USEFUL AI APPLICATIONS IN AGRICULTURE: AGGREGATION OF MACHINE LEARNING TECHNIQUES FOR WEATHER FORECASTING AND BANANA PLANT COUNTING

BY

MR. BIPUL NEUPANE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE (ENGINEERING AND TECHNOLOGY)

SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2018

USEFUL AI APPLICATIONS IN AGRICULTURE: AGGREGATION OF MACHINE LEARNING TECHNIQUES FOR WEATHER FORECASTING AND BANANA PLANT COUNTING

BY

BIPUL NEUPANE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE (ENGINEERING AND TECHNOLOGY)

SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2018

THAMMASAT UNIVERSITY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

THESIS

BY

MR. BIPUL NEUPANE

ENTITLED

USEFUL AI APPLICATIONS IN AGRICULTURE: AGGREGATION OF MACHINE LEARNING TECHNIQUES FOR WEATHER FORECASTING AND BANANA PLANT COUNTING

was approved as partial fulfillment of the requirements for
the degree of Master of Science (Engineering and Technology)

on July 15, 2019

Chairman _____
(Preesan Rakwatin, Ph.D.)

Member and Advisor _____
(Assistant Professor Nguyen Duy Hung, Ph.D.)

Member and Co-advisor _____
(Assistant Professor Teerayut Horanont, Ph.D.)

Member _____
(Associate Professor Toshiaki Kondo, Ph.D.)

Director _____
(Professor Pruettha Nanakorn, D.Eng.)

Thesis Title	USEFUL AI APPLICATIONS IN AGRICULTURE: AGGREGATION OF MACHINE LEARNING TECHNIQUES FOR WEATHER FORECASTING AND BANANA PLANT COUNTING
Author	Mr. Bipul Neupane
Degree	Master of Science (Engineering and Technology)
Faculty/University	Sirindhorn International Institute of Technology, Thammasat University
Thesis Advisor	Assistant Professor Nguyen Duy Hung, Ph.D.
Thesis Co-Advisor	Assistant Professor Teerayut Horanont, Ph.D.
Academic Years	2018

ABSTRACT

Modern problems in agricultural seek methods to increase the convenience, reliability accuracy and at the same time, cost efficacy in precision farming. To exploit the global research gaps in sustainable agriculture, this thesis organizes three different works committed to provide solutions to three modern agricultural problems: reliable crop modeling, accurate crop counting and cost-effective crop health monitoring. As strategic solutions, this study proposes the use of emerging advancements on recent artificial intelligence technologies, weather forecasting, web-services, aerial imagery and digital image processing to connect the crucial cores of sustainable agriculture visions. The outputs of this work is targeted help decision makers, and specially the farmers themselves, to increase the productivity of farms for better yield, and to draw young people to the farms to meet the global hunger of 21st century with sustainability in life support and food security.

The unavailability of seasonal weather data at the simulation time (often beginning of a growth season) forces decision makers in crop growth assessment to consider multiple weather scenarios, which are often generated from longtime observed data. Ironically these scenario immediately become “outdated” as soon as the season begins, because they are always different from newly observed data. In the first research of this thesis, we investigate this dilemma and in particular address three questions: determination of most successful scenario, classification of scenarios into fresh and stale, and generation of a new scenario from fresh scenarios. Algorithms to solve these questions are given as strategies for prediction games of weather generators. We also elaborate on the applications of our results in networking existing weather generation web services.

The production of banana - one of the highly consumed fruits - is highly affected due to loss of certain number of banana plants in an early phase of vegetation. This affects the ability of farmers to forecast and estimate the production of banana. In the second research of this thesis, we propose a deep learning based algorithm for detection and counting of banana plants, using high resolution RGB aerial images collected from Unmanned Aerial Vehicle (UAV). An attempt to detect the plants on the normal RGB images resulted only 72.8% recall for our sample images of a commercial farm in Thailand. To improve this result, we use several image processing methods to enhance the vegetative properties - radiance, hue, saturation and values (HSV), and chlorophyll content of banana leaves - to generate multiple variants of aerial images. Then we separately train a parameter-optimized Convolutional Neural Network (CNN) on manually interpreted banana plant samples, to produce multiple results of detection. We apply the same algorithm on images collected from multiple flying altitudes, and merge the detection results to increase the recall to 97.6%.

Unmanned Aerial Vehicle (UAV) photogrammetry has allowed to monitor crop growth/health and remotely estimate biomass through calculation of Vegetation Indices (VI). However, some of the indices require costly sensors, and the process of generating VI maps from UAV images also requires commercial off-the-shelf software packages. The third research of this thesis uses existing open-source tools and methods to develop an

algorithm for a web-service that can create orthophotos, canopy height model and VI's for red-green-blue (RGB) images collected from UAV. We also discuss on ways to balance between the processing speed and quality of outputs, and further compare them to the outputs of an existing state-of-the-art commercial service.

Keywords: Sustainable Agriculture, Weather Scenario Generation Game, Prediction Games, Deep-Learning, UAV Photogrammetry, Individual Crop Detection, Crop Counting, Vegetation Indices, UAV Image Processing, Crop Growth Monitoring, Web-service

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor Asst. Prof. Dr. Nguyen Duy Hung and my co-advisor Asst. Prof. Dr. Teerayut Horanont (School of Information, Computer, and Communication Technology (ICT), SIIT, Thammasat University), whose constant motivation and academic advice tremendously furnished my research work and as a result helped me accomplish my study. It is a great honor to work under both of their supervision.

I would also like to express my sincere appreciation to my committee members Assoc. Prof. Dr. Toshiaki Kondo (School of Information, Computer, and Communication Technology (ICT), SIIT, Thammasat University), and Dr. Preesan Rakwatin (Vice President, Digital Agriculture Development and Promotion Department, Digital Economy Promotion Agency, Thailand) for their endless support and constructive comments throughout the research work.

I am deeply grateful to Asst. Prof. Dr. Duc Hoang Nguyen (Asian Institute of Technology, Thailand), whose expertise has motivated to focus on my research problems in my early stage of research. And I would also like to thank the Ms. Naratsita Trirathanun (Secretary, School of ICT, SIIT), Mr. Witchapon Jomprapan (Secretary, School of ICT, SIIT) and all the staff of SIIT and Thammasat University who helped me directly and indirectly on my stay in SIIT during my studies.

I would also like to thank Advance Geospatial Technology Research Unit, SIIT for technical and software support; and TU Basic and Applied Research Grant 2018 for their generous financial support.

I owe my deepest gratitude to my parents, Mr. Dayaram Neupane and Mrs. Sumitra Neupane, and my dear friend Mr. Suman Ghimire, for their continuous encouragement and moral support to complete this report as well as my study at SIIT, Thammasat University, Thailand.

Mr. Bipul Neupane

TABLE OF CONTENTS

Page

ABSTRACT (2)

ACKKNOWLEDGEMENTS (5)

LIST OF TABLES (9)

LIST OF FIGURES (10)

LIST OF SYMBOLS/ABBREVIATIONS (11)

CHAPTER 1 INTRODUCTION 14

1.1Background 14

1.1.1 Crop Modelling: Classification of weather predictions 15

1.1.2 Crop counting: Deep-learning and Image processing 16

1.1.3 Crop health monitoring: Open-source UAV Image Processing 19

1.2 Problem Statement 21

1.3 Objectives 22

1.4 Significance of Work 23

CHAPTER 2 LITERATURE REVIEW 26

2.1 Weather Scenario Generation Game 26

2.1.1 Prediction Game 28

2.1.1.1 Simple Meta-Inductivist 29

2.1.1.2 Weighted-Average Meta-Inductivist 31

2.2.1.3 Regret-Weighted Meta-Inductivist 32

2.2 Deep Learning and UAV-collected RGB Images for Agriculture 33

2.3 Open-source UAV Image Processing for Crop Health Monitoring 38

CHAPTER 3 MATERIALS AND METHODS 42
3.1 Weather Scenario Generation Game 42
3.1.1 Obtain Weather Scenarios 42
3.1.2 Determination of the most successful scenario 44
3.1.3 Classification of scenarios into fresh and stale scenarios 45
3.1.3.1 Binary Game for RAIN 46
3.1.3.2 Real-value Game for other variables 47
3.1.4 Generation of new scenarios from fresh scenarios 48
3.2 Deep Learning based Banana Plant Detection and Counting using UAV collected RGB Images 49
3.2.1 Dataset 49
3.2.2. Workflow 50
3.2.2.1 Orthophoto Generation 52
3.2.2.2 Image Processing 53
3.2.2.3 Image Preparation and Tiling 56
3.2.2.4 Train images using parameter-optimized CNN model 58
3.2.2.5 Production of multiple detection results 60
3.2.2.6 Merge the detection results 61
3.3 Open-source UAV Image Processing for Crop Health Monitoring 62
3.3.1 Generation of an Orthophoto 63
3.3.2 Generation of Canopy Height Model 63
3.3.3 Generation of Vegetation Indices 64
3.3.4 Fine-tuning image processing speed and quality 66

CHAPTER 4 EVALUATION AND DISCUSSION 69
4.1 Weather Scenario Generation Game 69
4.1.1 Binary Game on RAIN 69
4.1.2 Real-value Game 72

4.2 Deep Learning based Banana Plant Detection and Counting using UAV collected RGB Images 74
4.2.1 Performance of image processing methods 76
4.2.2 Performance in varying flight altitude of UAV 77
4.2.3 Combining the results of multiple flight altitudes 78
4.3 Open-source UAV Image Processing for Crop Health Monitoring 80

CHAPTER 5 CONCLUSION AND FUTURE PROSPECTS 82

REFERENCES 84

BIOGRAPHY 97

LIST OF TABLES

Tables Page

3.1 Sample of Historical Weather Data from 1979 to 2015 43

3.2 Run-time parameters of ODM under fine-tune options 67

4.1 Success rate of bMI on general and sub-game of binary RAIN data. 71

4.2 Success rate of RW on general and sub-game of real-value TAVG data 73

4.3 Detection performance of Faster-RCNN model on variants of ROI taken from 40m flight altitude 76

4.4 Detection performance of Faster-RCNN model on variants of ROI taken from 50m flight altitude. 78

4.5 Detection performance of Faster-RCNN model on variants of ROI taken from 60m flight altitude. 78

4.6 Detection performance of our algorithm after combining results from 40, 50 and 60m altitude. 79

4.7 Histogram comparison of orthophoto obtained from Low, Medium, High and Pix4D 81

LIST OF FIGURES

Figures	Page
2.1 General workflow for UAV image processing	40
3.1 Study area for DL based banana plant detection and counting	50
3.2 Proposed Algorithm for DL based banana plant detection and counting	51
3.3 Workflow for DL based banana plant detection and counting	52
3.4 Histogram of (a) Original RGB orthophoto and (b) Contrast stretched orthophoto	54
3.5 Orthophoto of farm after (a) LCS (b) SCT and (c) TGI	56
3.6 Study area separated into training region and ROI (top right).	57
3.7 Sample image tile of (a) LCS (b) SCT and (c) TGI	58
3.8 Detection results on variants of image processing methods on a sample image tile of (a) LCS (b) SCT and (c) TGI	61
3.9 Merging detection results from variants of image processing methods.	62
3.10 (a) Orthophoto, (b) DSM, (c) DTM and (d) CHM of the study area.	64
3.11 (a) VARI, (b) TGI and (c) NGRDI of study area.	66
4.1 The bMI's optimal scenario on binary RAIN data of DISAWGS	70
4.2 The bMI's optimal scenario (dotted line) on binary RAIN data with an assumption of a player B (dashed line) with winning time of n_B=20.	70
4.3 General game of RW on all scenarios of real-value TAVG data	72
4.4 Sub-game of RW on fresh scenarios of real-value TAVG data	73
4.5 Final detection on ROI image taken from 40m flight altitude.	75
3.6 Final detection on ROI after combining results of 40, 50 and 60m altitude variants.	80
3.7 Closer view of orthophoto generated by (a) Low, (b) Medium (c) High and (d) Pix4Dmapper for banana farm.	81

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
AI	Artificial Intelligence
UAV	Unmanned Aerial Vehicles
CNN	Convolutional Neural Networks
DL	Deep Learning
GNDVI	green normalized vegetation index
LAI	Leaf Area Index
RGB	Red Green Blue
NIR	Near infra-red
CMOS	Complementary Metal-oxide Semiconductor
CCD	Charge-coupled Device
WSG	Weather Scenario Generation Game
BBTV	Banana Bunchy Top Virus
PET	Potential Evapotranspiration
FAO	Food and Agriculture Organization
WG	Weather Generator
DSSAT	Decision Support System for Agrotechnology Transfer
MA	Moving Average
AR	Auto-regressive model
ARIMA	Auto-regressive Integrated Moving Average
k-NN	k-Nearest Neighbor
MI	Meta-inductivist
OI	Object Inductivist
bMI	Imitate-the-best Meta-inductivist
wMI	Weighted Meta-inductivist
RW	Regret-weighted Meta-inductivist
SfM / SFM	Structure from Motion

GCP	Ground Control Point
DTM	Digital Terrain Model
DEM	Digital Elevation Model
DSM	Digital Surface Model
CHM	Canopy Height Model
CSM	Crop Surface Model
3D	3 Dimension
MS COCO	Microsoft COCO: Common Objects in Context
VOC	Visual Object Classes
YOLO	You Only Look Once
RCNN	Regional Convolutional Neural Networks
mAP	Mean Average Precision
SSD	Single Shot Multibox Detector
ODM	Open Drone Map
API	Application Program Interface
DISAGWS	DisAg Weather Generator Web Service
TMIN	Minimum Temperature
TMAX	Maximum Temperature
RAIN	Rainfall
SRAD	Solar Radiation
TAVG	Average Temperature
ROI	Region of Interest
DN	Digital Numbers
GL	Grey Levels
LCS	Linear Contrast Stretch
SCT	Synthetic Color Transform
HSV	Hue, Saturation and Value
VI	Vegetation Index
TGI	Triangular Greenness Index

GT	Ground Truth
GPU	Graphic Processing Unit
JPEG	Joint Photographic Experts Group
RPN	Region Proposal Network
TP	True Positives
FP	False Positives
FN	False Negatives
IoU	Intersection over Union
GDAL	Geospatial Data Abstraction Library
NDVI	Normalized Difference Vegetation Index
VARI	Visible Atmospherically Resistant Index
NGRDI	Normalized Green Red Difference Index

CHAPTER 1
INTRODUCTION

1.1 Background

Rapidly growing population of perhaps 9 billion of this century faces dual challenge of food security and sustainable life supporting system (Council & others, 1999). Delimitation of these problems require innovative advancement over agricultural, environmental and climatic information. One of the golden achievement of climatic researches is the ability to forecast into years ahead of estimations (Weber & Stern, 2011) which has given the decision makers in agricultural and environmental sectors an immense opportunity to minimize unnecessary impacts and to benefit from favorable conditions. Without doubt, agricultural management, and food and livelihood security have been blessed with the prior and sufficient lead time to take action provided by advancement over information to adjust the criticality in agriculture.

Modern problems require modern solutions. And the advancements of artificial intelligence (AI) in almost every possible fields have also blessed the sector of agricultural management and precision agriculture. Countries build their current multi-year plans and economic models include precision agriculture. For example, Thailand 4.0 (C. Jones & Pimdee, 2018) has come up with strategies to change the traditional way of farming towards smart farming in Thailand. Across the globe, the concept of smart farming are being graciously welcomed because farmers wants to change the tedious way of labor, miscalculations, errors and loss due to conventional approaches in farming. This is why, I think widely accepted AI technologies like machine learning (Shavlik, Dietterich, & Dietterich, 1990), weather forecasts, web-services, aerial imagery and digital image processing need to be brought together for smart farming. This thesis classifies current problems of smart farming and their solutions into three categories as below.

1.1.1 Crop Modelling: Classification of weather predictions

Adaptive and sustainable responses to climate require reliable and salient information for decision support. Seasonal climatic forecasts can profit farmers only the information provided are reliable and are yield-oriented to the scale that impacts their decision (Baethgen, Carriquiry, & Ropelewski, 2009). Farmers are interested more into the data that can increase their economy by increasing productions but not into the decision support on rather forecasting a seasonal precipitation pattern. Forecasting the years ahead benefits the decision makers in higher level like the departments and agency whose job is to see the pattern in yield and economic benefits. The gap between different type of decision makers like regional level climatic out-lookers, prediction centers, multiple online web-services, and a farmer should be understood and their demands should be matched by fulfilling the gap in timely reliable climatic predictions (Hansen, Challinor, Ines, Wheeler, & Moron, 2006). Also the agricultural management and risk assessment requires great deal of converting raw information of climate and weather into distributable relevant outcomes.

Stochastic weather generators are the provider of daily weather scenarios from the historical data which can be simulated over time series. These series of data share the statistical similarity of properties with those observed in the past history or to the other related scenario affecting it. (Wallis & Griffiths, 1995; Wilks & Wilby, 1999) have given some reviews on the use of common weather generators. The studies regarding change in climates have benefited and have been impacted resourcefully by the use of stochastic weather generators (Dubrovsk\`y, Žalud, & Št'astná, 2000; Mearns, Rosenzweig, & Goldberg, 1997; Semenov & Barrow, 1997; Trnka, Dubrovsk\`y, Semerádová, & Žalud, 2004; Tubiello, Donatelli, Rosenzweig, & Stockle, 2000). These series of weather are the key inputs of crop growth model simulations and also the climate impact assessments are subsequently aided by the comparison of changed climates and the results provided by these weather series. Applications of weather generators are but not limited to climate and agriculture.

The weather generators produces a series of meteorological weather parameters including precipitation, temperature, sunshine, wind, humidity, as well as derivation of

potential evapotranspiration. Also the futuristic scenario are generated by the application of different regression models fitted to the historical observations. These weather generators are available both as web services and program software. The user community including the farmers are prone to the use of number of weather available weather scenarios provided by these generators. The guarantee of the weather scenario cannot be given due to the uncertainty involved in weather also aided by climate change. A mathematical statistical model needs to be formulated to classify these weather generators based on the reliability. The application of the reliable most scenarios on crop yield models is the major point interest in this study. An optimal prediction strategy could help him choose the best data forecast available.

The availability of weather scenario throughout a time series is significant in many fields. Online data is the source of information to any human these days. With the availability of mass data from the pool of internet, comes the dilemma of choosing the correct weather scenario. To minimize the dilemma, a solution can be derived from past-induced learning of data, to help choose the optimal predictions among the available network of online weather forecast. Talking about learning from the past data, meta-inductive prediction methods have been introduced in (Schurz, 2008) that uses expert's advices that provides foundations to the theory of prediction of individual sequences. This inductive inference has led to prediction game that is able to provide an optimal prediction induced from the predictions of other player's predictions that were laid based on historical dataset. This concept of prediction game is sought to be applied in this thesis, to the problem of obtaining an optimal weather forecast series for multiple period of time, which will further help online data users to classify better weather data from the outdated dataset. These series of weather forecasts are further aimed in improving crop simulation models and yield estimation.

1.1.2 Crop counting: Deep-learning and Image processing

Apart from using machine learning algorithm like prediction game in agricultural applications for improvement of crop simulation models, this thesis work dives further into

Deep Learning, which is taken as a family of machine learning; for agriculture using remotely collected images. Precision agriculture have been substantially blessed by remote sensing applications in last three decades (Bastiaanssen, Molden, & Makin, 2000; Mulla, 2013). However, most of the studies are focused on crop monitoring. Much effort have been made to calculate parameters like vegetation indices, crop height, crop yield, leaf area index, surface soil properties, ground biomass, water stress, canopy height models, and much more (C. Zhang & Kovacs, 2012). The count of individual crops does not yet make sense to research world, unless they are not large enough and well-spaced during plantation. While cultivating, banana plants need about 1.5 meters of spacing between each of them (STEWART et al., 1998). In fact, this plant is a tree-like herbaceous plant (D. W. Turner, 1972); tree-like because of its size being the largest of its species, and almost as big as a tree. This is why, the count of vegetative banana plants can be more related to the research area based on forest inventory management, rather than limiting to precision agriculture.

Remote Sensing technologies like satellite and airborne sensors developed rapidly since 1950s (Ke & Quackenbush, 2011). Visual interpretation of aerial images replaced field measurement for forest management since early 1960s (Singh, Sohlberg, Sokolov, & others, 1986). In 1980s, algorithms for digital imagery based automation in tree detection and delineation were demonstrated by (Pinz, 1998), where the author used local maxima of brightness in smoothed aerial images to locate the center of a tree crown. Algorithms like valley-following and rule-based algorithm (Gougeon, Leckie, & others, 1998), multiple scale analysis (Brandtberg, 1999) and model-based template matching techniques (Pollock, 1998) were used in the 90's to detect coniferous tree crowns. Much effort have been made to delineate tree crowns in a forest, and similarity in those research is that the trees have a point of maximum that allows those algorithms to fit for purpose. Peculiarly, the leaves of a banana plant are spread from its trunk, and can have multiple point of maxima. Thus, the afore-mentioned traditional methods do not fulfill our purpose.

