

Continuous Monitoring of High Risk Disaster Areas by Applying Change Detection to Free Satellite Imagery

Allison G. Roush, 2d Lt, USAF

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Kevin B. Kochersberger, Chair

A. Lynn Abbott

Rolf Mueller

May 6, 2024

Blacksburg, Virginia

Keywords: Disaster Monitoring, Satellite Imagery, Change Detection

Disclaimer: The views expressed in this thesis are those of the author and do not reflect
the official policy or position of the United States Air Force, Department of Defense, or the
U.S. Government.

Continuous Monitoring of High Risk Disaster Areas by Applying Change Detection to Free Satellite Imagery

Allison G. Roush, 2d Lt, USAF

(ABSTRACT)

Natural disasters can happen anywhere causing damage to land and infrastructure. When these disasters occur in remote areas without much human traffic, it may take a long time for someone to notice that an event has occurred and to respond to it. Response time and damages could be reduced if the area could be remotely monitored. Many satellites pass over the Earth everyday collecting valuable imagery data that is free to access. However, this data can be difficult to process and use in practical applications such as monitoring an area for changes. Existing programs that use satellite imagery to monitor areas for changes can cost a significant amount of money making it inaccessible to most people. In this paper, a software program is introduced to automatically retrieve, process, and analyze free satellite imagery data and notify the user of significant changes in their area of interest (AOI). First, a software program was developed to automatically download a package of satellite imagery data from Planet Labs that met certain requirements for AOI, date, and cloud cover. A second software program was developed to download this data from the Google Cloud Storage (GCS) space and compare a current image to the composite of previous images in order to detect a change. This program then creates a figure to display the current image, the previous image, the difference area, and a summary table of the difference metrics. This figure is saved and emailed to the user if the differences are greater than the set threshold. This program is also capable of running automatically in the background of a computer every time it is logged in. The success of the program in correctly identifying areas of

change was tested in three locations using historical satellite image data. The software was successful in identifying areas of change and delivering this information to the user in an easy to understand summary figure. Overall, the software was able to utilize free satellite imagery to detect changes in disaster areas and deliver a summary report to a user to take action showing that this software could be used in the future as an easy way to monitor disaster areas.

Continuous Monitoring of High Risk Disaster Areas by Applying Change Detection to Free Satellite Imagery

Allison G. Roush, 2d Lt, USAF

(GENERAL AUDIENCE ABSTRACT)

Natural disasters can happen anywhere causing damage to land and infrastructure. When these disasters occur in remote areas without much human traffic, it may take a long time for someone to notice that an event has occurred and to respond to it. Response time and damages could be reduced if the area could be remotely monitored. Many satellites pass over the Earth everyday collecting valuable imagery data that is free to access. However, this data can be difficult to process and use in practical applications such as monitoring an area for changes. Existing programs that use satellite imagery to monitor areas for changes can cost a significant amount of money making it inaccessible to most people. In this paper, a software program is introduced to automatically retrieve, process, and analyze free satellite imagery data and notify the user of significant changes in their area of interest (AOI). First, a software program was developed to automatically download a package of satellite imagery data from Planet Labs that met certain requirements for AOI, date, and cloud cover. A second software program was developed to download this data from the Google Cloud Storage (GCS) space and compare a current image to the composite of previous images in order to detect a change. This program then creates a figure to display the current image, the previous image, the difference area, and a summary table of the difference metrics. This figure is saved and emailed to the user if the differences are greater than the set threshold. This program is also capable of running automatically in the background of a computer every time it is logged in. The success of the program in correctly identifying areas of

change was tested in three locations using historical satellite image data. The software was successful in identifying areas of change and delivering this information to the user in an easy to understand summary figure. Overall, the software was able to utilize free satellite imagery to detect changes in disaster areas and deliver a summary report to a user to take action showing that this software could be used in the future as an easy way to monitor disaster areas.

Dedication

To my friends and family, thank you for helping me make it through.

Acknowledgments

I would like to thank many people who supported me on the way to completing my master's. First, I'd like to thank my advisor Dr. Kochersberger for his great mentorship and guidance throughout this process. I would also like to thank Dr. Abbott and Dr. Mueller for serving on my committee. Thank you to the Air Force Institute of Technology and the DAWN-ED program for allowing me to pursue this degree. Thank you to the Det 875 personnel and the other officers stationed in Blacksburg through AFIT for keeping me engaged with the Air Force while serving an unconventional first assignment. Thank you to all of my friends for supporting me and helping me have an amazing college experience. A huge thank you to my amazing roommates Zymmorrah Myers, Brie Kosko, and Alex Barker for goofing off with me and encouraging me to work on my research. Finally, thank you to all of my family, I am so grateful for their continual support of all my endeavors.

