. Tehran, Iran. Supervisor: Gholamreza Chegini. (2012-2014)
- **Assistant IT Manager.** ADAB Educational Complex and ImenParto. Tehran, Iran. 2010-2013

PROFESSIONAL ORGANIZATION

- American Society of Agricultural and Biological Engineers (2016-Present)
- University of Kentucky Wildcat Pulling Team, 2017-present

HONORS AND AWARDS

- Outstanding student award in MSc
- Outstanding student award in BSc
- Recognized as an exceptional talent by National Assessment Organization, Iran

PROFESSIONAL MEETINGS ATTENDED

- ASABE Annual International Meeting. Spokane, WA. 2017
- ASABE Agricultural Equipment Technology Conference. Louisville, KY. 2017
- ASABE Machinery Systems- Soil-Plant-Machine Dynamics Committee, Spokane, WA. 2017
- ASABE Machinery Systems- Unmanned Aerial Systems Committee, Spokane, WA. 2017

CERTIFICATES

- Machine Learning (Coursera)
- Control of Mobile Robots (Coursera)
- Mechatronic (Technical and Vocational Training Organization, Tehran, Iran)
- MCSE-MCITP (Tehran Institute of Technology, Tehran, Iran)