In 2014, object-based image analysis was used to delineate potential banana plantation area in high resolution satellite images (Johansen et al., 2014), to contribute to

the detection of BBTV virus in Queensland, Australia. However, the objective of the research was limited not to detect individual plants, but to detect banana cultivated areas on satellite images. In contrast to the past studies, for our purpose, we use Unmanned Aerial Vehicles (UAV) for robustness in collection of aerial images from multiple altitudes. UAVs provide higher temporal and spatial resolution compared to the satellite images, and also increases cost-effectiveness depending upon the study area. Some of the recent works in individual tree detection using UAV can be found in (Mohan et al., 2017).

The shape of a banana plant on aerial images is mostly generalized by shape of a star. But in reality, it can be seen on aerial images that the shapes are irregular and uncommon from each other. Even though the detection of individual banana plants have not been performed to the best of our knowledge, star-shaped tree - palm trees - have been detected over high resolution satellite images in (Srestasathiern & Rakwatin, 2014), where they use local peak detection method to find the maximal point of the palm tree. As mentioned before, bananas plats do not always have a single peak point. This makes us consider feature matching on images, using convolutional neural networks (CNN) (Krizhevsky, Sutskever, & Hinton, 2012), as a solution to our problem of detecting and counting banana plants.

Deep Learning (DL) (LeCun, Bengio, & Hinton, 2015; LeCun, Bengio, & others, 1995) architectures have excelled the computational power of machine learning to higher degree of precision and performance by increasing the number of ``layers'' or ``depths''. DL allows fast and automatic feature extraction from adequately large dataset, iteratively using complex models to reduce classification errors in regression (Pan, Yang, & others, 2010). In recent years, DL have become core method in many researches in plant recognition (Reyes, Caicedo, & Camargo, 2015), plant disease recognition (Amara, Bouaziz, Algergawy, & others, 2017; Mohanty, Hughes, & Salathé, 2016), weed detection (Huang et al., 2018) and crop type classification (Kussul, Lavreniuk, Skakun, & Shelestov, 2017; Mortensen et al., 2016). A review on DL in agriculture can be found in (Kamilaris & Prenafeta-Boldú, 2018). However, most of the studies are carried out on non-aerial image dataset, taken in a controlled environment with proper lighting conditions. Others

use satellite images as their data source, and very few used UAV-collected images (Rebetez et al., 2016).

A DL based oil palm tree detection was carried out on satellite images by using CNN architecture in (Li, Fu, Yu, & Cracknell, 2016). Because of the complexity provided by irregularity in shape of banana plants, we apply CNN on the results given by different image processing methods, on the orthophoto of banana farm. Different image processing methods are available in regards to remote sensing applications (Jensen & Lulla, 1987), and methods like contrast stretch (Yang, 2006), synthetic color transformation of hue-saturation-value (Daily, 1983) and calculating vegetation index (Bannari, Morin, Bonn, & Huete, 1995; Xue & Su, 2017) are quite significant for vegetative measures. We apply these methods to obtain multiple variants of input images. The results obtained from the application of CNN over multiple variants of image processing methods and image variants from multiple flight altitudes, are finally assembled and combined to obtain banana plant detection and count.

1.1.3 Crop health monitoring: Open-source UAV Image Processing

Remote sensing technologies has been providing timely and accurate information related to crop health and productivity; and has been widely used in agricultural health monitoring(Atzberger, 2013) at both regional and global scale since decades. One of the important characteristics for any agricultural health monitoring system is to timely disseminate information related to plant health, growth and yield to the farmers. With recent advancements in technology, light-weight airborne remote sensing(C. Zhang & Kovacs, 2012) equipped with multispectral sensors presents a unique advantage over traditional satellite borne images (Lamb & Brown, 2001) in terms of high spatial & temporal resolution with reduced effect of cloud cover during data acquisition. However, the associated costs with such multispectral sensors are generally high which suggests researchers to focus on cheaper alternatives that could be financially implemented by farmers from developing countries.

Numerous studies has been performed in the application of UAV based imagery in precision agriculture (Alexandridis et al., 2018; Allahyari, Mohammadzadeh, & Nastis, 2016; Shaw, Lark, Williams, Chadwick, & Jones, 2016). All these studies suggest in improving georeferencing, image mosaicking algorithms and more concrete and automated workflows, as well as directly involve farmers in different design phases for better data interpretation and to provide good services to the farmers. The remote sensing platforms including air and space borne suffers atmospheric scattering in the blue & green region, consequently it is recommended using larger wavelength such as red and NIR for the purpose of agricultural applications (Nijland et al., 2014). Therefore, an inexpensive digital camera setup, with internal infrared filter removed and replaced with a blue blocking filter allowed blue channel to record the NIR light with red recorded in its original channel (Hunt, Daughtry, Eitel, & Long, 2011; Zigadlo, Holden, Schrader, & Vogel, 2001), resulting in a promising technique for agricultural monitoring. In recent years, a lot of research have implemented such modification of standard RGB digital camera to near infrared for assessment of crop health (Hunt, Cavigelli, Daughtry, Mcmurtrey, & Walthall, 2005; Hunt et al., 2011; Rabatel, Gorretta, & Labbe, 2014), but very less has been explored in CMOS based action camera model (Ghazal, Al Khalil, & Hajjdiab, 2015; Wijitdechakul, Sasaki, Kiyoki, & Koopipat, 2016). For instance, (Hunt et al., 2011) studied the ability of CCD based digital color infrared photograph in crop monitoring using UAV. Their results in terms of green normalized vegetation index (GNDVI) was found to have a good correlation with leaf area index (LAI) suggesting their approach to be potential in providing accurate information related to crop health. Similarly, (Wijitdechakul et al., 2016) demonstrates a dual action camera model: one normal RGB camera while the other IR filter removed modified camera in UAV platform for real-time agricultural area management using SPA process. Their system was able to detect the healthy vs non-healthy plantations, and notify farmers about the unhealthy plantation area for improved decision making in agricultural practices.

All the improvements in the sensors are aimed to either improve the quality of work, or trying to reduce the cost of work. UAVs have already reduced the cost of remote sensing

in precision agriculture, but the cost of sensors and cost of post-processing software still keeps these technologies away from farmers, decision maker and students. We focus on reducing the cost of post-processing of UAV collected images for the images taken from UAV.

1.2 Problem Statement

Precision agriculture have been blessed with technological advancement over the years. Weather data collection through sensors, weather forecasts, crop simulation models and crop growth monitoring through remote sensing technology have made some crucial advancements in the past couple of decades. Weather forecasts, which are key input to crop simulation models, are easily available via internet and other web-services. And these predicted or forecasted scenarios most of the time do not match exactly to the real event. The ease in data comes therefore with a problem of separating “best” weather forecasts from “outdated” forecasts. This best weather scenario should have shown least difference from recent new observations, thus providing lowest regret rate among all the other available alternative weather scenarios. In this thesis, we create a Weather Scenario Generation Game (WSG) to address three problems:

1. Determination of most successful scenarios
2. Classification of scenarios into fresh and stale
3. Generation of new scenarios from fresh scenarios

On the other part of the thesis, we solely depend on remote sensing technologies. One of the daily consumed fruits - Banana (genus *Musa*) - has a long history of cultivation and impacts local and global trade. Regardless, this industry have been threatened several times by pests, and viruses like Banana Bunchy Top Virus (BBTV) (Dale, 1987). Diseases like Yellow Sigatoka, Leaf speckle and Cordana, affects the growth of banana plant leaves, and are more active in the hottest and wettest climate, during which the plant tend to grow the most (STEWART et al., 1998). Thailand being located in a tropical area, is prone to such diseases in banana plants. Despite of following all standards of planting, banana farms in Thailand still face a problem of losing plants within first few months of cultivation. To

keep track of productivity, counting the number of plants needs to be automated, rather than tedious manual approach. One of the research from this thesis exploits this problem.

UAV (Gallington et al., 1997) is no more new to the research world and is continuously developing ever since it arrived (Colomina & Molina, 2014; Eisenbeiß, 2009). With extensive applications in various fields (Nex & Remondino, 2014) including agriculture, UAVs have been a blessing to photogrammetry and remote sensing works (Everaerts, 2008; Honkavaara et al., 2013; C. Zhang & Kovacs, 2012). The images collected from UAV require post-processing to stitch small images together, for which, different commercial and closed-source tools and services are available (Sona, Pinto, Pagliari, Passoni, & Gini, 2014). These services are still unable to extend the reach of precision agriculture towards many farmers and students, because of their costs. They need a cost-effective solution for crop health/growth monitoring, using inexpensive sensors in UAV and open-source algorithms for image processing. We develop an algorithm for open-source UAV image processing web-service to monitor crop health and estimate biomass, using some of the existing open-source tools and algorithms. At the same time, we also want to provide users with the power to control the processing speed and quality of outputs they want, depending on their urgency and need. We breakdown our problem of building an algorithm for open-source UAV image processing web-service into four sub-problems:

1. Generation of an Orthophoto
2. Generation of Canopy Height Model
3. Generation of Vegetation Indices
4. Fine-tuning image processing speed and quality

1.3 Objectives

Based on the problem statement, the objective of this study are:

1. To prepare develop a Weather Scenario Generation Game to provide theoretical framework for the determination of most successful scenario; classification of scenarios into fresh and stale; and generation of new scenarios from fresh scenarios.

2. To automate the count of banana plants during early stage of growth, in which they are prone to lose maturity.
3. To create an open-source UAV image processing web-service for crop health monitoring.

1.4 Significance of Work

The significance of this thesis work is not limited to a single, but multiple significant research areas. The three objectives mentioned before have their own specific advantages in their own research world. The first objective (Objective 1) derives an architecture to a game, to answer the three problems stated in regards to obtain "best" weather scenarios that are intended to increase the reliability of crop simulation, when used by the crop simulation models. The fundamental backbone to the game is provided by prediction game theory of (Schurz, 2008) that have not yet been applied to the field of weather generation. The game can be directly compared to ensemble-based classification (Rokach, 2010) that generally uses same learning algorithm to construct multiple number of classifiers, and later uses a combining model such as majority voting rule (Perrone & Cooper, 1995). Comparatively, our game uses the learning experiences of other classifiers, and then uses meta-strategies to combine their outputs and finally improve through experience.

Our strategic approach to obtain precise automation in counting banana plants (Objective 2) is to first collect multiple high resolution red-green-blue (RGB) aerial images of an affected banana farm, using UAV, from multiple flight altitudes. These images are then processed to obtain orthophotos that provides complete picture of the farm. The idea thereafter is to use deep learning (DL) architecture to learn the shape of banana plants on small tiles of orthophoto. As the RGB images alone cannot provide high level of accuracy in detection of banana plants, we combine detection results obtained from application of DL on (i.) multiple image processing methods, based on their vegetative properties and (ii.) images taken from multiple flight altitudes from UAV. Therefore, our research contribution

will be to evaluate the detection performance of each image variant produced from multiple image processing methods, and each variant produced from multiple flight altitudes.

An upper hand of our approach is the use of normal RGB camera, rather than some sophisticated multi-spectral sensors mounted in UAV, which belittles the expense of collecting aerial images from multiple altitudes. Also compared to satellite photogrammetry, UAV's provide cheaper but high resolution images in large scale, which directly affects performance of DL. Further, exploitation of vegetative properties of images collected from RGB sensors, in DL, will increase their potential in detection of banana plants. This could make visible band spectrum comparable to near-infrared (NIR), which are generally used to understand the vegetative indices on images. Another noteworthiness of this work is the combination of capabilities of multiple DL results on multiple image variants, also based on altitude. Furthermore, the automation in counting banana plants, which is the main objective of our work, allows to estimate loss, re-calculate production, and to set a contingency plan for re-plantation in affected areas.

For our third objective (Objective 3), the first problem is to stitch the images collected from UAV together, to produce an orthophoto, which is further used to generate Digital Elevation Models (DEM) (W. Zhang & Montgomery, 1994), that describes the surface of study area. The orthophoto and DEM are generated using an existing open-source Application Programming Interface (API) called Open Drone Mapping (ODM). The outputs from ODM is used to solve our second and third problem of generating Canopy Height Model (CHM) and Vegetation Indices (VIs). CHM is often used in crop modelling to monitor crop height, and vegetation indices help to understand crop health, growth and biomass estimation. As a solution to our fourth problem, we fine-tune some of ODM's run-time parameters to scale the work into desired quality depending on how fast user wants the results of image processing.

Extensively, we develop an open-source image processing web service, not only to obtain vegetation indices from UAVs with sophisticated cameras having infra-red sensors, but also for some simple UAV's that only have a normal cameras with RGB bands. These kinds of cameras are cheaper and are often used for photographic purpose. The outcome of

our work provides open-source service to study the crop growth, without the use of expensive sensors like infra-red sensors, further increasing the cost-effectiveness also during data collection.

CHAPTER 2
LITERATURE REVIEW

2.1 Weather Scenario Generation Game

Weather variables or parameters includes precipitation, mean temperature, minimum temperature, maximum temperature, wind speed, wind direction, solar radiation sunshine duration, cloud cover, and everything that impacts weather in any time instance. The parameters for the weather forecast are used with the accountability and scale of work. Food and Agriculture Organization (FAO) calculates Potential Evapotranspiration (PET) using a concept that requires Weather Generator (WG) variables; precipitation, mean temperature, temperature range and solar radiation to monitor crop growth. Crop modelling faces disability without weather scenarios. Climate impact assessments are incomplete without the weather observations and forecasts.

Commonly used crop modelling systems like DSSAT (Decision Support System for Agrotechnology Transfer) (J. W. Jones et al., 2003) uses weather generating methods like WGEN (Richardson & Wright, 1984) and SIMMETEO (Soltani & Hoogenboom, 2003) to generate daily weather data. SIMMETEO was more advantageous than WGEN because of its ability to estimate the parameters not from daily data which is the input for WGEN, but from the monthly summaries instead (Geng, de Vries, & Supit, 1986). Despite the log historical advancements, these parametric approaches suffered several limitations and drawbacks. DSSAT currently uses a weather generating utility software package called WeatherMan which implements WGEN and SIMMETEO methods and provides furthermore assistance of data cleaning during input to resolve the problems of format types, unit of data and missing data records. These generated weather scenario data need furthermore filtering into the usable and non-usable data to help users to get the most reliable simulations of crop yield.

Out of many approaches proposed to generate stochastic weather variables, (Wilks & Wilby, 1999) have grouped them into two major categories, namely – parametric and non-parametric approaches. The traditional weather generators that are now called

parametric weather generators uses rainfall as the base variable. Time series forecasting models such as Moving Average (MA) are used to generate maximum and minimum temperature data. There are more models such as Auto-regressive model (AR), Auto-regressive Integrated Moving Average (ARIMA), Neural Networking, and recently Machine Learning Algorithms (Hyndman & Athanasopoulos, 2018) recently in use for generating time series of such data.

The alternative and more attractive method to the parametric is non-parametric approach, which is data-centric and the assumptions on the scattering of variables of interest is not required. Based on the state of climate, simulations can easily be obtained from these approaches and the framework are even flexible to use (Wilks & Wilby, 1999). One of the method in this approach, the k-Nearest Neighbor (k-NN) bootstrap approach have been continuously modified over the time and have been in use regularly (Bannayan & Hoogenboom, 2008). This method developed by (Lall & Sharma, 1996) was later extended to multivariate data method from the delimitation of univariate time series resampling by (Rajagopalan & Lall, 1999). Later, the k-NN bootstrap weather generation was extended to multisite generation with successful results by (Buishand & Brandsma, 2001) and (Yates, Gangopadhyay, Rajagopalan, & Strzepek, 2003). Furthermore, this approach as adapted for conditional resampling on atmospheric indices and hydrological time series (Beersma & Buishand, 2003; Mehrotra, Srikanthan, & Sharma, 2006).

The weather generators use precipitation as their fundamental parameter to produce a series of meteorological weather parameters including temperature, sunshine, wind, humidity, as well as derivation of potential evapotranspiration. They use the regression relationships between the other parameters to rainfall, and the past values of the variables to produce weather scenarios (Kilsby et al., 2007; Semenov & Barrow, 1997). The future scenario are generated by the application of different regression models fitted to the historical observations. We use prediction game to learn the success and regret of multiple such weather scenarios that are generated by weather generators, for our purpose.

In this thesis, we present solutions to the problems of our first objective as strategies of meta-players in prediction games, in which weather scenarios are *object-level players*.

In the literature, prediction games have been first formalized by Schurz in (Schurz, 2008) and applied in different fields such as in (Schurz, 2012; Schurz & Thorn, 2017; Thorn & Schurz, 2012). However to the best of our knowledge, prediction games have not been applied in the field of weather generation. In theoretic prediction games, meta-player's strategy is to obtain an optimal prediction among a finite number of accessible object-level players who lay their prediction based on historical data. For our application, we have developed so-called *Weather Scenario Generation Game* (abbr. WSG) providing a theoretical framework for addressing the above three problems. To demonstrate our formal results and also shed lights to their applications, we feed the scenarios provided by an existing weather generation web service to our game.

2.1.1 Prediction Game

Schurtz have narrowed down the perspective of adversarial forecasting game (Schurz, 2008) in which he discusses about a player that can predict an optimal forecast among a finite number of players. Abbreviated as meta-inductivist (*MI*), the player's job is not to foresee the accurate prediction, but an optimal among the list of experts in contrast of their own past success and regret.

Let us consider a series of events $(e=e_1, e_2, \ldots, e_n)$ that may or may not be sequential. Say we have a series of stock price of a company for example and the events be the prices that actually occurred. And let us suppose that there are number of experts $(P_1, P_2, \ldots, P_n)$ who predict $p=\{p_1, p_2, \ldots, p_n\}$ prediction of price of the stock in the upcoming event. These experts make a prediction at time $n-1$ for time n.

- The players or experts are of three types according (Cesa-Bianchi & Lugosi, 2006):
- Object Inductivist OI:= P1, who has the access to the past events. They can give a guess to the next value or they may use some method of prediction.
- Alternative Players (P2, P3, …) who may also predict a Clairvoyant number and may or may not have a success. They may also have the information on any information MI wants and may know the MI's favourite players. The players (1) and (2) are termed as non-MI-players.

- Meta-Inductivists (xMI), x being a variable that can specify the type of MI player. These players have access to the past events, game that non-MI-players have played so far and beyond.

The predictions made by the players could be a set of binary values (for say the precipitation forecast with dry or wet being [0, 1]) or a set of real-value numbers (for say prediction of temperature). The OI can either make a random closest prediction or can pass the mean value resulting of a prediction method from the past events in case of a real value. But for the binary value prediction, passing a straight value merely applies (Schurz, 2008). In case of binary values, maximum frequency of event *[0, 1]* could be passed as the prediction.

In a standard prediction game, normalized *loss function* $l(p_n, e_n) := |p_n - e_n|$ is calculated at first which is the absolute difference between event and prediction, followed by the score $s(p_n(P), e_n) := 1 - l(p_n, e_n)$. A player's absolute success rate $a_n(P)$ is the players sum of score till time *n*. Player *P*'s relative frequency of correct predictions till time *n* is called *success rate* which is then calculated as $suc_n(P) := a_n(P)/n$. Now based on this success rate, *MI* can judge a best player among the list of alternative players. There are several strategic meta-inductivists explained below.

2.1.1.1 Simple Meta-Inductivist

Schurz starts his inquiry from a simple kind of meta-inductivist who predicts the prediction of an alternative player who has highest success rate. This MI basically chooses a favorite alternative player among the set or say a favorite player grabs MI's attention. MI could choose or assumes his best player with any strategies. It could be random selection or could be selection with the first alphabet of the name and its ascendance. We can say that this is a *bMI*, replacing the variable 'x' with 'b' to indicate the best MI player. The *bMI* after assuming the first favorite, switches to next favorite only when then other one has better success rate. In this way, *bMI* will be predicting what the best player predicts along the time to give an optimal forecast for time *n+1*. Several strategies can be formed to one's usage of game and severity to accuracy.

For a real valued game like degree of temperature, a strategy of taking mean or weighted mean can be used to predict future from past that is, $p_{n+1}(OI) = \overline{e}_{n..}$ But for the binary game, taking average gives a decimal number which is not usable since the binary game works with the logical value of 0 and 1. One could either search for a way to convert the real values to 0 and 1 or could use fathom a strategic rule of $p_{n+1}(OI) = 1$ if more number of players says the event will be *1*, and else = 0. Eventually, the loss function will converge to zero for the predictions that are non-random and follows some trend. But for the binary game on random data, success rate of OI converges against the loss's limiting mean value. OI's success rate dominates the success of players in case of non-random sequence of prediction.

A meta-inductive player runs by a meta-strategy which has access to the predictions of all the players in the game and the OI players contains no prediction information from other players. One can formulate any type of strategy according to application and degree of reliability. But for the meta-player to run, all the players in the game should be assessable in a sense that the output of the players are known internally or externally to the meta-player.