Contents

List of Figures	x
1 Introduction	1
2 Review of Literature	4
2.1 Satellite Imagery Monitoring	4
2.2 Remote High Risk Disaster Areas	6
2.3 Change Detection	8
2.4 Corridor Detection	9
3 Software Design	12
3.1 Specifying the Area of Interest	12
3.1.1 Identifying a Corridor Using a Snaking Method	13
3.1.2 Specifying a Corridor Using Coordinates	16
3.2 Creating a Request for Satellite Images	17
3.2.1 Creating Image Filters	18
3.2.2 Data Delivery	19
3.2.3 Submitting the Request	19
3.3 Analyzing the Satellite Images	20

3.3.1	Downloading Most Recent Images	20
3.3.2	Addressing the Cloud Problem	22
3.3.3	Creating the Baseline Variance	24
3.3.4	Comparing the Current Image	28
3.3.5	Sending the Results	31
3.4	Automation for Continuous Monitoring	33
4	Results	36
4.1	Simple Tests for Processes	36
4.2	Change Detection Results	38
4.2.1	Hurley VA Flood Example	39
4.2.2	Wheeling WV Debris Flow Example	42
4.2.3	Baseline Example	45
4.2.4	HazMapper Comparison	48
5	Summary & Conclusions	52
Bibliography		57
Appendices		63
Appendix A GitHub Link for Code		64

List of Figures

3.1	Gray-scale satellite image of pipeline trail compared to the black and white image that resulted from applying a contrast threshold.	13
3.2	Contrast threshold image with region areas less than 20% of the greatest area removed.	14
3.3	Using snakes to fill in gaps in a trail. a-c shows the snaking of the first two regions in the image, d-f shows the snaking of the combined first two regions with the addition of a third region.	15
3.4	RGB satellite image compared to the same RGB satellite image with the detected path highlighted in blue.	16
3.5	Screenshot of geoson.io [1] being used to create an AOI polygon covering Lane Stadium.	17
3.6	Raw RGB satellite image data vs a visualized version of the data in which the RGB bands have been normalized.	22
3.7	Satellite image of a residential area in Blacksburg, VA obscured by clouds and shadows (a) followed by a mask of the clear pixels in the image (b) and the image when multiplied by the clear mask (c).	24
3.8	Pseudo-code to describe how to determine the maximum pixel value.	25
3.9	Pseudo-code to describe how to determine the minimum pixel value.	26

3.10 Maximum difference of a raw RGB satellite image vs the maximum difference of a visualized version of the data in which the RGB bands have been normalized.	27
3.11 Pseudo-code to describe how to determine the variable threshold.	28
3.12 Difference mask of Minden, IA after a tornado. White areas indicate a significant change, black areas indicate no change, and gray areas indicate that a comparison could not be made.	30
3.13 Summary figure of Minden, IA following a tornado strike depicting the comparison image, the current image, the tri-color difference, the RGB filled difference, and a summary table of the difference statistics.	32
3.14 Pseudo-code to describe how to determine if a significant change has occurred.	32
3.15 Screenshot of email generated notifying the user that a significant change was detected in Minden, IA following a tornado strike.	33
3.16 Screenshots of Windows Task Scheduler being used to specify triggers and actions to configure the analyzing satellite imagery python script to run autonomously.	35
4.1 Screenshot of geojson.io [1] in use to select a complicated geometry following the Cascades hiking trail.	37
4.2 Screenshot of an email that was automatically generated at computer login by the Windows Task Scheduler running the test script.	38
4.3 Summary figure generated to explain the comparison conducted on the Hurley, VA images.	40

4.4 Ortho photo generated from UAV images taken in Hurley Virginia (photo copied from Whitehurst et al. [2]) and overlaid with a satellite image.	41
4.5 Difference mask overlaid with UAV ortho photo, areas outlined in red mark the location of sturdy structures.	42
4.6 Summary figure generated to explain the comparison conducted on the Wheeling, WV debris flow images.	44
4.7 Image of debris flow in Wheeling, WV [3].	45
4.8 Satellite image post debris flow with regions of change masked in black. The blue line marks the treeline, the orange line marks the road, and the red circle marks the main tree.	46
4.9 Summary figure generated to explain the comparison conducted on the baseline images from a field in Blacksburg, VA.	47
4.10 Comparison of visualized and raw satellite images for the baseline area. . . .	48
4.11 Screenshot of HazMapper with red box surrounding debris flow area in Breathitt County, KY [4].	49
4.12 Summary figure generated to explain the comparison conducted on the Breathitt County, KY images.	50
4.13 Results from HazMapper and the proposed difference analysis method over a debris flow in Breathitt County, KY. The red boxes indicate the debris flow area.	51
4.14 Overlay of HazMapper result and proposed difference analysis result on the Breathitt County, KY images, the red box indicates the debris flow area. . . .	51