Schurz have made some investigations on the short term and long term performance of a simple *MI*. The results have shown that, despite of failures in short term, MI will certainly give a better result in long run. For a best player $B \in \{P_1, P_2, \ldots, P_n\}$, if there exists a time t_n at which the best player wins all other alternative player by having maximum success rate $maxsuc_n$, the *bMI* will choose this best player as the new favorite and predict what *B* predicts. In a long run, the success rate of *MI* approximates non-MI-player's highest success rate: $\lim_{n\to\infty}(\text{maxsuc}_n - \text{suc}_n(\text{MI})) = 0$. And for short run, $suc_n(MI) >= maxsuc_n - (t_n/n)$.

Every time when bMI choses a new favorite it is itself losing a point on every switch as bMI regrets of having predicted what the previous favorite had predicted. If bMI loses a point on every switch, and there are two players always fighting for the best position, bMI will be losing a lot of its own success rate, resulting the whole prediction into a failure.

This limitation of bMI is a result of the convergent oscillation between the two best players when they simultaneously win over each other.

To overcome this defect, Schurz has proposed to setup a threshold μ which is added on $maxsuc_n$ and the new best player should have success rate of $(maxsuc_n + \mu)$ to win over the current best player. Now in long run, *μMI* μ-approximates non-MI-player's highest success rate by $lim_{n->\infty}\ (maxsuc_n — suc_n(MI)) =< \mu$. And in short run, $suc_n(MI) >= maxsuc_n - (t_n + 1)/n - 2 * \mu$, where t_n is best player's winning time.

2.1.1.2 Weighted-Average Meta-Inductivist

The strategy of a weighted-average MI is to weight the player's predictions by 'attractiveness' which is the extension of the concept of success rate, and take the weighted average to give a prediction. This strategy can only be applied for the real valued games, but not for the binary game with the predictions of logical 0's and 1's as the average of 0 and 1 is a real value number.

From (Cesa-Bianchi & Lugosi, 2006), using a polynomially weighted average forecaster we can relate the attractiveness in to it. The weighted-average meta-inductivist (wMI) is defined as follows. For every non-MI-player *P* the attractiveness of that player at any time *n* will be $at_n(P) := suc_n(P) — suc_n(wMI)$ if and only if $suc_n(P) > suc_n(wMI)$ on that round, else $at_n(P) = 0$. If *PP(n)* are all the non-MI-players with positive attractiveness at round *n*, the wMI strategy at any round *n* can be defined as,

$$\forall n \geq 1: p_{n+1}(wMI) := \frac{\sum_{P \in PP(n)} at_n(P) \,.\, p_n(P)}{\sum_{P \in PP(n)} at_n(P)}$$

So the strategy of *wMI* can be understood as the attractiveness weighted average of next round predictions of attractive players. If by any chance, there exists no attractive player at round *n*, and *PP(n)* gets empty, $p_{n+1}(wMI)$ can be set as any random number.

Both the *bMI* and *μMI* players mentioned previously depend on the predictions of a single best player at a round. Say if the best player in the game is an evil-minded player who wants to fail the game, or predicts wrong data after reaching to the top, the meta-player immediately loses success rate. Say if all the players are motivated to turn apart the game and starts giving false data as soon as reaching the highest success rate, the meta-player

collapses with zero success. But in case of any player who predicts wrongly as soon as attaining the attractiveness above the designed threshold, it will still not be able to deceive the meta-player as it takes the weighted average of all the attractive player's predictions. So for *wMI*, the meta-player fails only if all the players in the game are giving wrong predictions or the predictions are not in the designed threshold of deviation from the actual event.

Now, for a long run, non-MI-player's success will be $suc_n(wMI) := lim_{n->\infty} (maxsuc_n — suc_n(MI)) = 0$ and for short run, $suc_n(MI) >= maxsuc_n - \frac{\sqrt{m}}{n}$ where m is the number of accessible alternative prediction strategies and n is the total time till the event. In long run, $n>>m$ such that the factor is nearly zero.

2.2.1.3 Regret-Weighted Meta-Inductivist

In the *wMI* prediction game of attractiveness, the weightage was done with the difference in success rate of non-MI-players and *wMI* player itself. There is one more way to weight the players and that is to weight by regrets. Regrets are the exact opposites of success. If success is 40%, regret is 60%. Using this concept of regret, another meta-player has been derived and called as regret-weighted meta-inductivist (*RW*).

Despite the unavoidable short-run loss in success rates compared to the best player, in a long-run, the regret-weighted meta-inductivist is able to give the predictions as accurate and reliable as the best accessible player's prediction in the game, even is the success rate of each player are non-convergent. Talking about the simple MI player which imitates the alternative player, MI could also have been imitating a deceiving data.

Weightage can be given to each players on the basis of relative regret $reg_i(P_i) := (abs_i(P_i) - abs_i(RW))/n$ where $abs_i(P_i) - abs_i(RW) := Reg_i(P_i)$ is called absolute regret of *RW* with respect to P_i. Now the weightage of player P_i is $w_i(P_i) := Reg_i(P_i)/n$. Such that, the prediction of *RW* for $n>0$ and positive denominator is

$$p_{n+1}(RW) := \frac{\sum_{i=1}^{m} w_n(P_i) * p_{n+1}(P_i)}{\sum_{i=1}^{m} w_n(P_i)}$$

Should it happen that the denominator is zero, $p_{n+1}(RW)$ is set a random number. Similar to the *wMI* strategy, this strategy is inappropriate to binary prediction games.

2.2 Deep Learning and UAV-collected RGB Images for Agriculture

There are various challenges for obtaining optimum processing of multispectral images, for instance, (Laliberte, Goforth, Steele, & Rango, 2011) described some of the challenges and solutions associated with obtaining radiometrically and geometrically calibrated orthomosaic for accurately classifying rangeland vegetation. They applied object-based image classification approach, and automated batch processing to generate species-wise classification with an accuracy of 87 %. Their results obtained good correlation between ground spectral reflectance and spectral reflectance from airborne/satellite data for their selected vegetation/soil targets, with an accuracy of R2 = 0.92. (Honkavaara et al., 2012) investigated on the applications of UAV embedded with a hyperspectral camera and high resolution RGB based camera for precision agriculture. They developed an image processing pipeline for robust production of high density point clouds and integrated the orthomosaic with the hyperspectral reflectance, which was applied in the process of biomass estimation. They also discussed on the factors such as image quality, processing framework, surface models which highly affects the accuracy of biomass estimation. Their results confirmed that it is possible to apply light weight, low cost imaging for UAV remote sensing and proved to be powerful and cost efficient technology for possible remote sensing applications.

(Agarwal, Snavely, Simon, Seitz, & Szeliski, 2009) applied an efficient method for image matching where he identified small amount of feature match for each images, instead of matching all the image to each other which still preserved enough feature matches for structure from motion increasing efficiency in terms of time. (Wu, 2013) studied on the large scale image reconstruction from linear time structure from Motion. They introduced a preemptive feature matching that was able to reduce the image pair match by 95%, and still recovered to detect good feature match for the reconstruction process. They were able to examine the complexities in time for the gradient bundle adjustment methods. Their

results show that many sub steps in the process of image reconstruction like feature detection, matching, filtering and stitching required O (n), where n being the number of images in the function of time using their novel bundle adjustment strategy which previously required O (n4) in the function of time. Their method also maintained high level of precision through regularizing the triangulation phase across the feature matches, until every feature map are triangulated.

(D. Turner, Lucieer, & Watson, 2012) presented an approach for robust radiometric and geometric calibration to enable accurate UAV photogrammetry using SfM algorithm. The flight images were processed to create a 3D point clouds in an arbitrary coordinate system, which was later transferred into real world coordinate system using two techniques, either with direct georectification approach that utilized the estimated camera coordinates through camera EXIF file, or through a Ground Control Point (GCP). The point cloud was used to produce DTM which was required for the correction of the images, and subsequently producing an orthomosaic of the project area. An absolute spatial accuracy of 65-120 cm was achieved using direct georectification, whereas a more accurate results were obtained with GCP technique of 10-15 cm. Similarly, the comparison between two approaches for determining the crop height determination namely, i.) Difference method and ii.) Statistical method has been compared by (Grenzdörffer, 2014), with recommendation on difference method due to its simple and accurate results when supplied a high resolution reference DTM whereas the statistical approach doesn't necessarily requires a reference DTM for computation of CHM.

Likewise, (Enciso et al., 2016) studied the application of UAV for plant phenotyping analysis using two different UAV platforms, mainly an octacopter and a quadcopter for the purpose of monitoring plant growth, cover and yield forecasting in tomato & potato plantation. The octacopter was embedded with two sensors, namely, 12 Megapixel RGB camera and Tetracam multispectral camera recording RGNIR wavelengths. The flight was taken on March 17, 2016 at an altitude of 30 m, maintaining an effective front and side overlap of 80% and 70% respectively. These data were processed using SfM algorithm to generate an orthomosaic and surface models which

resulted in the extraction of crop height, cover and plant health using vegetation indices acquired using Tetracam multispectral camera for their entire field plots. Similarly, the application of cheap modified infrared cameras with the IR mirror filters disassembled from a normal RGB camera and replaced with two band pass filters will be particularly tested for its application in monitoring plant health in banana plantations (Hunt et al., 2010).

(Krizhevsky et al., 2012) trained a deep ConvNet, which comprised of 60 M parameters having five convolution layers & subsequently, maxpool layers together with two fully connected layers, on the ImageNet datasets consisting of millions of dataset to classify 1000 classes. They implemented an efficient GPU framework and a regularization technique for the rapid performance and prevent overfitting during the training phase. Another popular dataset, namely, MS COCO (Lin et al., 2014) is comprised of 2.5 M labelled dataset captured from 328 K ground based images that is able to accurately classify 91 different object types. Likewise, there is another publicly accessible dataset namely, PASCAL VOC (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010) has been trained from 20k ground based training images, and is able to classify 20 different object types.

(Carlet & Abayowa, 2017) presented an improved the performance of YOLOv2 detector for the purpose of fast vehicle detection in an aerial imagery which performs cutting edge detection at 4x speed. Their dataset included several aerial imagery which are publicly available, some of them are: Vehicle Detection in Aerial Imagery (VEDAI), AFVID and DLR3k etc. They made their neural net shallower to increase its output resolution and changed the net shape to match the aspect ratio of data, which increased the speed making it near real time object detector for aerial imagery. However, their precision and recall (Carlet & Abayowa, 2017) when compared to faster RCNN (Ren, He, Girshick, & Sun, 2015) is still slower. Unlike Yolov2, many literatures has also implemented region proposal based neural networks (Girshick, 2015) for the purpose of real time object detection applications (Gavrila & Philomin, 1999; Redmon, Divvala, Girshick, & Farhadi, 2016). For instance, (Ren et al., 2015) implemented region proposal based approach, where

they applied 300 regions proposal/image, and achieved an accuracy in terms of detection of 73.2% mean average pixel (mAP) for VOC 2007 dataset and 70.4% mAP for 2012 dataset. Likewise, (He, Zhang, Ren, & Sun, 2016) implemented the residual learning approach for training the deeper neural net, on ImageNet dataset (Krizhevsky et al., 2012) with their network consisting of 152 layers, which is nearly 8 times larger than some popular networks like VGG net (Simonyan & Zisserman, 2014), yet manage to attain minimal complexities. Their result achieved minimal error percentage of 3.57%, when trained on ImageNet dataset, likewise, achieved 28% improved results on COCO dataset; as a result of which it managed to score first position in ILSVRC 2015 classification competition (He et al., 2016). On other hand, (Simonyan & Zisserman, 2014) applied convolutional approach to train a deep neural network for the purpose of large scale detection, where their network comprised of 19 layers implemented with 3 x 3 kernels which demonstrated good results on both localization and classification.

Unlike region proposal based object detection algorithms (Ren et al., 2015), Single Shot Multibox Detector (Liu et al., 2016) as the name suggests follows a unified detection approach implementing a sole deep neural network. Their approach estimates a predefined set of bounding boxes which could be scaled over different aspect ratio depending on the objects to detect. This method also terminates the creation of region based proposals, together with other resampling techniques and integrates all workflow in a sole neural net, making the overall procedure simple and uncomplicated. Their results (Liu et al., 2016) were tested on VOC 2007 dataset (Everingham et al., 2010), and attained 74.3% mAP for an input image dimension of 300x300, whereas, an accuracy of 76.9% mAP for input dimension of 512x512, both at 58 frames/second. The results from the conventional SSD was impressive, however (Jeong, Park, & Kwak, 2017) further enhanced the performance of the detector by swapping the VGGNet (Sujana, Abisheck, Ahmed, & Chandran, 2017) in original detector to ResNet. This modification further improved the performance, which when trained with VOC dataset (Everingham et al., 2010) resulting in 78.5% mAP with input image dimension 300x300 at 35 frames/second, whereas the accuracy further improved with 512x512 input dimension at 80.8% mAP but with 16.6 frames/second.

Similarly, (Girshick, Donahue, Darrell, & Malik, 2014) integrated the region proposal technique with the convolutional implementation, referred as RCNN, which enhanced the performance by boosting the precision by 30% i.e. attained mAP 53.3% when trained on VOC dataset (Everingham et al., 2010). They also suggested in applying transfer learning in cases where there are insufficient training dataset which was followed by fine tuning to significantly improve the overall performance.

Furthermore, (Puttemans, Van Beeck, & Goedemé, 2018) performed research in finding an optimal algorithm for automated and robust detection on aerial images consisting palm tree plantation, where they applied object detection algorithm based on (Viola, Jones, & others, 2001) which followed the principle of boosted cascade of simple features. With significant reduction in computation costs and the addition of robust deep learning architectures which performs well on both classification and detection; it has become feasible for solving the complex task of quick and real time object detection and classification in an aerial imagery. Furthermore, they (Puttemans et al., 2018) mentioned that many other pretrained network such as Caffe model zoo (Jia et al., 2014) might serve as an alternative to their approach.

YOLO object detection algorithm (Redmon & Farhadi, 2017), uses a single neural network that predicts the class probabilities and bounding box directly for the input full images in one evaluation; enabling this architecture to work extremely fast (Redmon et al., 2016). This algorithm helps neural network output precise bounding box. Likewise, with the implication of a finer grids like (19 x 19) (Redmon & Farhadi, 2017) decreases the chance of assigning same grid to multiple objects, and also allows neural network to output the bounding box in any aspect ratio which aren't dictated by striding size as in sliding window classifiers (Girshick, 2015) as well as outputs much precise coordinates. Furthermore, the algorithm is a convolutional implementation ie. the grids are supplied in a full size image as a single convolutional network, which makes the network pretty fast and enables real time object detection (Redmon et al., 2016; Redmon & Farhadi, 2017).

One of the main concerns in deep learning is subjected to the training data and the hardware for computation, and without large amount of data, our research focused on

transfer learning, where we utilized existing deep learned models which was trained on large dataset such as COCO (Lin et al., 2014) for adapting to our task of object detection through fine tuning the weights of the convolutional neural network onto our single new object class (i.e. Banana Plants) . All the above literatures are very significant and outlines proposed methodologies together with the limitations which will be vital in adapting a methodological workflow for our research. These literatures suggests that it is possible to apply the modified near infrared remote sensing, multitemporal crop surface models and deep learning for monitoring crop health, growth and stand count attributes. The purpose of this study is to highlight and discuss related literatures, which can be highly potential in their ability of producing accurate, faster and efficient results to be adapted in the workflow of our work.

2.3 Open-source UAV Image Processing for Crop Health Monitoring

Sustainable farm management strategies depends on accurate prediction of crop yield, which are normally generated from the computation over remotely-sensed data, i.e. vegetation indices. These indices combined with crop surface model, forms a more concrete methodology which has proven to improve the accuracy in for crop yield prediction (Geipel, Link, & Claupein, 2014). They gathered multitemporal RGB images using unmanned aerial vehicle from early to mid-growth phase, which was processed to simple vegetation index for crop classification, whereas crop height information was extracted from multitemporal crop surface models at multiple resolutions. Their results demonstrated coefficient of determination (R2) value of 0.74, and best results were obtained at a spatial resolution of 0.04 m/pixel. Their results proved that combining vegetation indices with crop height information could form a more concrete methodology for predicting corn yield.

The basic requirement for photogrammetric mapping from UAV is a collection of properly overlapped image pairs taken from an altitude with consideration of geometric and radiometric effects (Dare, 2005), depending on required precision in work. After collecting images, image processing software follows some algorithm like structure from

motion (Westoby, Brasington, Glasser, Hambrey, & Reynolds, 2012) to stitch together the aerial images. A general framework for any image processing software looks like in Figure 1 where selected images are first pre-processed for geometric and radiometric calibration using different models like in (D. Turner et al., 2012). Then multiple pair of images that share same view are detected to extract and match key points and tie points from each images (Lowe, 2004). These points help align the images, and allows to produce sparse three dimensional (3D) points and camera positions. After the aligned sparse cloud of 3D points (referred as ‘point clouds' (Rosnell & Honkavaara, 2012) is created, an algorithm is applied to remove lens distortion using camera calibration parameters (Colomina & Molina, 2014), and further, 3D dense point cloud is reconstructed. The dense reconstruction allows to deploy all pixel values to create a mesh, which is further textured to obtain a textured mesh (Remondino, 2003). The mesh is used to project an orthophoto (aka. orthomosaic) using some projection system. The Digital Surface Model (DSM) and Digital Terrain Model (DTM) - which describes surface and terrain of study area - are interpolated from either dense point cloud or mesh, with the classification of points into all and ground points respectively.

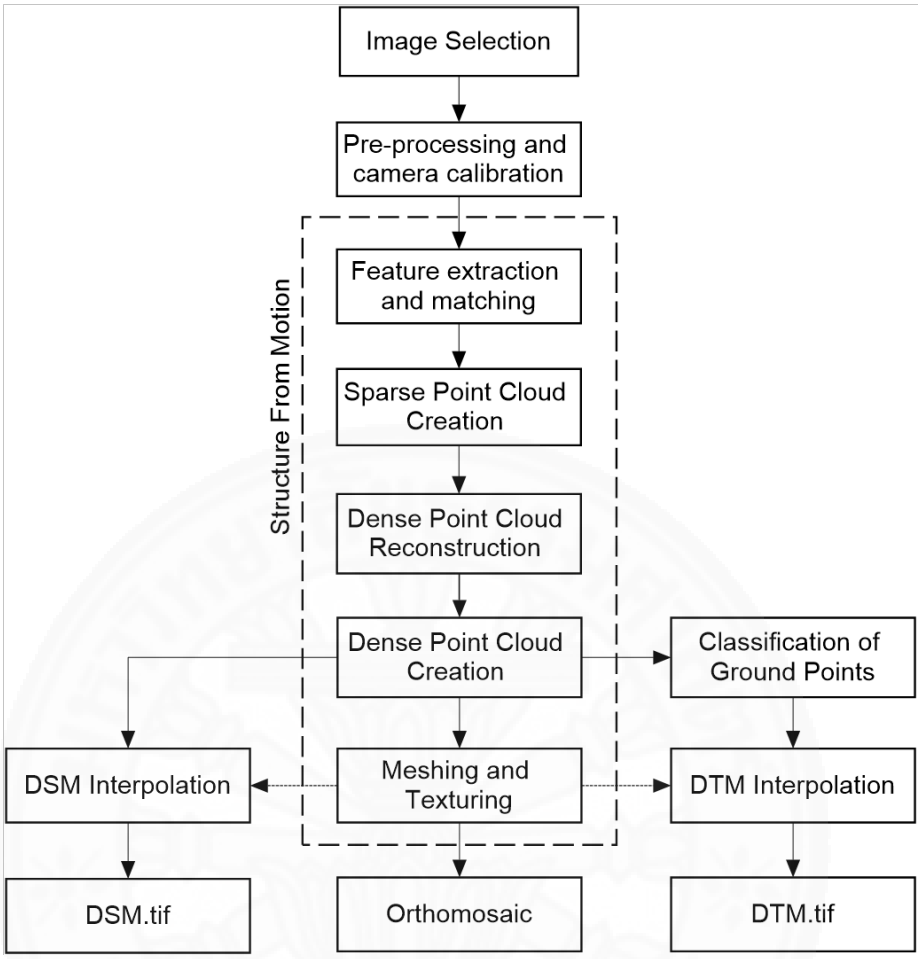


Figure 2.1 General workflow for UAV image processing

The image processing methodology among different software are almost similar, while the algorithms and theory used to perform each task may vary. There are many tools available to perform specific tasks and some offline and online software are available to perform all the tasks. In this paper, we focus on web-service because we target the users with less computational expense, like farmers and students.

Closed-source softwares (e.g., PhotoModeler Scanner, Eos Inc; PhotoScan, Agisoft) and web-interface of Pix4Dmapper (Pix4D, 2014) and DroneDeploy (Smith, 2017) provide image processing for a certain trial period. Some free web-interfaces (e.g., Photosynth, 123DCatch, etc.) used to be available, but discontinued now. Also, there are individual structure from motion (SFM) tools like VisualSFM (Wu & others, 2011) and

Bundler (Snavely, 2008). But it is tedious for a lay-man users to arrange and connect all the tools. An open-source web-service to integrate all the tools and algorithms is thus necessary.

Open Drone Map (ODM) (Org, n.d.) is an open-source drone mapping software to generate maps, point clouds, 3D models and DEM. ODM is available in command line interface and user-interface (UI). For a UI, it provides a web-interface called WebODM that can be hosted locally or globally by user as an API, to call UAV image processing toolkit. ODM is now in a development phase and is being supported by many contributors over internet (see https://github.com/OpenDroneMap/ for details), where they collaboratively integrate their ideas and solution to existing problems in image processing.

ODM's workflow for UAV image processing uses open-source algorithms like Open Structure from Motion (OpenSFM), to generate point clouds and 3d mesh that is used for DSM and DTM generation. And the extensive advantage of ODM is that it provides an API that allows to host our own image processing web-service. The developer acquires an authentic token ID by entering username and password first. After that, projects, tasks, can be created and allowed to run to obtain the *assets*, which are the outputs of ODM. A distinguished feature allowed by ODM is that it allows to create multiple processing nodes, to process images via an API, which allows to share the processing power of multiple computers, or multiple segments of same computer. In our research, we develop an algorithm for our own UAV image processing web-service using ODM API for agricultural applications, and extend the usability of ODM (version 0.4.1) to generate vegetation indices for crop monitoring.

CHAPTER 3
MATERIALS AND METHODS

3.1 Weather Scenario Generation Game

Weather Scenarios are the key input to crop modelling, which is why those scenarios need to be reliable to make crop models reliable. We develop a game called Weather Generation Scenario Game (WSG) to provide strategic solution to three problems that were previously mentioned. For simplicity, we state our problems again:

1. Determination of most successful scenarios
2. Classification of scenarios into fresh and stale
3. Generation of new scenarios from fresh scenarios

In this section, we will show the framework of the workflow that is bound to solve these three problems, which will be provided by WSG. To demonstrate our formal results and also shed lights to their applications, we feed the scenarios provided by an existing weather generation web service to our game. The workflow of WSG goes as follows.

3.1.1 Obtain Weather Scenarios

Weather generators provide multiple weather scenarios, which are calculated from a historical dataset, for variables like rainfall, solar radiation, minimum temperature and maximum temperature (abbr. RAIN, SRAD, TMIN, TMAX respectively). They use RAIN as primary variable to derive other variables using some regression relationship between them, and from the previous values of the variable.

A web-service that uses two weather generators: DisAg (Hansen & Ines, 2005) and predictWTD (Han & Ines, 2017), to provide weather scenarios including RAIN, TMAX, TMIN and SRAD was developed recently as DisAg Weather Generator Web Service (DISAGWS) (Chinnachodteeranun, Hung, Honda, Ines, & Han, 2016). For the prediction game, we use HTTP GET Request to invoke DISAGWS to obtain weather scenarios in Weather Data (WTD) format.

HTTP GET Request:

http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DISAGWS/rest/generate?num=100&from=2\\016,1&to=2017,1&weatherhistory=http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/DataTransformation/rest/transform/http://ec2-52-69-188-223.ap-northeast-1.compute.amazonaws.com:8080/HW_SOS_2/service?service=SOS&version=2.0.0&request=GetObservation&MergeObservationsIntoDataArray=true&offering=weatherhistory_urn:IBUNYA:AMeDAS-NIAES:GAMAGOORI-51281

Modified HTTP GET Request:

http://ec2-52-69-188-223.ap-northeast1.compute.amazonaws.com:8080/DISAGWS/rest/generate?num=100&from=2016,1&to=2017,1&weatherhistory=https://drive.google.com/open?id=1QspUU73P-yQXeTk-hLPhNZibBEMYSJGm

HTTP GET Request asks for the weather scenarios from 2016 to 2017 using the historical data obtained from GetObservation request of a weather station. The data obtained from it was manually cropped to separate the data from 2016 on-wards to be used as the actual events *e* in the game. And the scenarios generated from the historical data until 2015 are considered as predictions *p* produced by OI players for the game. A sample of weather data from 1979 to 2015, shown in table 1 is supplied in a modified HTTP Get Request given below as a link to Google Drive. It was later noticed that some of the scenarios were missing the data for initial two months. Discarding the missing data, 100 weather scenarios for 156 days was obtained from DISAWGS for the game.

Table 3.1 Sample of Historical Weather Data from 1979 to 2015

@ DATE	SRAD	TMAX	TMIN	RAIN
1979010	0.4	15.3	10.8	0
1979011	1	13	2.3	0
1979012	0.4	13	1.5	0
…	…	…	…	…
2015363	1	10.4	3	0
2015364	1.5	11.2	1.9	0
2015365	0.9	11.4	3.1	0

3.1.2 Determination of the most successful scenario

We define the most successful scenario as the scenario with highest success rate in the past. To understand this, let's define a prediction game, which is a pair $((e), \Pi)$ where

- $(e)=(e_1, e_2, \dots, e_n, \dots)$ is a possibly an infinite sequence of events e_n, where n is an instance of time corresponding each round of the game.
- $\Pi = \{O_1, \dots, O_m, M_1, \dots, M_k\}$ is a finite set of prediction methods (aka players) whose task, in each round n, is to predict the next event e_{n+1}. There are two kinds of players:
    - $O_1, \dots, O_m$ are object-level players who base their predictions on the observed events.
    - $M_1, \dots, M_k$ are meta-level players who base their predictions on the predictions of object-level players.

  Several other notions associated to this game are:
- $p_n(\mathrm{X})$ is the prediction p_n of any player $X \in \Pi$ for event e_n delivered at time $n-1$. In binary prediction game, $p_n(\mathrm{X}) \in \{0,1\}$.
- $l(p,e) = |p-e|$ is the *natural loss* function describing the absolute distance between p_n and e_n.
- $score_n(X) = 1 - l(p_n, e_n)$
- $abs_n(\mathrm{X}) = \sum_{i=1}^{n} score_n(X)$ is the absolute success of X at time n.
- $suc_n(\mathrm{X}) = abs_n(\mathrm{X})/n$ is success rate of X at time n.

Depending on the type of data, *events* can be of two forms: binary and *real-value*; and the prediction game is called *binary game* and *real-value game* respectively. For a binary game, $e_n \in \{0,1\}$ and for real-value game, $e_n \in [0,1]$. For example, wet (rain) and dry (no rain) events of rainfall are binary events and the amount of rainfall is a real-value event. Loss function provides the foundation for scoring the players which further derives success and regret of any players and this function can be of different forms, such as *homogeneous* and *non-homogeneous* (Schurz & Thorn, 2017). A simple example of loss function is: $l(p,e) \in \{0,1\}$ where, $l(p,e) = 0$ if $p = e$ and $l(p,e) = 1$ if $p \neq e$. Further, $score$ is given to each players as $1 - l$ on each round to derive the *absolute success* and

success rate. For weather prediction, this simple loss function may not be appropriate for real-value game; and we will discuss this later for the games played on other weather variables except RAIN.

The solution to our first problem demands the most successful scenario. With respect to our given prediction game $((e), \Pi)$, where $\Pi = \{O_1, \dots, O_m, M_1, \dots, M_k\}$, the most successful scenario at round n is the scenario of player B_n:

$$B_n \coloneqq argmax\{suc_n(\mathrm{X}) \mid X \in \{O_1, \dots, O_m\}\}$$

B_n being the most successful, solves our first problem. Simulations in this research shows the success rate of scenarios plotted with number of rounds, where the top-most scenario at round n is the most successful scenario for n. Success rate $suc_n(\mathrm{X})$ will be derived differently, depending on the loss function for binary game and real-value game. We still need more scenarios which are successful than some scenarios, which leads us to the strategic solution to our second problem.

3.1.3 Classification of scenarios into fresh and stale scenarios

As a single best scenario is not enough for crop modeling, and more number of scenarios results better simulations in agricultural application, in our second problem, we classify the obtained scenarios into fresh and stale. The notion of fresh and stale is that all the scenarios are usable such that a scenario stale at one instance of time can be fresh in another instance.

To classify the scenarios into fresh and stale, we determine a scenario that can be taken as a reference scenario that separates fresh from stale. WSG generates this reference scenario by using a meta-player's strategy. We define fresh and stale scenarios as: In the prediction game $((e), \Pi)$, where $\Pi = \{O_1, \dots, O_m, M_1, \dots, M_k\}$, at round n, some scenario Om is *fresh* if $succ_n(O_m) > succ_n(M_k)$, M_k being a meta-player's scenario (aka. reference scenario). Otherwise, O_m is *stale*. The meta-player's strategy is different for binary game and real-value game. We show binary game for RAIN and real-value game on TMAX variable in the following two subsections.

3.1.3.1 Binary Game for RAIN

For binary game, we use a meta-player called *simple MI*, which predicts what the best player B_n predicts for the next round, which is given as, $p_{n+1}(bMI) \coloneqq p_{n+1}(B_n)$. The prediction of B_n is called *optimal prediction*. This definition being the foundation, WSG therefore predicts at each round, what the best player B_n from previous section predicts for the next round, making B_n an *OI* player with maximum success rate at round *n*. And the success rate of bMI at each round is a reference that separates fresh from stale scenarios. By the end of all rounds, we obtain a reference scenario as shown in figure 13 in results section with dashed line.

For demonstration, we play binary game for RAIN variable of DISAWGS. Before applying bMI on binary values of RAIN as wet and dry (or rain and no-rain), the real-values obtained from DISAWGS is converted to binary form of $\{0, 1\}$ with a simple transformation definition as: zero-rain (dry) = 0 and non-zero (wet) = 1. A natural loss function $f_z = \{0,1\}: 0\ for\ p_n = e_n\ and\ 1\ for\ p_n \neq e_n$ to calculate loss. Various loss functions can be found in (Schurz, 2008, 2012; Schurz & Thorn, 2017). Selection of these loss function depends on the type of data.

OI's predictions p_n provided by DISAWGS immediately contradicts e_n, and for 156 days of RAIN data, bMI continuously predicts at every round, what player B_n predicts for round $n + 1$. The first B_n player is chosen in an ascending order, and in the following rounds, as the success rate of B_n increases, there will fewer number of such players with maximum success rate. Theoretically, the success rate of such player *B* will approximate the maximum success rate among all non-MI players (in our case, scenarios).

Recalled from (Schurz, 2008), for each prediction game $((e), \Pi)$, where $\Pi = \{O_1, \dots, O_m, bMI\}$ including a best object-level player B, the following holds:

- Short-run: $(\forall n \geq 1:)\ succ_n(bMI) \geq maxsuc_n - \left(\frac{n_B}{n}\right)$, where n_B is B's winning time, after which B continuously wins for all rounds.
- Long-run: bMI player's success rate approximates the maximal success rate of the non-MI-players: $\lim_{n\to\infty}\left(maxsuc_n - suc_n(bMI)\right) = 0$.

3.1.3.2 Real-value Game for other variables

WSG on real-value data uses a different meta-player strategy called *regret-weighted meta-induction* (abbr. *RW*) to determine the reference scenario. Recalled from (Schurz & Thorn, 2017), RW is identical to polynomial weighted forecaster described in (Cesa-Bianchi & Lugosi, 2006) in page 12, and is defined as: Regret-weighted meta-induction for a set of object-level players $\{O_1, \dots, O_m\}$:

- *Absolute regret* of RW $reg_i(O_i) \coloneqq (abs_i(O_i) - abs_i(RW_i))/n$ where $abs_i(O_i) - abs_i(RW_i) \coloneqq Reg_i(O_i)$ is called absolute regret of RW with respect to O_i.
- The predictions of *RW* for $n > 0$ are defined as:

$$p_{n+1}(RW) \coloneqq \frac{\sum_{i=1}^{m} w_n(O_i) \times p_{n+1}(O_i)}{\sum_{i=1}^{m} w_n(O_i)}$$

Where, $w_n(O_i) \coloneqq \max(reg_n(O_i), 0)$. The denominator must be positive; otherwise $p_{n+1} = some\ random\ value$.

The scenarios generated from self-contradicting original scenarios are not expected to exactly predict the real event. Also for crop growth, precise time of rainfall and degree of temperature or solar radiation is least concerned, if amount of it being received is similar. For example, amount of solar radiation received by crop in a week is more significant, regardless the exact time at which crop received it in that week. It is less of a concern if plant absorbs any amount of rainfall on a Tuesday morning rather than on Wednesday afternoon. This is why setting up a degree of threshold in loss function of real-value game is necessary to provide the flexibility it requires for WSG. If threshold of $\alpha°$ unit of temperature is considered, loss function $l \in \{0, 1\}$ is setup as: $l(p, e) = 0$ if $p \in \{e - \alpha, e + \alpha\}$ and $l(p, e) = 1$ if $p \notin \{e - \alpha, e + \alpha\}$.

According to theorem 1 of (Schurz & Thorn, 2017), *RW* might face some short term possible regrets, but in long run, *RW* 's relative regret is upper-bounded to $\sqrt{m/n}$ (m: number of alternative players available) and approximates to *0* when $n \geq\geq 1$ or else it oscillates endlessly with $limsup$ converging to *0*. But when we applied *RW* on the data obtained from DISAWGS with our loss function, even if the flexibility in threshold was increased, success rate of *RW* was still less than all the scenarios of TMAX. This is because

RW is predicting weighted-average of predictions of all alternative players, who are themselves gaining high regrets. This is doomed to happen because in weather generator, primary variable - dry and wet rainfall scenario - are themselves contradicting from observed events. This results into high difference in observed value (e_n) and predicted value (p_n), and our designated loss function with threshold of $\alpha°$ is not appropriate for our sample data. Leaving behind the problem of formation of proper loss function for WSG for future research gaps, we rather test WSG on the data that is randomly created, in the results section.

To test the performance of WSG on randomly generated temperature scenarios, historical weather dataset of Bangkok Metropolis Station TH000048455 since 1987 was downloaded from National Oceanic and Atmospheric Administration (NOAA)'s Climate Data Online System (https://www.ncdc.noaa.gov/cdo-web/). For our study, we took real-valued average temperature (abbr. TAVG) data for 1000 days from the dataset. Then we generated 1000 scenarios within random deviation of 1 to 5 percent from the corresponding day's real event. These randomly generated TAVG scenarios are not affected by rainfall which is why they have statistically less oscillation of success rate. After generating scenarios, we applied *RW*'s meta-strategy to classify them in WSG with loss function of threshold with $\alpha = 2°$.

The output of the classification from this section are the fresh scenarios which will further be used to generate new optimal scenario and with some algorithm aggregating multiple fresh scenarios, WSG produces ready-to-use number of best weather scenarios for crop simulations in the following section.

3.1.4 Generation of new scenarios from fresh scenarios

The outcome of the solution to our second problem are the fresh scenarios, and now we like combine new strategy into WSG to generate a new optimal scenario and produce new set of fresh scenarios. There can be several alternative approaches like major/weighted voting, simple/weighted averaging or even stacking of several models to generate a new scenarios. But as we focus on prediction game, our algorithm to generate a new optimal

scenario from a set of fresh scenarios obtained from classification can be viewed as a sub-game within WSG. With respect to prediction game $((e), \Pi)$, where $\Pi = \{O_1, \dots, O_m, M_1, \dots, M_k\}$, suppose that $X \subseteq \{O_1, \dots, O_k\}$ are fresh scenarios and *MI* is the same meta-player used to classify the fresh scenarios, then the prediction game played among $\{O_1, \dots, O_k, MI\}$ is the sub-game. There is no point of playing sub-game if there already exists only one or no fresh scenario, because *MI* is already most optimal. The results are shown in chapter 4.

3.2 Deep Learning based Banana Plant Detection and Counting using UAV collected RGB Images

3.2.1 Dataset

A banana farm in Phra Nakhon Si Ayutthaya province of Thailand (14°19'04.5"N \$100 °45'18.0"E) lost some banana plants at the third month of plantation, despite proper plantation. Sixteen acres of this farm was selected as our study area, for which a flight was planned for our DJI Phantom 3 UAV. High-resolution RGB aerial images of 4000x3000 pixels were collected from 40, 50 and 60 meters above ground for our planned flight. The side and front overlap was fixed to be 75\%, with flying speed of 10m/s and focal length of the camera was 4mm. Figure 2 shows the study area below.

Figure 3.1 Study area for DL based banana plant detection and counting

3.2.2. Workflow

In this research, we propose a strategic algorithm to count the number of banana plants in an aerial image. Let us assume a training dataset (D) = an orthophoto (I), with k number of training samples of banana plants. Then $\{D_1, \ldots, D_i, \ldots, D_n\}$ are variants of the orthophoto, which are obtained by using different processing methods $\{p_1, \ldots, p_i, \ldots, p_n\}$ on the original training dataset D. Each variant D_i is then tiled into smaller images to train a parameter-optimized DL algorithm to produce a trained detection model $\{m_1, \ldots, m_i, \ldots, m_n\}$. The model m_i is then used to detect banana plants on image tiles of each variant D_i of a separate testing area (any other farm except the training area), to produce multiple detection outputs $\{O_1, \ldots, O_i, \ldots, O_n\}$. These O_n detection outputs are then merged to get the final detection O of I. A schematic diagram of the algorithm is shown in figure 3, and to perform this algorithm to our problem of counting banana plants, we use the workflow as shown in a schematic diagram of figure 4.

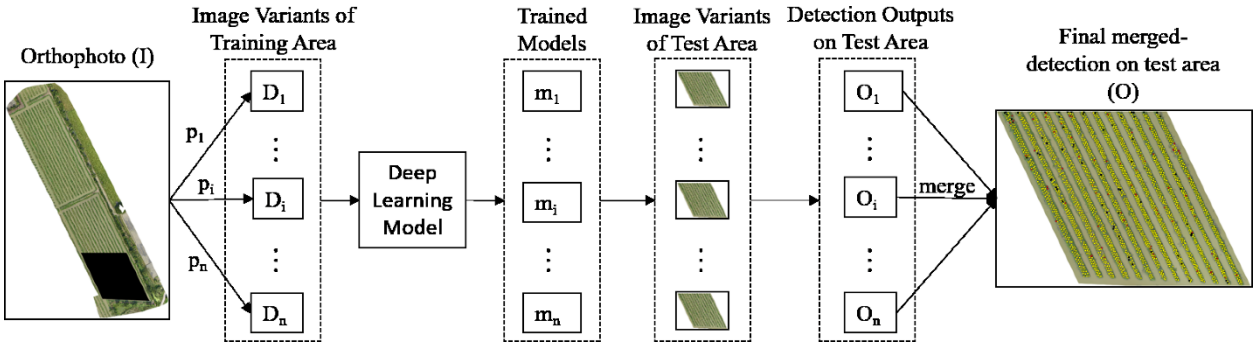


Figure 3.2 Proposed Algorithm for DL based banana plant detection and counting

The first step after collecting images is to create orthophoto (I) for images collected, from multiple flight altitude. This high-resolution orthophoto is of consists of RGB bands, and RGB image is not sufficient to obtain precise count of banana plants because of their irregular shape, effects of shadow, slope and aspect, and less number of training dataset for training the DL model. This is why, we process the orthophoto using various image processing methods (p_n), to produce different variants of orthophoto D_i. From D_i, we separated the training area and region of interest (ROI) (i.e. validation area), to test our final CNN model obtained after training. For both training and validating area, the large-sized images are then tiled to smaller size of 600x600 pixels.

After image preparation by tiling, next step is to carefully annotate banana plant samples seen on the image tiles of training area, by manually drawing bounding boxes of rectangular shape around the plants. Annotation of banana plants completely depends on human interpretation. The annotated image tiles of each D_i are then separately used to train and test CNN based parameter-optimized model for each D_i, producing n number of ready-to-use trained CNN model for each p_n. These trained models can now be used to test the image tiles of each p_n of our validation area, to produce n detection results of ROI. These multiple results are then finally merged to obtain the final detection results. The step-wise workflow of our algorithm is presented below, where we show some of sample results for the image tiles produced from orthophoto generated from the images collected from 40m altitude.

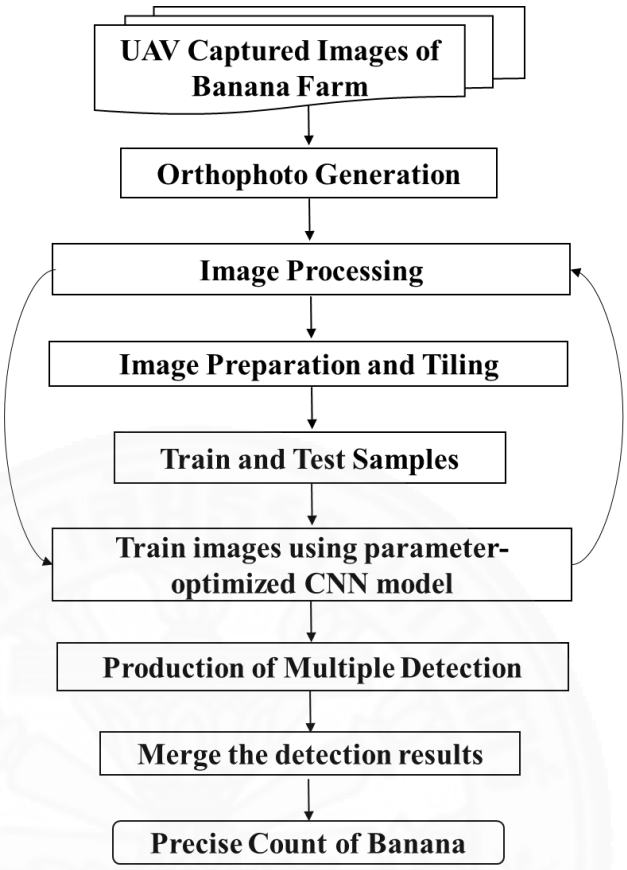


Figure 3.3 Workflow for DL based banana plant detection and counting

3.2.2.1 Orthophoto Generation

After collecting images, first step of post-processing UAV aerial images in photogrammetric mapping is to stitch together the overlapping image pairs. We used a commercial software called Pix4Dmapper (Pix4D, 2014) that is capable of adjusting geometric and radiometric effects, extract pixels on the images that share common view to produce 3D point clouds (D. Turner et al., 2012) that can be used to create texture model, and finally project them into an orthophoto (aka. orthomosaic). A high-resolution orthophoto of 23655x34390 pixel was produced from 398 images we collected from 40m flight altitude over our study area.