List of Abbreviations

AOI Area of Interest

API Application Programming Interface

DEM Digital Elevation Model

GAN Generative Adversarial Network

GCS Google Cloud Storage

InSAR Interferometric Synthetic Aperture Radar

KLT Kanade-Lucas-Tomasi

LiDAR Light Detection and Ranging

NDVI Normalized Difference Vegetation Index

NDWI2 Normalized Difference Water Index 2

NIR Near Infrared

rdNDVI Relative Difference in the Normalized Difference Vegetation Index

RGB Red-Green-Blue

SAR Synthetic Aperture Radar

SfM Structure from Motion

UAV Uncrewed Aerial Vehicle

VLOS Visual Line of Sight

Chapter 1

Introduction

Climate change is leading to an increase in the prevalence of extreme hydrological events, floods and droughts, which in turn has also led to an increase in landslides and wild fires [5]. These events cost the U.S. billions of dollars each year [5]. Oil spills as well as illegal logging have also had severe negative environmental and economic impacts [6] [7]. Remote areas that are difficult to monitor can be seriously impacted by such events. These areas suffer from slow response after an event and no method of early detection.

Satellite imagery data has come a long way since the first satellite was launched in 1957 [8]. Now high resolution data that covers the Earth's entire land mass is available on a near daily basis [9]. This data could be used as an aid to monitor for changes due to disasters thanks to its wide availability. Processing this data, however, can be difficult for those inexperienced with image processing and coding.

Planet Labs was chosen as the source of imagery data in this paper for its high frequency coverage, high resolution, pre-processing abilities, and cost. Planet Labs provides near daily imagery of the entire land surface of the earth at 3m resolution as well as corrects images for geometric distortions and atmospheric conditions [9]. Additionally, they provide researchers with a free limited download quota [10]. All satellite images used in this paper were provided by Planet Labs [11].

This paper presents a system to automatically retrieve satellite imagery data for a specified area, process this data to detect changes in the environment, and finally notify the user when changes occur and present a summary of the findings. This system requires minimal time and experience to setup. After initial setup the system can run autonomously without any user inputs and will send an email notification to the user when significant changes have been detected in the specified region.

The system described in this paper could then be used in many applications to monitor regions for significant changes. This would be especially useful in remote areas that are at high risk of natural disaster. This monitoring could provide early warnings of small changes before catastrophic events occurred allowing for time to address the problem before it gets worse. It could also be used to assess the extent of damages after an event when the area is difficult to access or dangerous to enter.

While the capabilities of the system presented in this paper can be applied to monitor regions across the globe, case studies in the Appalachian regions of southwest Virginia, northern West Virginia, and eastern Kentucky were chosen to evaluate the system. Whitehurst [2] previously demonstrated the ability to use drone imagery after a flood event in Hurley, VA to update flood models. These updated flood models could then be used to evaluate the area for future flood risks and inform property development decision.

One advantage of drone-based imagery and other imagery captured by aircraft is that it can easily be used to create 3D reconstructions of an area using structure from motion (SfM) methods which results in a digital elevation model (DEM) which can then be used in a flood model. However, drone flight capabilities are limited by many factors including limited flight

time, visual line of sight (VLOS) restrictions, and a skilled operator required on site. With VLOS restrictions and limited flight time, imaging a 10km length of stream could take a crew of people one or two days to complete.

The method described in this paper allows for near daily monitoring of areas with no human input required, greatly reducing man hours when compared to monitoring with drone imagery and allows for more frequent updates . Although 3D reconstructions can not be created from satellite imagery to update flood models like in [2], a change analysis method is used to determine areas of damage post events and can be used to inform future construction and emergency response decisions.

The system's ability to monitor large areas for changes on a near daily basis across the globe without any user input after initial setup shows great potential for future use of the program to assist in the response to natural disasters, construction planning, and other monitoring scenarios. In particular, the ease of setup for a non-technical user makes this tool accessible to a wide population of people increasing potential for adoption in many use cases.