3.2.2.2 Image Processing

After generating orthophoto I, the next step is to use image processing methods to obtain multiple variants D_n of I. The reason to use these methods was figured out once we produced initial results based running CNN model on normal RGB images, in which the count of banana plant was considerably low. It was learnt that, edge detection being the fundamental step of CNN, DL needs objects (banana plant) to be properly distinguished from background, to make the objects distinct and produce better results. In our case, banana farm had grasses covering the soil and the color of banana plant and grass is almost similar. Also, the shadow of top leaves blocks lower leaves of the plants, making their shape seem irregular from zenith. Another reason for lower number of plants being detected is that CNN models require large number of training data. A solution to these problems is to separately apply different image processing methods p_n, and separately train CNN models on processed image variants D_n to produce optimized inference models from each set of uniquely processed images. The results O_n obtained from different inference models can be finally combined to obtain better detection.

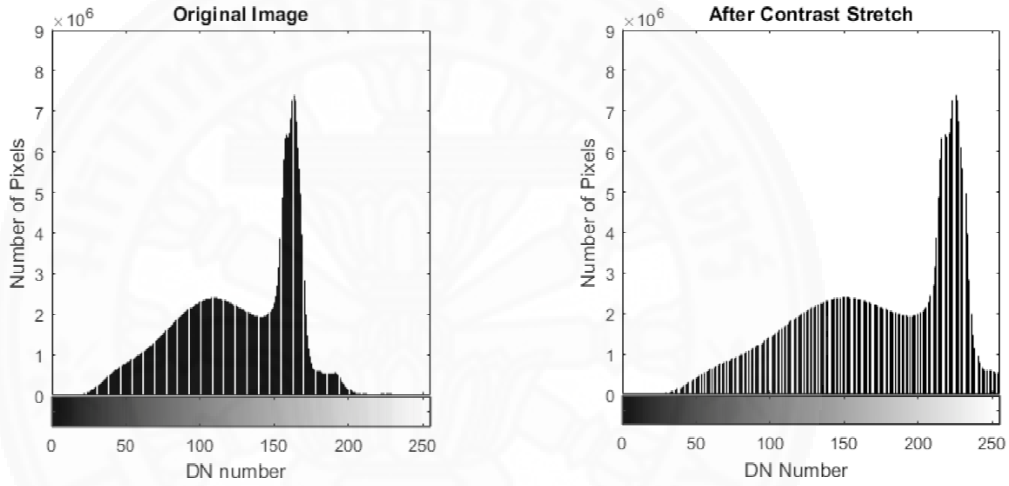
There exists number of methods for image enhancement. But we explore the methods that can enhance the vegetative properties on RGB large scale aerial images in canopy scale. For our purpose, we use three methods: Linear Contrast Stretch p_1, Synthetic Color Transform p_2 and Triangular Greenness Index p_3, to enhance color histogram and reflectance values in orthophoto image, which are explained below.

Linear Contrast Stretch

Each pixel of an aerial image consists of radiance value from real scene, which is converted into range of 0 to 255 to store in the image. These converted numbers are called Digital Numbers (DN). However, the range of radiance value is generally less than the full range of DN. Due to this difference in available range and actual range, images have low contrast and do not use the full range of display. In contrast enhancement (Schowengerdt, 2006), these DN numbers are transformed into Gray Levels (GL) display space, using a mapping function, to improve visual quality of image. This transformation of DN into GL

such that DN range fills available GL range, is called *Contrast Stretch*. Similarly, one way is to fill the display range of [0, 255] from minimum DN to maximum DN of image, and this is one of the methods of *Linear Contrast Stretch* (abbr. LCS).

We use *saturation stretch* of 1% with *linear stretch* to obtain greater increase in contrast. In this approach, min-max range of DN is made to fit to the range of GL i.e. [0, 255], and 1% of upper and lower extremes of DN range are "saturated" (clipped). This exploits the radiance values by brightening the pixels of banana plants and their leaves in our image. Figure 5 shows the histogram of original RGB orthophoto and contrast stretched orthophoto, and figure 6(a) shows contrast-stretched orthophoto.



(a) (b)

Figure 3.4 Histogram of (a) Original RGB orthophoto and (b) Contrast stretched orthophoto

Synthetic Color Transform

Synthetic Color Transform (abbr. SCT) is a process to assign synthetic colors to a gray scale image or a single band of image. One of the ways to do it is to transform hue, saturation and value (HSV) data of an image into RGB color space, such that hue and value are assigned with the low frequency and high frequency information (Daily, 1983). The saturation can be fixed to some level. We used the green band of our orthophoto to transform into synthetic colors. Initially, a single band of an image is a gray scale image. High pass and low pass filter was used to extract the high and low frequency information

respectively. And the saturation was fixed to 0.5. SCT is useful to see the low frequency information present due to scattering of surface from vegetation that are dominated by high frequency information in an image. Figure 6(b) shows the orthophoto obtained after applying SCT.

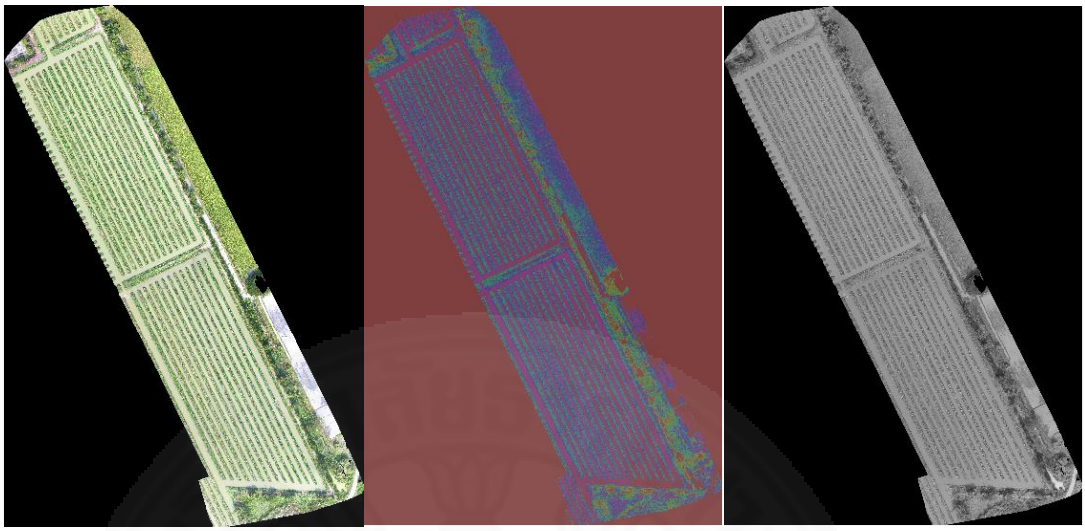
Triangular Greenness Index

Another method of image processing that we use is to calculate a Vegetation Index (VI), and to understand which VI to use, we need to keep in mind that we are using a digital camera with RGB bands, and near-infrared (NIR) band is not available in our images. On the other hand, the plant we are focusing on is made of large leaves of green color. On the other hand, abundant grass surrounding the banana plants in our images, makes it difficult for the computer vision techniques and human eyes to understand the difference between grass and banana leaves due to minimal difference in chlorophyll content of grass and banana leaves. We need a VI that is sensitive towards the change in the difference in leaf chlorophyll content of leaves in canopy scale, as the UAV flies in lower altitude during data collection. Triangular Greenness Index (TGI) was developed to solve this problem in (Hunt et al., 2011) for digital cameras with RGB bands, and hence is the best-fit VI for our purpose. TGI was developed based on equation below.

$$TGI = -0.5 * \{190(R - G) - 120(R - B)\}$$

TGI derives chlorophyll content in leaves in canopies by calculating the area of triangle formed by three points: (480nm, R_{480}), (550nm, R_{550}), and (670nm, R_{670}). In short, it uses spectral reflectance of 670, 550 and 480 nanometer, which are the wavelength of colors Red (λ_{670}), Green (λ_{550}) and Blue (λ_{480}) respectively. The green leaves of banana plant have significantly high chlorophyll content, and similarly the greenish-yellow and yellow-greenish leaves (mature banana plant leaves) have lower chlorophyll (Vergeiner, Banala, & Kräutler, 2013). The banana plants in our field had green leaves because they were only 3 months old. Therefore, TGI brightens and highlights the pixels with green banana leaves in an aerial image, from the background, as they have high chlorophyll

content. After calculation of TGI, the RGB image input gets converted into an image of grayscale color, as seen in figure 6(c).



(a) (b) (c)

Figure 3.5 Orthophoto of farm after (a) LCS (b) SCT and (c) TGI

3.2.2.3 Image Preparation and Tiling

The training and ROI area is now separated from each D_1, D_2 and D_3 as shown in figure 7. Our ROI has 2695 banana plants in ground truth (GT) that needs to be counted.

Figure 3.6 Study area separated into training region and ROI (top right).

After separating ROI, image formats are converted to decrease the computational expense of work. The three variants of training area from three image processing methods $\{p_1, \ldots, p_i, \ldots, p_n\}$ are of tiff file format and are of same dimension as orthophoto i.e. 23655x34390. On one hand, tiff images are difficult to decode, and on the other hand, larger images occupy heavy amount of memory and Graphic Processing Unit (GPU) of a computing machine, making DL computationally expensive. Therefore, we convert D_1, D_2 and D_3 into smaller tiles of Joint Photographic Experts Group (JPEG) format, to minimize the file size without loss in spectral data. We produced 2116 image tiles of 600x600 pixels for training area, out of which, only 458 tiles consisted of atleast a single banana plant. These 458 tiles were manually separated into two sets of images: *train tiles* (366) and *test tiles* (92). Similarly, 204 tiles of 600x600 pixels, of ROI (abbr. ROI tiles) were also produced to validate the final CNN model. The train tiles and test tiles can now be used to train parameter-optimized DL to produce frozen models, and the tiles from ROI can be used to obtain multiple detection results O_n for evaluation. A sample of image tiles of D_1, D_2 and D_3 are shown in figure 8(a), 8(b) and 8(c) respectively.

(a) (b) (c)

Figure 3.7 Sample image tile of (a) LCS (b) SCT and (c) TGI

3.2.2.4 Train images using parameter-optimized CNN model

This step deals with the detection of banana plants using DL model, and is a crucial step in our approach. The set of image tiles produced for each variant of training area are now separately used to train DL model to produce different sets of outputs O_1, O_2 and O_3 for our ROI. For detailed clarification, we break the tasks performed to use DL on our image dataset, into two sub-steps below.

Image Annotation

The image tiles obtained in previous step are manually annotated by drawing rectangles to sufficiently surround the objects. Total number of 7212 samples of banana plants were annotated for the train and test tiles, out of which 5770 training samples and 1442 test samples were produced from 366 and 92 train and test tiles, respectively. These training and test samples were then forwarded to train a CNN model mentioned below.

CNN Training and Parameter Optimization

Now we use the annotated images consisting of training and test samples of our training area to train an object detection method that is based on CNN architecture. Out of many available models, we chose Faster-RCNN (Girshick et al., 2014) for our purpose as it is in state-of-the-art performance. Faster-RCNN works in two components: a Region

Proposal Network (RPN) that proposes region of object location on the images and a CNN network to classify objects in those proposed regions. An input image is first processed by Faster-RCNN with a feature extractor to produce feature maps. A feature map is a network of CNN consisting of a convolution layer and a pooling layer. The feature map is then passed to RPN for region proposal. Unlike the original Faster-RCNN that used CNN models like ZFNet (Zeiler & Fergus, 2014) and VGG-16 (Simonyan & Zisserman, 2014) as feature extractor, we used Inception-v2 (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016) model. The RPN then scans the feature maps by sliding windows to check whether an object (banana) exists or not. As the sliding windows are of fixed-size, multiple anchor boxes of varying scales and aspect ratios are used, to deal with dissimilarity in the shape of objects. We used 12 anchor boxes of 4 different scales and 3 different aspect ratios for our purpose. By default, RPN proposes multiple regions for same object, and they become superfluous. Therefore, the number of regions were limited to 300 using non-maximum suppression. The variation in sizes of these regions were set to fixed-size using a Pooling Layer, and then passed to a fully connected layer with SOFTMAX score converter for the calculation of bounding box regression loss and object classification loss, as originally suggested by (Girshick et al., 2014).

We used Tensorflow Object Detection API (Abadi et al., 2016) - a DL framework in Python language - to implement the Faster-RCNN. We took the pre-trained model of Inception-v2 trained in Common Objects in Context (COCO) (Lin et al., 2014) image dataset, and optimized the configuration parameters mentioned in previous paragraph, despite the unavailability of literature on parameter-optimization for banana plants. Also, data augmentation methods like random horizontal and vertical flip were activated to randomly augment the image tiles during the training, to provide variability in object samples. Data augmentation provides variability during the learning for DL models, and parameter optimization increases the performance of DL, thus increasing the number of true positives (TP), while decreasing false positives (FP) and false negatives (FN) of the objects detected.

During the training, loss function for bounding box regression and object classification was carefully monitored, and the training was continued until the loss decreased to 0.007, 0.025 and 0.037 for variants D_1, D_2 and D_3 respectively. Stopping the training at minimum loss, the models were frozen at the corresponding iteration step, and saved as frozen detection models m_1, m_2 and m_3 for each D_i. These models can be now used to detect banana plants on image tiles of corresponding variants of ROI, to obtain multiple detection outputs O_n.

3.2.2.5 Production of multiple detection results

The models m_n are now used to produce detection results O_1, O_2 and O_3 over the three sets of ROI tiles for D_1, D_2 and D_3 respectively. While running each m_i into ROI tiles, intersection over union (IoU) threshold was set to be 0.5, instead of default and generally used value of 0.8. IoU measures the accuracy of detection by calculating the ratio of area of overlap between two bounding boxes to the area of union. IoU threshold ranges from 0 to 1, and decreasing it results higher recall and lower precision in detection, and vice-versa. It makes sense to lower this threshold at this step, because in the next step, we merge the detected multiple bounding box into a single box anyways.

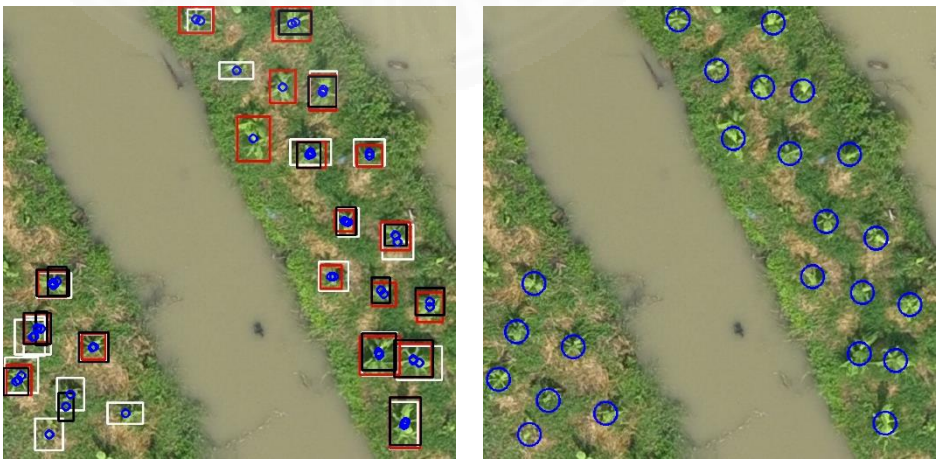
Figure 9(a), 9(b) and 9(c) shows a sample of O_1, O_2 and O_3, where it can be seen that 15, 18 and 19 out of 24 plants are respectively detected. Some immature small plants, plants that are closely occluded, and the plants at edge of image tiles are not detected. There also exists a problem of multiple detection of same object in some tiles, as seen in figure 9(a) and 9(b), which is due to the lowering of IoU threshold. This problem is solved while merging the results in next step.

(a) (b) (c)

Figure 3.8 Detection results on variants of image processing methods on a sample image tile of (a) LCS (b) SCT and (c) TGI

3.2.2.6 Merge the detection results

Now we combine the results of O_1, O_2 and O_3 to increase the count of banana plants and to solve the problem of multiple detection of same plant as seen in the sample before, in three steps. The first step is to collect the bounding box of all results and overlay into one set of image tiles. The second step is to calculate centroids of all the bounding boxes. And thirdly, merge the centroids that are clustered within some threshold of *Euclidean Distance*. This threshold is estimated by taking a value less than the euclidean distance between the closest plants. For our case, we used 30 pixels of threshold. Figure 10(a) shows the result of first and second step for two sample image tiles, and figure 10(b) shows the merged output (O) obtained after third step of combination for our sample tile.



(a) (b)

Figure 3.9 Merging detection results from variants of image processing methods. Demonstration of (a) Bounding boxes from LCS (Black), SCT (White) and TGI (Red), and their centroids (small blue circles) on a sample image tile of ROI. (b) Merged centroids by taking Euclidean distance threshold of 30 pixels are represented by blue circles of 15px radius.

It can be seen that, after merging the results, total detection has increased to 23 out of 24 plants, in our sample tile. This is because of decreased false negatives (FN) due to combination of multiple detection results. It can be seen in figure 10(a) that some of the plants are detected by only a one variant of image processing method. The merged detection results in sample image tiles in Fig figure 10(b) shows that the detection of plants can be increased by enhancing the vegetative property of aerial image, using methods like contrast stretch, vegetation index and HSV color transform.

Figure 10(b) is a sample of detection result obtained by the application of overall algorithm on orthophoto (I) of 40m altitude. In the next chapter, we show our final detection results on whole ROI, evaluate the accuracy in detection of banana plants, and improve the detection at the edges of the image tiles by combining the detection results obtained from multiple flight altitudes.

3.3 Open-source UAV Image Processing for Crop Health Monitoring

For our demonstration of algorithm of open-source UAV image processing web-service, we collected aerial images of a tropical farm in Thailand (N 14.25 E100.89, decimal degree-WGS84) consists of Banana (genus *Musa*) plantation with flight plan performed using DJI Ground Station Pro.

We assume a web-service with some user-friendly interface, with back-end setup on a host server with inter-connected segments of computer units, forming multiple processing nodes. A processing node is a segment obtained after division of a computer's physical units including processor, memory and storage. These nodes are setup with the dependencies of our algorithm to solve the problems stated in section 1.2. Web-interface

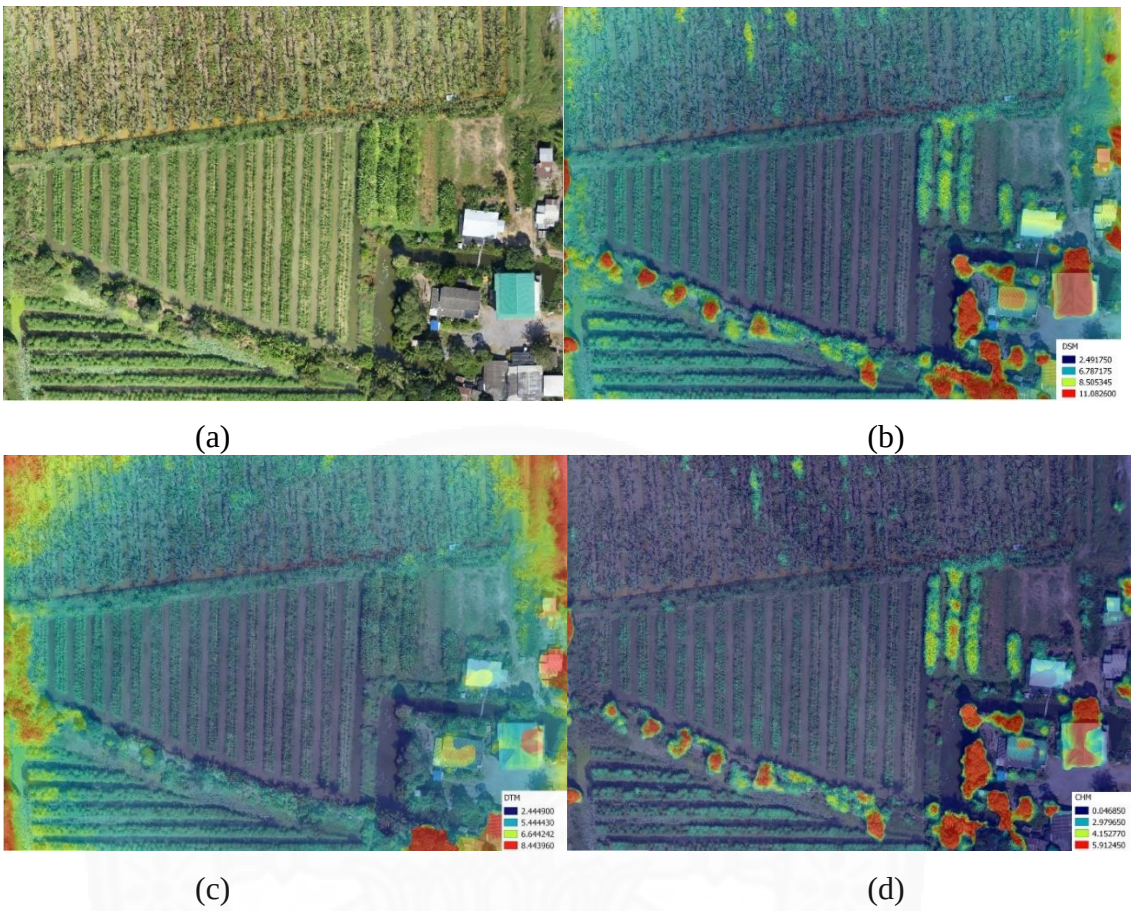
allows to upload the UAV-collected aerial images, which will be passed to a processing node. ODM processes the images inside node using OpenSFM algorithm, to generate point clouds and mesh, as mentioned before. From the point clouds and mesh, ODM further generates orthophoto (problem 1) and DEMs of two types: DSM and DTM. Using these outputs of ODM, we calculate CHM (problem 2) and Vegetation Indices (problem 3). Finally, the processing speed and quality is fine-tuned (problem 4) by changing some parameters of ODM. This section proceeds further to provide step-wise solution to each problem. We have hosted a web-interface implementation of our algorithm at http://drone.paantee.com.

3.3.1 Generation of an Orthophoto

The solution to our first problem is to generate an orthophoto, for which ODM's API is invoked to create a token to process our images. The generated token provides project ID and task ID inside ODM's framework. The images are uploaded to database, and using the project ID and task ID, they are provided to ODM to produce orthophoto, DSM and DTM which are shown in figure 11(a), 11(b) and 11(c) respectively. DSM and DTM provided by ODM are in single gray-scale tiff file format. For demonstration, we used linear color interpolation to prepare a colored classification, and overlaid them over orthophoto with 50% transparency. The value of elevation increases from low to high in 11(b) and 11(c).

3.3.2 Generation of Canopy Height Model

The DSM and DTM generated by ODM are then used to solve our second problem of generating CHM, which is simply the difference between DSM and DTM. An open-source library called Geospatial Data Abstraction Library (GDAL) (Warmerdam, 2008) is used to subtract the DTM from DSM. The CHM of our study area is demonstrated in figure 11(d), where it is overlaid over orthophoto with transparency for better visualization, and again the value of elevation increases from low to high.



(a) (b)

(c) (d)

Figure 3.10 (a) Orthophoto, (b) DSM, (c) DTM and (d) CHM of the study area.

3.3.3 Generation of Vegetation Indices

Vegetation Indices (VI) are quantitative measurements indicating the health of vegetation (Campbell & Wynne, 2011). These indices give a numerical value to the intensity of spectral response to any vegetation area. Their significance were noted with remark ever since the first earth resources satellite was launched in 1972. Since then, multiple such VI were formulated to study and monitor crop health, which can be found in (Bannari et al., 1995; Xue & Su, 2017). Furthermore, these VIs can be differentiated from unique relations they provide between spectral response and vegetation area. Therefore, they are highly dependent on the number of spectral bands in the images.

A commonly used VI is Normalized Difference Vegetation Index (NDVI), which relies on infrared (IR) band of aerial image. But in this research, we only study the VI

calculated in visible spectrum (i.e. red, green and blue bands). Besides a VI that uses infrared band, we allow the users to get three common VI that can be obtained from only RGB band of image. The significance of providing such VI is that most of the UAVs are not available with camera sensors having multispectral or hyperspectral bands including infrared, and integrating such sensor comes with an increase in expense. Those three VIs are explained below.

Visible Atmospherically Resistant Index (VARI)

Visible Atmospherically Resistant Index (VARI) uses contrast between 700 nano-meter (green), 470-490nm (blue) and 660-680nm (red) bands of RGB color image, which was proposed in (Gitelson, Kaufman, Stark, & Rundquist, 2002), where it showed more robustness in estimating VI of corn and wheat with minimal atmospheric effects. VARI provides good estimation of Vegetation Fraction (VF) because it is less sensitive towards atmospheric effects.

$$VARI = \frac{G - R}{G + R - B}$$

Triangular Greenness Index (TGI)

Triangular Greenness Index (TGI) detects nitrogen requirement, by studying leaf chlorophyll content of a crop, and is often preferable in low-altitude aerial images collected from low-cost cameras (Hunt Jr et al., 2013). It uses spectral reflectance of 670, 550 and 480 nm in aerial images.

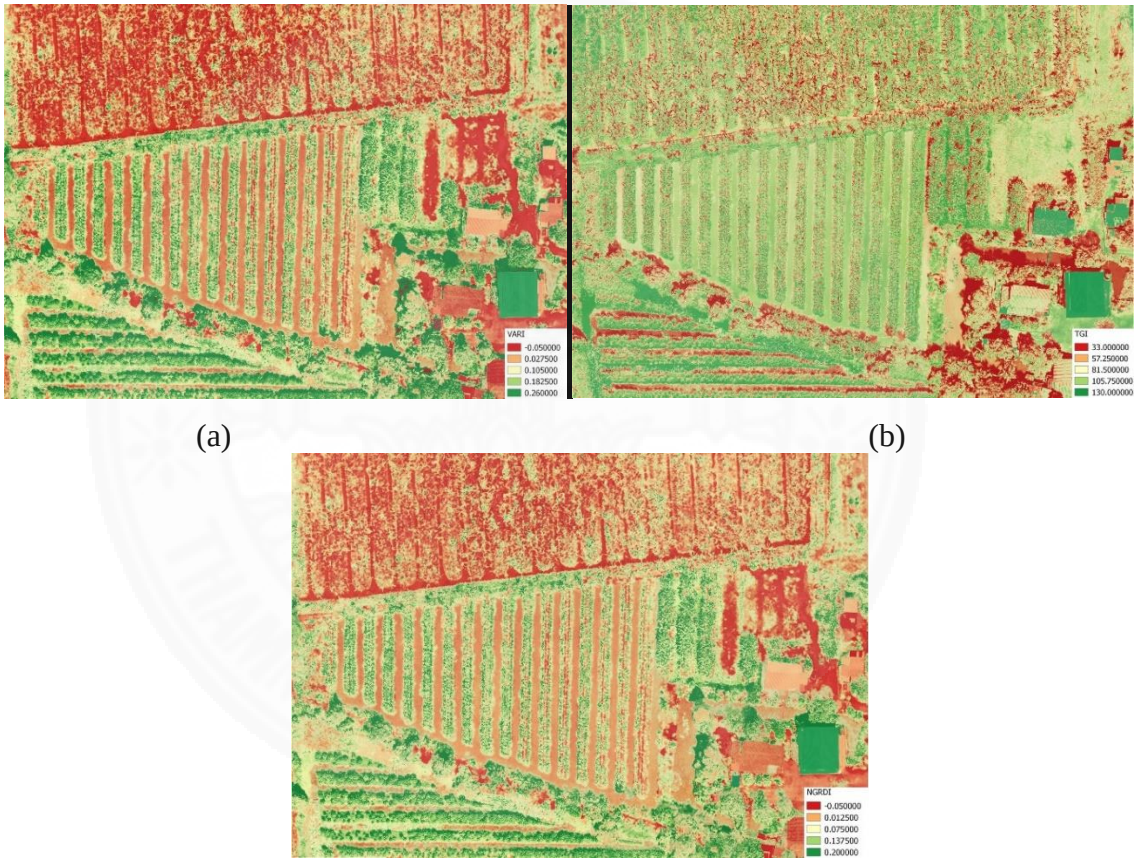
$$TGI = -0.5 * \{190(R - G) - 120(R - B)\}$$

Normalized Green Red Difference Index (NGRDI)

Normalized Green Red Difference Index (NGRDI) is another VI that uses only red and green spectral reflectance of image and was first studied in (Tucker, 1979). NGRDI is regarded as an indicator for phenology, and allows biomass estimation.

$$NGRDI = \frac{G - R}{G + R}$$

The solution to our third problem is to generate the three VIs mentioned above. The orthophoto generated is used to calculate the VIs for our RGB orthophoto. The three equations shown above are used to calculate VARI, TGI and NGRDI respectively. For demonstration, we show the calculated VARI, TGI and NGRDI indices in figure 12(a), 12(b) and 12(c) respectively, for the orthophoto of our banana farm. The higher value in the indices visualizes healthier plants in figure 12(a) and 12(c), and the plants with their leaves with higher chlorophyll content in figure 12(b).



(a) (b)

(c)

Figure 3.11 (a) VARI, (b) TGI and (c) NGRDI of study area.

3.3.4 Fine-tuning image processing speed and quality

The fine-tuning of image processing speed and quality of outputs can be done by collecting some prior information from users, before uploading the images to server. The

users are first asked to select the assets they want among: orthophoto, DSM, DTM, CHM, VARI, TGI and NGRDI. There are many run-time parameters that ODM allows developers to change (See https://github.com/OpenDroneMap/OpenDroneMap/wiki/Run-Time-Parameters for detail). The selections are then passed to ODM to generate DSM and DTM, if the users provides that they want them. Deselection of DSM or DTM will skip the generation of CHM using GDAL. The vegetation index maps are generated by using the orthophoto generated by ODM.

After selecting the assets, the users are asked to state how fast they want the assets, and in what resolution the assets needs to be processed. For this, we provide three options: 'Low', 'Medium' and 'High'. If user wants the processing to be done faster and cares less about resolution, 'Low' run-time parameters are used. Similarly, 'Medium' is for higher resolution assets that take longer processing time, and 'High' is for the ones who wants even higher resolution, and cares less about processing time. The parameters in ODM that we change for fine-tuning speed and quality of ODM outputs are shown in table 2, which are adjusted seamlessly, and the users need not have their understanding.

The parameter *min-num-features* changes minimum number of features to extract per image, *depthmap-resolution* changes the density of point cloud and *mesh-octree-depth* increases vertices in mesh reconstruction. Parameters *orthophoto-resolution* and *dem-resolution* changes resolution of orthophoto and DEMs respectively in centimeters, and are by default 5cm. The resolution however depends on the resolution of raw images provided by users, and also the type of mesh used during image processing. ODM has an option *use-3dmesh*, which asks to use 3D mesh to compute orthophoto instead of 2.5D. But this parameter did not produce good results for our sample of orthophoto, which is why, for now, we rely on 2.5D mesh. For option 'Low', we enable ODM's parameter to process orthophoto directly from sparse reconstruction, skipping dense reconstruction. This minimizes the processing time as it also skips 3D model generation while processing.

Table 3.2 Run-time parameters of ODM under fine-tune options

S.N.	General Parameters	Low	Medium	High
1	min-num-features	4000	8000	10000
2	orthophoto-resolution	10	5	5

3	use-3dmesh	Disabled	Enable	Enable
4	depthmap-resolution	640	640	1000
5	mesh-octree-depth	8	9	12
6	dem-resolution	10	5	5
7	fast-orthophoto	Enable	Disabled	Disabled
8	texturing-nadir-weight	16	24	32

The processing time depends on the number and resolution of raw images, and the assets that users want. We processed 122 aerial images of our farm using ODM, in the same machine with 3.30GHz four-core CPU and 32 gigabytes of memory. ODM took 42, 76 and 257 minutes to process the 122 images, for our parameter-optimization options of ‘Low’, ‘Medium’ and ‘High’ respectively. We evaluate the quality of assets provided by ODM in next chapter.

CHAPTER 4
EVALUATION AND DISCUSSION

4.1 Weather Scenario Generation Game

In this section, we demonstrate the results obtained on solving the three problems that are sought for solution regarding: (1.) Determination of most successful scenario, (2.) Classification of scenarios into fresh and stale and (3.) Generation of new scenarios from fresh scenarios. The design method was formulated previously on chapter 3, where WSG was formulated as a solution to the three problems, ultimately aiming to produce better weather scenarios for crop modelling. Now, we show the results to the three problems as binary and real-value game on rainfall (RAIN) and temperature (TMAX/TAVG) variable.

4.1.1 Binary Game on RAIN

Problem1: The top most scenario at each round is the most successful scenario at that round. Such that, the top most scenario at each round, as seen in figure 13, is the scenario with maximum success rate.

Problem2: As the weather data is mostly contradicting the real event values, scenario of such player *B* provides a reference scenario by the end of each round, which will allow us to separate fresh and stale scenarios. The determined scenario in figure 13 could not approximate the maximal success rate among all OI players because the game played for 156 rounds did not have player *B* with winning time n_B, after which it continuously wins. Suppose there was such a player whose winning time was round 20, then the bMI's prediction would have generated the success rate as shown in figure 14, where the player *B* is just an assumption and is not obtained from DISAWGS data.

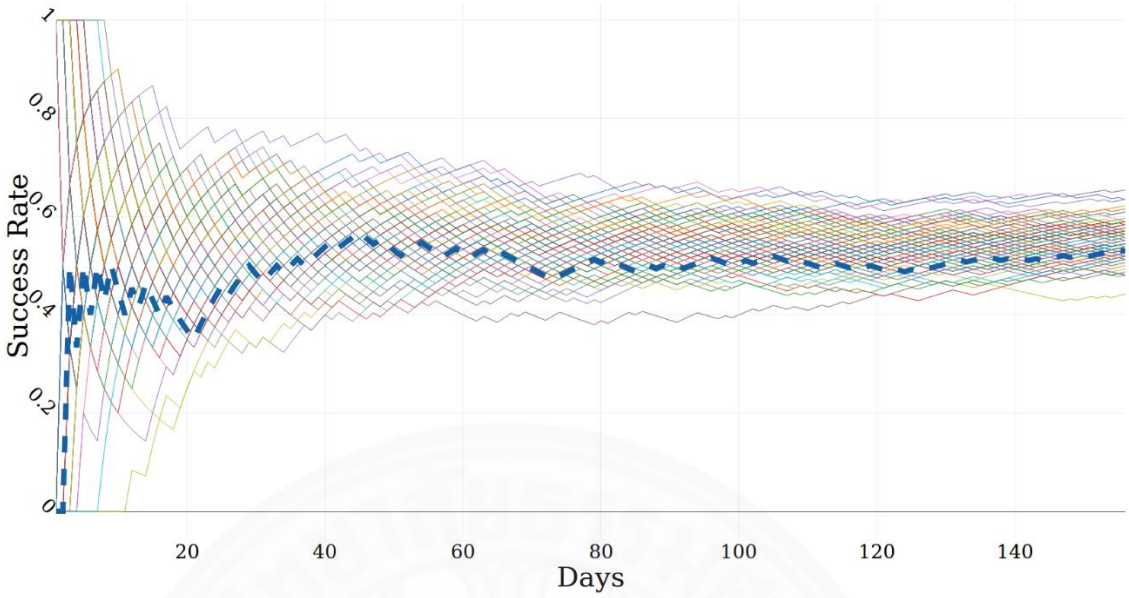


Figure 4.1 The bMI's optimal scenario on binary RAIN data of DISAWGS

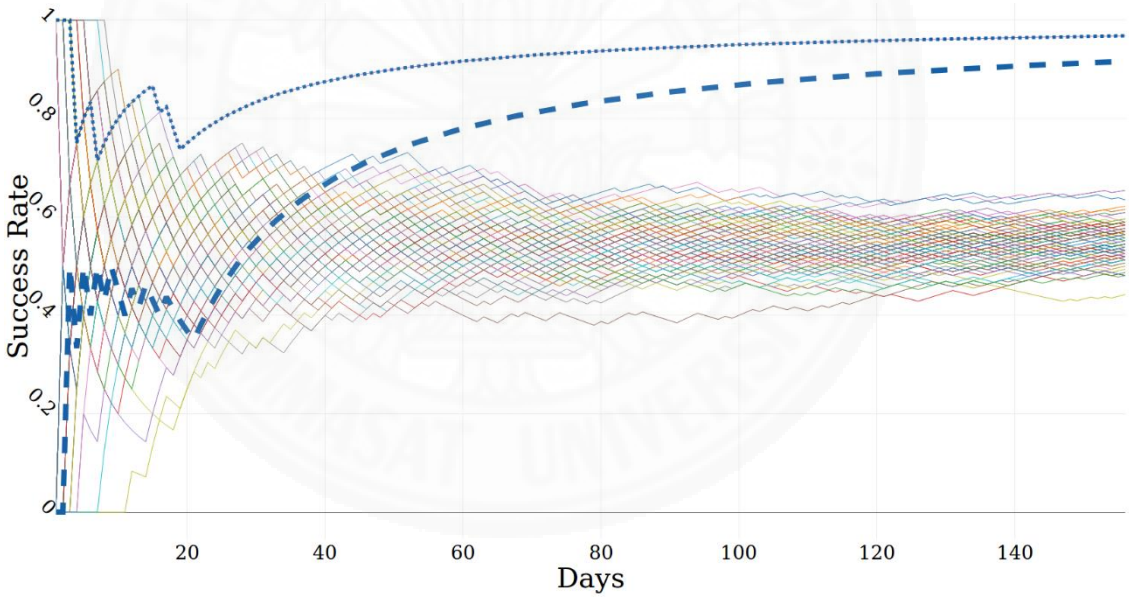


Figure 4.2 The bMI's optimal scenario (dotted line) on binary RAIN data with an assumption of a player B (dashed line) with winning time of n_B=20.

Problem3: Let us call the prediction game played for classification of fresh scenarios as "General Game", and the game repeated only on fresh scenarios as new *OI*

player, as “Sub-game”. The success rate of the meta-player *bMI* in general and sub-game on all fresh scenarios of RAIN scenarios are shown in table 3 It shows that sub-game on all fresh scenarios have made no change to newly generated scenario for our sample run. The reason is that all the scenarios *bMI* followed in general game are included in the set of fresh scenarios. This proves that there is no effects of stale scenarios on the new optimal scenario of *bMI*, and using only fresh scenarios is enough to ensure the same result. Instead of using all fresh scenarios in sub-game, another approach is to take only a desired number of them who have relatively high success rate in general game. The desired number depends on the number of original scenarios and it being constant, number of fresh ones can vary at each round. The performance of new scenarios obtained from sub-game of *15* and *2* fresh scenarios from the binary RAIN data of DISAWGS are also shown in table 3.

Table 4.1 Success rate of bMI on general and sub-game of binary RAIN data.

Round	**General Game**	**Sub-game**		
		all	15	2
1	0.000	0.000	0.000	0.000
40	0.539	0.539	0.615	0.744
80	0.506	0.506	0.595	0.633
120	0.496	0.496	0.563	0.563
156	0.529	0.529	0.561	0.574

The success rate of newly generated scenarios from *15* and *2* fresh scenarios, in our sample run, are seen to be *0.561* and *0.574* respectively in the last round which is better than the scenario generated from all fresh scenarios. Also short-run regret for fewer fresh scenarios seems better. This is because, selecting less fresh scenarios accumulates less regrets for *bMI*. However, this does not always mean that taking less fresh scenarios increases the performance of *bMI* because success rate depends on how the scenarios are performing at each round. But if there exists non-deceiving object-level players, like fresh scenarios, the meta-player's success approximates to the one with high success rate in long run. Now that the scenarios have been classified into ready-to-use desired number of fresh

scenarios, they can be used by weather generators to produce better scenarios for other variables that are key input for crop modeling systems.

4.1.2 Real-value Game

Figure 15 and 16 shows the success rate of *RW* on randomly prepared real-value TAVG data, among all 1000 scenarios in the general game and sub-game among fresh scenarios respectively. It can be seen that majority of players are above the performance of *MI* in figure 15. Depending on loss of the players, the meta-player could either perform better or fail to do so. Unlike *bMI*, this game does not depend on one best player, but performs better if the majority of players are performing well. *RW* collapses to zero success along with all the players who are providing false predictions. After the classification of players in the general game of *RW* on real-value TAVG data, a sub-game is run among only the *fresh players* till 1000th round and the success rate of *RW*'s predictions have dominated that of all the other players as seen in figure 16.

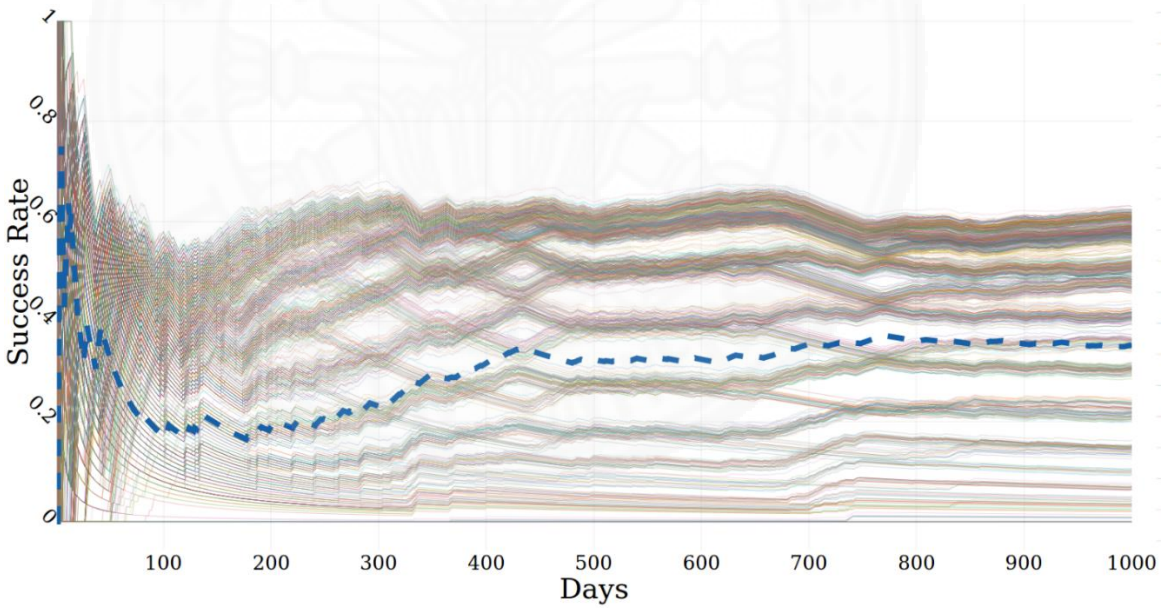


Figure 4.3 General game of RW on all scenarios of real-value TAVG data

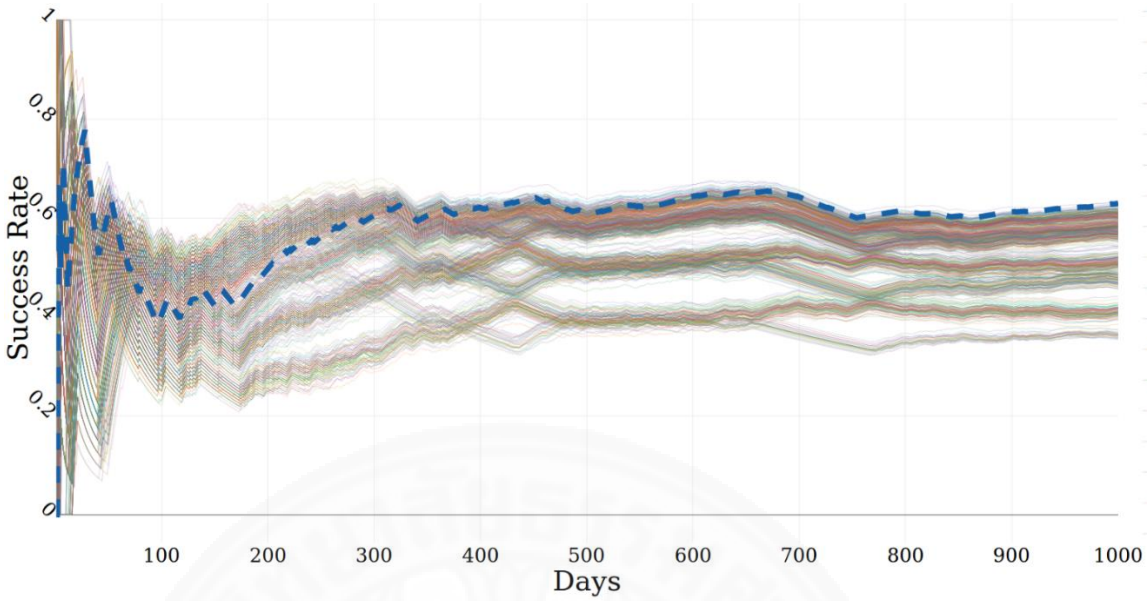


Figure 4.4 Sub-game of RW on fresh scenarios of real-value TAVG data

Among *1000* TAVG scenarios, *736* outperformed *RW*, to become fresh scenarios. table 4 shows the general game compared with sub-game of all *736*, *15* and *2* fresh scenarios, where the corresponding meta-players of sub-game compete each other. *RW* 's success rate have approximated the most successful fresh player in the sub-games for our less oscillated sample data and have proved that the classification of weather scenarios into fresh and stale not only out-performs the general game of WSG but also provides promising weather scenarios for crop simulation applications.

Table 4.2 Success rate of RW on general and sub-game of real-value TAVG data

Round	**General Game**	**Sub-game**		
		all	**15**	**2**
1	0.000	0.000	0.000	0.000
200	1.191	0.493	0.497	0.558
400	0.318	0.619	0.622	0.632
600	0.326	0.643	0.644	0.658
800	0.362	0.610	0.611	0.611
1000	0.353	0.629	0.630	0.622

Similar to the game on TMAX or TAVG, WSG can also be played on TMIN, SRAD and real-value RAIN, to obtain desired number of complete weather scenarios. But the real-value game of these variables being primarily dependent on dry and wet values of rain, binary game of RAIN of can be used to segregate rainfall scenarios provided by rainfall models in stochastic weather generators. So by integrating binary WSG on rainfall models and real-value WSG on the weather generator output of four variables, we can classify the weather scenarios into fresh and stale. Thus obtained desired number of fresh scenarios will result believable crop simulation outputs.

4.2 Deep Learning based Banana Plant Detection and Counting using UAV collected RGB Images

In this section, we evaluate quantitative and qualitative performance of our algorithm in detection of banana plants on our ROI. Previously, we only showed the results on some sample of ROI tiles that was obtained from the orthophoto of images taken from 40m altitude. Here, we combine the ROI tiles to get the detection over whole area of ROI. Then we investigate the performance of detection results on the variants provided by image processing methods, and also discuss results on the variants provided by different flight altitudes of UAV during data collection.

The overall algorithm explained in previous section was run over all ROI tiles, and then merged to get the final detection over the whole area of ROI, taken from the orthophoto generated from the images taken from 40m of flight altitude. This result can be seen in figure 17, where 2472 out of 2695 plants were correctly detected (yellow markers represents banana plants). It can be seen that smaller plants and the plants that were missed in ROI tiles being at the edge of the image are the ones undetected in the final merged ROI. This result can be further improved if the banana plants at the edges of image tiles are also detected, which is done later in this section 4.1.

Figure 4.5 Final detection on ROI image taken from 40m flight altitude.

To calculate the overall accuracy of our algorithm in ROI, we calculate precision and recall of banana plant detection to compare against ground truth. The equations below shows the formula for precision, recall and overall accuracy, where a correct detection is a case where a bounding box obtained from detection result sufficiently or insufficiently surrounds a single banana plant. We use these formulas to discuss the difference in performance due to different image processing methods, and also discuss the performance of DL models trained on the image collected from varying flying altitudes.

$$Precision = \frac{Total\ number\ of\ correct\ detection\ of\ plants}{Total\ number\ of\ all\ detected\ objects}$$

$$Recall = \frac{Total\ number\ of\ correct\ detection\ of\ plants}{Total\ number\ of\ plants\ in\ ground\ truth}$$

$$Overall\ Accuracy = \frac{Precision + Recall}{2}$$

4.2.1 Performance of image processing methods

The overall idea of algorithm that we have created to increase the performance of banana plant detection using multiple variants of image processing methods and a DL architecture, was composed after looking at the performance results on normal RGB image. The detection results obtained on ROI image of 40m altitude, which in total has 2695 banana plants in ground truth, is shown in tabulated in table 5. The initial results of running the algorithm only on RGB images gave the recall of 72.8%, even though the IOU threshold in CNN model was fixed to be 0.5. But again, the precision was of 99.8% even with the IOU threshold decreased from normal value of 0.8 to 0.5. This gave us motive to use the same CNN model with same configuration to train on the image variants produced from multiple image processing methods, and as seen in the results below, other image variants except RGB have shown significantly better performance.

Table 4.3 Detection performance of Faster-RCNN model on variants of ROI taken from 40m flight altitude

Image Processing Method	**Correct Detection**	**All detected Objects**	**Recall**	**Precision**	**Overall Accuracy**
Normal RGB	1962	1965	0.728	0.998	0.863
LCS	2136	2166	0.793	0.986	0.889
SCT	2221	2235	0.824	0.994	0.909
TGI	2285	2285	0.848	1.000	0.924
Combined	2472	2486	0.917	0.994	0.956

The LCS method stretched the contrast of images, and also increased recall to 79.3% by detecting 2136 plants correctly, also detecting smaller plants that were missed in the normal RGB image. This is due to the stretch of GL value range in the images. The bright pixels became vibrantly brighter and the dark pixels even darker, separating pixels covering grass from the pixels covering bright banana leaves. The variant of SCT made 2221 correct detection, with recall value 82.4%. This variant also could detect plants that were missed by LCS, with higher performance in detection. This is because SCT emphasizes the low-frequency information that are present due to surface scattering difference from vegetation on the ground. Furthermore, as the chlorophyll content of three

months old green banana leaves are brightened by TGI, 2285 plants were correctly detected, with recall of 84.8%.

After the combination of results of LCS, SCT and TGI, 2472 plants were correctly detected, making recall to be 91.7%. This increment in detection is achieved due to the combination of detection from all individual image processing methods. With few false detection, the precision in detection was affected for all image variants that were combined, and the overall accuracy for variants LCS, SCT and TGI were 88.9%, 90.9% and 92.4% respectively. The overall accuracy is thus, 95.6% for the ROI, due to less number of false detection. The recall value can be furthermore increased if the plants at the edge of the ROI tiles are detected. For this, we combine the final results of obtained from the ROI of multiple flight altitudes.

4.2.2 Performance in varying flight altitude of UAV

Previously, our algorithm for DL based banana plant detection and counting was demonstrated on the UAV based aerial images taken from 40m of flight altitude. In this section, we discuss on the performance of the same algorithm on aerial images collected from different flight altitude, in search of proper flight altitude for banana plant detection and counting, and to detect the plants undetected previously for 40m image tiles. For simplicity, we call the multiple variants of images due to flight altitude as *altitude variants*.

As the focal length of camera we used in our UAV was 4mm, photographic scale of images collected from 40, 50 and 60m altitude were therefore 1:10000, 1:12500 and 1:15000 respectively. To simplify, if a banana plant occupies $9m^2$ in ground truth, it occupies 0.09, 0.06 and 0.04 mm^2 on the altitude variants of 40, 50 and 60m respectively. So the object size gets smaller as flight altitude increases. The performance of object detection in CNN is directly affected by object size, and this can be seen in table 6 and 7, which shows the results of our algorithm on altitude variants of 50m and 60m respectively. Comparing to table 5, the recall value of 50m decreases to 84.3%, and further decreases to 73.7% for 60m. With small changes in precision, overall accuracy dropped to 91.5% and 86.2% for 50 and 60m respectively.

Table 4.4 Detection performance of Faster-RCNN model on variants of ROI taken from 50m flight altitude.

Image Processing Method	Correct Detection	All detected Objects	Recall	Precision	Overall Accuracy
Normal RGB	1779	1794	0.660	0.992	0.826
LCS	1846	1859	0.685	0.993	0.839
SCT	1931	1963	0.717	0.984	0.850
TGI	1912	1920	0.709	0.996	0.853
Combined	2271	2302	0.843	0.987	0.915

Table 4.5 Detection performance of Faster-RCNN model on variants of ROI taken from 60m flight altitude.

Image Processing Method	Correct Detection	All detected Objects	Recall	Precision	Overall Accuracy
Normal RGB	1130	1143	0.419	0.989	0.704
LCS	1300	1326	0.482	0.980	0.731
SCT	1543	1561	0.573	0.988	0.781
TGI	1444	1467	0.536	0.984	0.760
Combined	1987	2015	0.737	0.986	0.862

Close observation of results showed that with the increase in flight altitude, two plants that were close together were detected as a single plant in both altitude variants (50 and 60m) of ROI. This decreased the precision of detection for both of them, also lowering the overall accuracy. From table 5, 6 and 7, we can say that, lower the flight altitude gets, or higher the resolution of image, better is the detection of banana plants. For us, 40m is the best altitude for banana plant detection among the three. In the next section, we combine the detection results of our altitude variants.

4.2.3 Combining the results of multiple flight altitudes

The recall value of 40m altitude variant of ROI, which was 91.7%, dropped even more as the altitude increased to 50 and 60m. While merging the ROI tiles, some of the plants on the edges, and some plants that were significantly smaller than mature plants were not detected on individual altitude variants.

There can be various approach to detect the undetected plants on the edges of tiles of aerial images. One approach can be to implement sliding windows with strides to cut the tiles in first place, and merge the detection results using location of bounding boxes. This approach is necessary for satellite images, unless we can compute DL on large images. But the use of UAV for data collection allows us to collect multiple altitude variants, which can be separately trained to obtain multiple detection results. These results can be merged to detect the undetected plants in our case.

To increase recall, we combine the detection results of 40, 50 and 60m altitude variants. For this combination, we first linearly transform the scale of centroids of bounding boxes of lower altitude variants (40m and 50m) to the dimension size of higher (60m) variant. All centroids of bounding boxes from all altitude variants are collected and overlaid on ROI image of 60m. The closely clustered centroids are then merged using Euclidean distance threshold of 15 pixels, to avoid repeated detection.

The merged points, which are the final detection of location of banana plants, are then overlaid on ROI image of 60m in figure 18, where the location of detection are represented by markers. The yellow, red and black markers represents the correct, incorrect and missed detection respectively. The final tabulated result is summarized in table 8, which shows that out of 2695 plants in ROI, 2629 were correctly detected, making the recall of 97.6%. False detection from the results of higher altitude variants (especially 60m) accumulated and lowered the precision to 97.2%. Therefore, the overall accuracy is 97.4% after the combination of results of altitude variants.

Table 4.6 Detection performance of our algorithm after combining results from 40, 50 and 60m altitude.

Evaluation Index	**Value**
The number of banana plants in ground truth	2695
The number of correctly detected banana plants	2629
The number of all detected objects	2705
Recall	0.976
Precision	0.972
Overall Accuracy	0.974

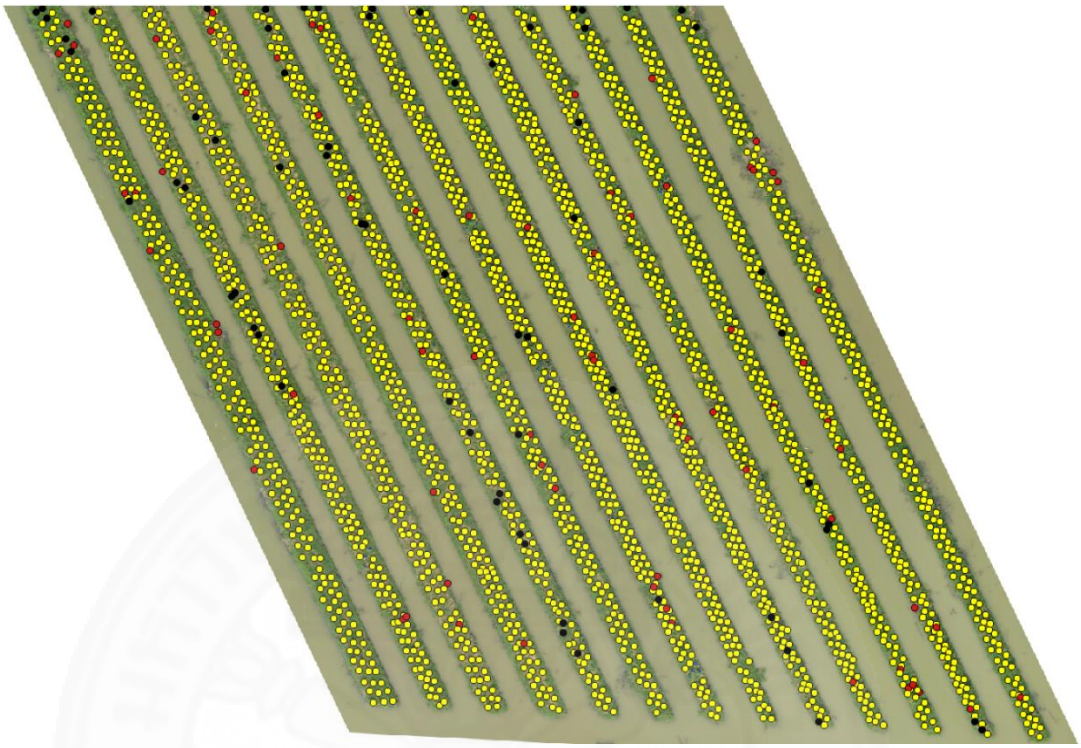


Figure 3.6 Final detection on ROI after combining results of 40, 50 and 60m altitude variants.

4.3 Open-source UAV Image Processing for Crop Health Monitoring

In this section, we evaluate our open-source UAV image processing web-service to Pix4dmapper, which is a closed-source commercial software commonly used in literature, for UAV image processing. Pix4D provides a separate software called Pix4Dfields to generate VI maps. Distinctively, our algorithm uses only the open-source tools to generate the assets mentioned before.

Despite being under-development, ODM could generate orthophoto upto 5.87 centimeter per pixel (cm/px) resolution, which is close to 3.42 cm/px from state-of-the-art Pix4Dmapper. A closer view snippet of orthophoto generated by ODM with ‘Low’, ‘Medium’ and ‘High’ fine-tune options, and from Pix4Dmapper for same location are shown in figure 19(a), 19(b), 19(c) and 19(d) respectively. It can be seen that there still exists some no-data values in the snippet of Pix4Dmapper. The ODM outputs have larger

cell-size, but still can clearly visualize the banana plants in canopy scale, and most importantly, 5.87 cm/px is enough resolution to calculate VI's.

(a) (b) (c) (d)

Figure 3.7 Closer view of orthophoto generated by (a) Low, (b) Medium (c) High and (d) Pix4Dmapper for banana farm.

We also compare the histogram of orthophoto obtained from all fine-tune options to those of Pix4D in table 9. It shows that the output of all options are correlated by more than 97.95% to the orthophoto produced by Pix4D. The VI's calculated from these orthophoto will therefore be similar to that obtained from Pix4D. The DSM from ODM however falls behind in terms of resolution, affecting the quality of CHM. We expect that, with the undergoing collaborative contributions of developers of ODM, the resolution of DEMs will be improved in future.

Table 4.7 Histogram comparison of orthophoto obtained from Low, Medium, High and Pix4D

Histogram Comparison	**low**	**medium**	**high**	**Pix4d**
Count	22891416	22338195	22732604	64936640
Mean	97.301	103.989	103.086	106.928
Std. Dev.	61.051	58.512	45.673	56.535
Red (mean)	102.652	110.042	109.037	113.345
Green (mean)	110.967	118.33	117.245	121.358
Blue (mean)	78.284	83.597	82.976	86.082
Correlation (to pix4d)	0.9795	0.9926	0.9883	1

CHAPTER 5
CONCLUSION AND FUTURE PROSPECTS

This thesis organizes three different works relating emerging advancements on machine learning, deep learning, weather generators, web-services, remote sensing, UAV photogrammetry and digital image processing into the crucial cores of sustainable agriculture visions. The three research works are:

1. Weather Scenario Generation Game
2. Deep learning-based banana plant detection and counting using UAV collected RGB images
3. An open-source UAV image processing web-service for crop health monitoring

The major conclusions, critical discussions and further recommendations derived from the three research are given in this section.

The first research develops Weather Scenario Generation Game (WSG), which is essentially a Prediction Game in which each object level players represents a weather scenario. WSG provides solutions to all three problems stated in problem statement, thus determining the most successful scenario, classifying the outdated scenarios into fresh and stale scenarios and generating new scenarios from fresh scenarios, for crop simulation applications. It doubtfully questions the current approach of generating weather scenarios, and the current approach of learning. With the application of WSG in the four variables (RAIN, TMIN, TMAX, SRAD) and also including humidity, wind, sunshine, we can also derive potential evapotranspiration for other application like climatic assessment of agriculture and water system management. Also extending the applicability in spatial dimension, regional climate and agricultural forecast can be studied with best forecasts of multiple spatial location. The possibilities in other applications is yet to be studied and could be the follow-up of the WSG into any other sectors. Moreover, WSG will provide decision makers with better seasonal weather forecasts in different applications. And with the integration of WSG on web-services of weather generators, a believable web-service of crop modeling system can be developed.

The second research developed an algorithm to precisely detect and count the number of banana plants in a farm, by combining detection results of applying a parameter-optimized CNN model on variants of UAV collected RGB images, using multiple image processing methods. Further we combined the detection outputs on variants of flight altitudes to increase the accuracy of detection. Finally, using the proposed algorithm, 97.6% of banana plants in our region of interest were correctly detected. With the precision of 97.2%, the overall accuracy of detection increased upto 97.4%. This algorithm can be further used in other farms to detect and count banana plants, and the performance of detection can be increased even more by training the CNN model in larger dataset, which will be our future work. Also an alternative solution to combine the detection results and an algorithm to run the DL models on large images with less computational expenses should be sought out for remote sensing applications. This research will help the farmers to estimate their production of banana, and also help to map the location of farm that needs to be re-cultivated, furthermore increasing the production.

The third research provided an algorithm for an open-source web-service to could solve the problems of generating orthophoto, CHM and VIs, for the images collected from normal RGB camera. A web-interface have been hosted and we expect it to be continued as a web-service. The VIs and CHM are calculated using the outputs generated by an open-source service that is under improvement by community contributors. Also, allowing the users to decide the optimization of their work, with mediocre knowledge of image processing, will increase the usability of such precision farming technology, which is a major challenge in many countries. The evaluation of the generated outputs to those obtained from state-of-the-art commercial software, showed small difference in resolution, and in some aspects, are even better. Improvement of open-source service like ODM promises to improve the quality of our algorithm, furthermore in future.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning. In *12th ${$USENIX$}$ Symposium on Operating Systems Design and Implementation (${$OSDI$}$ 16)* (pp. 265–283).

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., & Szeliski, R. (2009). Building rome in a day. In *2009 IEEE 12th international conference on computer vision* (pp. 72–79).

Alexandridis, T. K., Andrianopoulos, A., Galanis, G., Kalopesa, E., Dimitrakos, A., Katsogiannos, F., & Zalidis, G. (2018). An integrated approach to promote precision farming as a measure towards reduced-input agriculture in northern Greece using a spatial decision support system.

Allahyari, M. S., Mohammadzadeh, M., & Nastis, S. A. (2016). Agricultural experts' attitude towards precision agriculture: Evidence from Guilan Agricultural Organization, Northern Iran. *Information Processing in Agriculture*, *3*(3), 183–189.

Amara, J., Bouaziz, B., Algergawy, A., & others. (2017). A Deep Learning-based Approach for Banana Leaf Diseases Classification. In *BTW (Workshops)* (pp. 79–88).

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.

Baethgen, W. E., Carriquiry, M., & Ropelewski, C. (2009). Tilting the odds in maize yields: how climate information can help manage risks. *Bulletin of the American Meteorological Society*, *90*(2), 179–184.

Bannari, A., Morin, D., Bonn, F., & Huete, A. R. (1995). A review of vegetation indices. *Remote Sensing Reviews*, *13*(1–2), 95–120.

Bannayan, M., & Hoogenboom, G. (2008). Predicting realizations of daily weather data for climate forecasts using the non-parametric nearest-neighbour re-sampling technique. *International Journal of Climatology*, *28*(10), 1357–1368.

Bastiaanssen, W. G. M., Molden, D. J., & Makin, I. W. (2000). Remote sensing for

irrigated agriculture: examples from research and possible applications. *Agricultural Water Management*, *46*(2), 137–155.

Beersma, J. J., & Buishand, T. A. (2003). Multi-site simulation of daily precipitation and temperature conditional on the atmospheric circulation. *Climate Research*, *25*(2), 121–133.

Brandtberg, T. (1999). Automatic individual tree based analysis of high spatial resolution aerial images on naturally regenerated boreal forests. *Canadian Journal of Forest Research*, *29*(10), 1464–1478.

Buishand, T. A., & Brandsma, T. (2001). Multisite simulation of daily precipitation and temperature in the Rhine basin by nearest-neighbor resampling. *Water Resources Research*, *37*(11), 2761–2776.

Campbell, J. B., & Wynne, R. H. (2011). *Introduction to remote sensing*. Guilford Press.

Carlet, J., & Abayowa, B. (2017). Fast vehicle detection in aerial imagery. *ArXiv Preprint ArXiv:1709.08666*.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chinnachodteeranun, R., Hung, N. D., Honda, K., Ines, A. V. M., & Han, E. (2016). Designing and Implementing Weather Generators as Web Services. *Future Internet*, *8*(4), 55.

Colomina, I., & Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *92*, 79–97.

Council, N. R., & others. (1999). *Our common journey: a transition toward sustainability*. National Academies Press.

Daily, M. (1983). Hue-saturation-intensity split-spectrum processing of Seasat radar imagery. *Photogrammetric Engineering and Remote Sensing*, *49*, 349–355.

Dale, J. L. (1987). Banana bunchy top: An economically important tropical plant virus disease. *Advances in Virus Research*, *33*, 301–326.

Dare, P. M. (2005). Shadow analysis in high-resolution satellite imagery of urban areas.

Photogrammetric Engineering & Remote Sensing, *71*(2), 169–177.

Dubrovsk\`y, M., Žalud, Z., & Št'astná, M. (2000). Sensitivity of CERES-Maize yields to statistical structure of daily weather series. *Climatic Change*, *46*(4), 447–472.

Eisenbeiß, H. (2009). *UAV photogrammetry*. ETH Zurich.

Enciso, J., Maeda, M., Landivar, J., Avila, C., Jung, J., & Chang, A. (2016). Unmanned Aerial System (UAS) for Precision Agriculture and Management Decisions. In *2016 ASABE Annual International Meeting* (p. 1).

Everaerts, J. (2008). Unmanned aerial vehicles for photogrammetry and remote sensing. In *Advances in Photogrammtery, Remote Sensing and Spatial Information sciences: 2008 ISPRS Congress Book. Eds. Li, Chen y Baltsavias* (pp. 117–126).

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Gallington, R. W., Berman, H., Entzminger, J., Francis, M. S., Palmore, P., & Stratakes, J. (1997). Unmanned aerial vehicles. *Future Aeronautical and Space Systems(A 97-26201 06-31), Reston, VA, American Institute of Aeronautics and Astronautics, Inc.(Progress in Astronautics and Aeronautics.*, *172*, 251–295.

Gavrila, D. M., & Philomin, V. (1999). Real-time object detection for" smart" vehicles. In *Proceedings of the Seventh IEEE International Conference on Computer Vision* (Vol. 1, pp. 87–93).

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. https://doi.org/10.3390/rs61110335

Geng, S., de Vries, F. W. T. P., & Supit, I. (1986). A simple method for generating daily rainfall data. *Agricultural and Forest Meteorology*, *36*(4), 363–376.

Ghazal, M., Al Khalil, Y., & Hajjdiab, H. (2015). UAV-based remote sensing for vegetation cover estimation using NDVI imagery and level sets method. In *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*

(pp. 332–337).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).

Gitelson, A. A., Kaufman, Y. J., Stark, R., & Rundquist, D. (2002). Novel algorithms for remote estimation of vegetation fraction. *Remote Sensing of Environment*, *80*(1), 76–87.

Gougeon, F. A., Leckie, D. G., & others. (1998). Forest regeneration: Individual tree crown detection techniques for density and stocking assessment. In *Proceedings of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry* (pp. 10–12).

Grenzdörffer, G. J. (2014). Crop height determination with UAS point clouds. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Han, E., & Ines, A. V. M. (2017). Downscaling probabilistic seasonal climate forecasts for decision support in agriculture: A comparison of parametric and non-parametric approach. *Climate Risk Management*, *18*, 51–65.

Hansen, J. W., Challinor, A., Ines, A., Wheeler, T., & Moron, V. (2006). Translating climate forecasts into agricultural terms: advances and challenges. *Climate Research*, *33*(1), 27–41.

Hansen, J. W., & Ines, A. V. M. (2005). Stochastic disaggregation of monthly rainfall data for crop simulation studies. *Agricultural and Forest Meteorology*, *131*(3–4), 233–246.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Honkavaara, E., Kaivosoja, J., Mäkynen, J., Pellikka, I., Pesonen, L., Saari, H., … others. (2012). Hyperspectral reflectance signatures and point clouds for precision agriculture

by light weight UAV imaging system. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, *7*, 353–358.

Honkavaara, E., Saari, H., Kaivosoja, J., Pölönen, I., Hakala, T., Litkey, P., … Pesonen, L. (2013). Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture. *Remote Sensing*, *5*(10), 5006–5039.

Huang, H., Deng, J., Lan, Y., Yang, A., Deng, X., & Zhang, L. (2018). A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery. *PloS One*, *13*(4), e0196302.

Hunt, E. R., Cavigelli, M., Daughtry, C. S. T., 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.

Hunt, E. R., Daughtry, C. S. T., Eitel, J. U. H., & Long, D. S. (2011). Remote sensing leaf chlorophyll content using a visible band index. *Agronomy Journal*, *103*(4), 1090–1099.

Hunt, E. R., Hively, W. D., Fujikawa, S., Linden, D., Daughtry, C. S., & McCarty, G. (2010). Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sensing*, *2*(1), 290–305.

Hunt Jr, E. R., Doraiswamy, P. C., McMurtrey, J. E., Daughtry, C. S. T., Perry, E. M., & Akhmedov, B. (2013). A visible band index for remote sensing leaf chlorophyll content at the canopy scale. *International Journal of Applied Earth Observation and Geoinformation*, *21*, 103–112.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Jensen, J. R., & Lulla, K. (1987). Introductory digital image processing: a remote sensing perspective.

Jeong, J., Park, H., & Kwak, N. (2017). Enhancement of SSD by concatenating feature maps for object detection. *ArXiv Preprint ArXiv:1705.09587*.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T.

(2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675–678).

Johansen, K., Sohlbach, M., Sullivan, B., Stringer, S., Peasley, D., & Phinn, S. (2014). Mapping banana plants from high spatial resolution orthophotos to facilitate plant health assessment. *Remote Sensing*, *6*(9), 8261–8286.

Jones, C., & Pimdee, P. (2018). Innovative ideas: Thailand 4.0 and the fourth industrial revolution. *JOURNAL OF ETHNIC MINORITIES RESEARCH*, (22).

Jones, J. W., Hoogenboom, G., Porter, C. H., Boote, K. J., Batchelor, W. D., Hunt, L. A., … Ritchie, J. T. (2003). The DSSAT cropping system model. *European Journal of Agronomy*, *18*(3–4), 235–265.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90.

Ke, Y., & Quackenbush, L. J. (2011). A review of methods for automatic individual tree-crown detection and delineation from passive remote sensing. *International Journal of Remote Sensing*, *32*(17), 4725–4747.

Kilsby, C. G., Jones, P. D., Burton, A., Ford, A. C., Fowler, H. J., Harpham, C., … Wilby, R. L. (2007). A daily weather generator for use in climate change studies. *Environmental Modelling & Software*, *22*(12), 1705–1719.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, *14*(5), 778–782.

Laliberte, A. S., Goforth, M. A., Steele, C. M., & Rango, A. (2011). Multispectral remote sensing from unmanned aircraft: Image processing workflows and applications for rangeland environments. *Remote Sensing*, *3*(11), 2529–2551.

Lall, U., & Sharma, A. (1996). A nearest neighbor bootstrap for time series resampling. *Water Resour. Res*, *32*(3), 679–693.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

LeCun, Y., Bengio, Y., & others. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Li, W., Fu, H., Yu, L., & Cracknell, A. (2016). Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, *9*(1), 22.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Mearns, L. O., Rosenzweig, C., & Goldberg, R. (1997). Mean and variance change in climate scenarios: methods, agricultural applications, and measures of uncertainty. *Climatic Change*, *35*(4), 367–396.

Mehrotra, R., Srikanthan, R., & Sharma, A. (2006). A comparison of three stochastic multi-site precipitation occurrence generators. *Journal of Hydrology*, *331*(1–2), 280–292.

Mohan, M., Silva, C. A., Klauberg, C., Jat, P., Catts, G., Cardil, A., … Dia, M. (2017). Individual tree detection from unmanned aerial vehicle (UAV) derived canopy height model in an open canopy mixed conifer forest. *Forests*, *8*(9), 340.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, *7*, 1419.

Mortensen, A. K., Dyrmann, M., Karstoft, H., Jørgensen, R. N., Gislum, R., & others. (2016). Semantic segmentation of mixed crops using deep convolutional neural network. In *CIGR-AgEng Conference, 26-29 June 2016, Aarhus, Denmark. Abstracts and Full papers* (pp. 1–6).

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.

Nex, F., & Remondino, F. (2014). UAV for 3D mapping applications: a review. *Applied Geomatics*, *6*(1), 1–15.

Nijland, W., De Jong, R., De Jong, S. M., Wulder, M. A., Bater, C. W., & Coops, N. C. (2014). Monitoring plant condition and phenology using infrared sensitive consumer grade digital cameras. *Agricultural and Forest Meteorology*, *184*, 98–106.

Org, O. (n.d.). 2018. *OpenDroneMap: Open Source Toolkit for Processing Aerial Drone Imagery*.

Pan, S. J., Yang, Q., & others. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Perrone, M. P., & Cooper, L. N. (1995). When networks disagree: Ensemble methods for hybrid neural networks. In *How We Learn; How We Remember: Toward an Understanding of Brain and Neural Systems: Selected Papers of Leon N Cooper* (pp. 342–358). World Scientific.

Pinz, A. (1998). Tree isolation and species classification. In *Proceedings of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry, Victoria, BC* (pp. 127–139).

Pix4D, S. A. (2014). Pix4dmapper.

Pollock, R. (1998). Individual tree recognition based on a synthetic tree crown image model. In *Proc. of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry* (pp. 25–34).

Puttemans, S., Van Beeck, K., & Goedemé, T. (2018). Comparing Boosted Cascades to Deep Learning Architectures for Fast and Robust Coconut Tree Detection in Aerial Images. In *VISIGRAPP (5: VISAPP)* (pp. 230–241).

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.

Rajagopalan, B., & Lall, U. (1999). A k-nearest-neighbor simulator for daily precipitation and other weather variables. *Water Resources Research*, *35*(10), 3089–3101.

Rebetez, J., Satizábal, H. F., Mota, M., Noll, D., Büchi, L., Wendling, M., … Burgos, S.

(2016). Augmenting a convolutional neural network with local histograms—a case study in crop classification from high-resolution UAV imagery. In *European Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 515–520).

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263–7271).

Remondino, F. (2003). From point cloud to surface: the modeling and visualization problem. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *34*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Reyes, A. K., Caicedo, J. C., & Camargo, J. E. (2015). Fine-tuning Deep Convolutional Networks for Plant Recognition. In *CLEF (Working Notes)*.

Richardson, C. W., & Wright, D. A. (1984). WGEN: A model for generating daily weather variables.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1–2), 1–39.

Rosnell, T., & Honkavaara, E. (2012). Point cloud generation from aerial image data acquired by a quadrocopter type micro unmanned aerial vehicle and a digital still camera. *Sensors*, *12*(1), 453–480.

Schowengerdt, R. A. (2006). *Remote sensing: models and methods for image processing*. Elsevier.

Schurz, G. (2008). The meta-inductivist's winning strategy in the prediction game: A new approach to Hume's problem. *Philosophy of Science*, *75*(3), 278–305.

Schurz, G. (2012). Meta-induction in epistemic networks and the social spread of

knowledge. *Episteme*, *9*(2), 151–170.

Schurz, G., & Thorn, P. (2017). A Priori Advantages of Meta-Induction and the No Free Lunch Theorem: A Contradiction? In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)* (pp. 236–248).

Semenov, M. A., & Barrow, E. M. (1997). Use of a stochastic weather generator in the development of climate change scenarios. *Climatic Change*, *35*(4), 397–414.

Shavlik, J. W., Dietterich, T., & Dietterich, T. G. (1990). *Readings in machine learning*. Morgan Kaufmann.

Shaw, R., Lark, R. M., Williams, A. P., Chadwick, D. R., & Jones, D. L. (2016). Characterising the within-field scale spatial variation of nitrogen in a grassland soil to inform the efficient design of in-situ nitrogen sensor networks for precision agriculture. *Agriculture, Ecosystems & Environment*, *230*, 294–306.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Singh, K. D., Sohlberg, S., Sokolov, V. E., & others. (1986). Conceptual framework for the selection of appropriate remote sensing techniques.

Smith, I. (2017). Making successful drone maps: Photogrammetry is more art than science. Drone Deploy.

Snavely, N. (2008). Bundler: Structure from motion (sfm) for unordered image collections. *Http://Phototour. Cs. Washington. Edu/Bundler/*.

Soltani, A., & Hoogenboom, G. (2003). A statistical comparison of the stochastic weather generators WGEN and SIMMETEO. *Climate Research*, *24*(3), 215–230.

Sona, G., Pinto, L., Pagliari, D., Passoni, D., & Gini, R. (2014). Experimental analysis of different software packages for orientation and digital surface modelling from UAV images. *Earth Science Informatics*, *7*(2), 97–107.

Srestasathiern, P., & Rakwatin, P. (2014). Oil palm tree detection with high resolution multi-spectral satellite imagery. *Remote Sensing*, *6*(10), 9749–9774.

STEWART, L., CAMPAGNOLO, D., DANIELLS, J., LEMIN, C., GOEBEL, R., PINESE, B., … others. (1998). Tropical banana information kit. *Nambour:*

Queensland Department of Primary Industries.

Sujana, S. R., Abisheck, S. S., Ahmed, A. T., & Chandran, K. R. S. (2017). Real time object identification using deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)* (pp. 1801–1805).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Thorn, P. D., & Schurz, G. (2012). Meta-induction and the wisdom of crowds. *Analyse & Kritik*, *34*(2), 339–366.

Trnka, M., Dubrovsk\`y, M., Semerádová, D., & Žalud, Z. (2004). Projections of uncertainties in climate change scenarios into expected winter wheat yields. *Theoretical and Applied Climatology*, *77*(3–4), 229–249.

Tubiello, F. N., Donatelli, M., Rosenzweig, C., & Stockle, C. O. (2000). Effects of climate change and elevated CO2 on cropping systems: model predictions at two Italian locations. *European Journal of Agronomy*, *13*(2–3), 179–189.

Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, *8*(2), 127–150.

Turner, D., Lucieer, A., & Watson, C. (2012). An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (UAV) imagery, based on structure from motion (SfM) point clouds. *Remote Sensing*, *4*(5), 1392–1410.

Turner, D. W. (1972). Banana plant growth. 1. Gross morphology. *Australian Journal of Experimental Agriculture*, *12*(55), 209–215.

Vergeiner, C., Banala, S., & Kräutler, B. (2013). Chlorophyll breakdown in senescent banana leaves: catabolism reprogrammed for biosynthesis of persistent blue fluorescent tetrapyrroles. *Chemistry--A European Journal*, *19*(37), 12294–12305.

Viola, P., Jones, M., & others. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, *1*, 511–518.

Wallis, T. W. R., & Griffiths, J. F. (1995). An assessment of the weather generator

(WXGEN) used in the erosion/productivity impact calculator (EPIC). *Agricultural and Forest Meteorology*, *73*(1–2), 115–133.

Warmerdam, F. (2008). The geospatial data abstraction library. In *Open source approaches in spatial data handling* (pp. 87–104). Springer.

Weber, E. U., & Stern, P. C. (2011). Public understanding of climate change in the United States. *American Psychologist*, *66*(4), 315.

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). 'Structure-from-Motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, *179*, 300–314.

Wijitdechakul, J., Sasaki, S., Kiyoki, Y., & Koopipat, C. (2016). UAV-based multispectral image analysis system with semantic computing for agricultural health conditions monitoring and real-time management. In *2016 International Electronics Symposium (IES)* (pp. 459–464).

Wilks, D. S., & Wilby, R. L. (1999). The weather generation game: a review of stochastic weather models. *Progress in Physical Geography*, *23*(3), 329–357.

Wu, C. (2013). Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013* (pp. 127–134).

Wu, C., & others. (2011). VisualSFM: A visual structure from motion system.

Xue, J., & Su, B. (2017). Significant remote sensing vegetation indices: A review of developments and applications. *Journal of Sensors*, *2017*.

Yang, C.-C. (2006). Image enhancement by modified contrast-stretching manipulation. *Optics & Laser Technology*, *38*(3), 196–201.

Yates, D., Gangopadhyay, S., Rajagopalan, B., & Strzepek, K. (2003). A technique for generating regional climate scenarios using a nearest-neighbor algorithm. *Water Resources Research*, *39*(7).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833).

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

Zhang, W., & Montgomery, D. R. (1994). Digital elevation model grid size, landscape representation, and hydrologic simulations. *Water Resources Research*, *30*(4), 1019–1028.

Zigadlo, J. P., Holden, C. L., Schrader, M. E., & Vogel, R. M. (2001). Electronic color infrared camera. Google Patents.

BIOGRAPHY

Name	Mr. Bipul Neupane
Date of Birth	March 29, 1993
Education	2015: Bachelor of Engineering (Geomatics Engineering) Kathmandu University
	2019: Master of Science (Engineering and Technology) Sirindhorn International Institute of Technology Thammasat University

Publications

Neupane, B., Hung, N.D., & Horanont, T. (2018). Weather Scenario Generation Game, *The 13th International Conference on Knowledge, Information and Creativity Support Systems*, 153-158.