Chapter 2

Review of Literature

2.1 Satellite Imagery Monitoring

The first satellite, Sputnik, was sent into space in 1957, kicking off the space race between the U.S. and the USSR. Since then, satellites and their imaging technology has rapidly improved allowing for more frequent imagery at higher resolutions across the globe leading to advancements in meteorological monitoring, global land maps, and more [8]. Today various types of satellite imagery are used to monitor the Earth for natural disasters.

One important way satellite imagery has been used is in the detection of oil spills. One study investigated the use of synthetic aperture radar (SAR) images to identify oil spills in the ocean [12]. SAR is an active sensor that sends out microwaves which are reflected off of the target surface, the reflections are read by the sensor and reveal properties of the surface. Due to the tendency for oil spills to dampen capillary waves in the ocean the study was able to identify oil spills as darker spots in an otherwise bright ocean [12].

Another study attempted to detect the Deepwater Horizon oil spill using optical satellite imagery [13]. They visually inspected the images to look for differences to attempt to identify the spill; they also computed a pixel by pixel difference between the images to detect

the spill. Both methods were able to detect the spill [13]. Similarly, [14] used optical satellite image data to monitor the impact of oil spills on the surrounding environment. They calculated the normalized difference vegetation index (NDVI) to monitor the state of the vegetation as well as the normalized difference water index 2 (NDWI2) to monitor water content in reservoirs and related these values to the presence of oil in those areas [14].

Satellite imagery data has also been used in the area of landslide detection and prevention [15]. Optical as well as SAR imagery data can be used to identify areas where a landslide has occurred based on a machine learning classification of each pixel in the image [16]. In addition to optical and SAR data, satellite interferometric synthetic aperture radar (InSAR) can be a great tool for monitoring ground movement including landslides [17]. InSAR works by emitting radio waves which are reflected off of the ground and received by the satellite, the difference between the signals gives information on the ground displacement which can be shown in an image called an interferogram [17]. Bekaert et al. [18] successfully used InSAR satellite data, which was able to provide millimeter scale ground displacement, to monitor the movement of large, slow-moving landslides in Nepal.

The most relevant satellite monitoring platform to the research conducted in this paper is HazMapper. HazMapper is an open-source application used for mapping natural hazards globally through Google Earth Engine [4]. HazMapper detects natural hazards by computing the relative difference in the normalized difference vegetation index (rdNDVI). The rdNDVI is used to identify areas of vegetation loss which indicates that a natural disaster has occurred that removed the vegetation [4]. HazMapper was successfully used to identify locations of landslides and debris flows, confirmed by field inspections, post heavy rainfall in Kentucky [19]. While HazMapper has been shown to successfully identify natural disasters

from satellite imagery its functionality is limited. HazMapper's functionality is limited to a manual search of an AOI with specified pre and post event dates and can only be used to monitor areas of vegetation [4]. Additionally, maximum resolution is limited to 10m with a minimum re-imaging time of 5 days [4].

The examples listed above for satellite imagery monitoring offer great solutions to specific cases but face drawbacks when trying to apply them to a more complete and autonomous monitoring strategy. The data is often difficult to process, provides low resolution images, is available infrequently, or does not offer autonomous capabilities. Processing difficulty and automation can be addressed with the creation of the program discussed in this paper, additionally, careful selection of a data source can increase resolution and monitoring frequency.

2.2 Remote High Risk Disaster Areas

Climate change is leading to an increase in severity of extreme hydrological events, including an increase in precipitation and flooding as well as intense periods of drought [5]. Floods and droughts can also lead to other natural disasters such as landslides and wildfires. Major weather disasters cost the U.S. at least \$92.2 billion in 2023 [20], any monitoring that could aid in the prevention or response of these disasters would be incredibly valuable.

Many areas that are at high risk for natural disasters coincide with areas that are difficult to monitor or access by conventional ground methods due to steep and dangerous terrain or low vehicular access. One area that poses a great risk is buried pipelines through hilly terrain.

Buried pipelines often run through harsh environments with landslide potential [21]. When a landslide occurs the pipeline is subjected to excessive plastic deformation which causes the pipeline to experience increased loads as well as increased stress and strain which may lead to collapse or buckling [21]. These failures often result in a full rupture of the pipeline rather than minor holes and cracks leading to major environmental and service impacts [22]. Landslides and floods can also have a major impact on rural roadways, cutting transportation access for people and businesses, negatively impacting the local economy [23]. These pipeline ruptures and roadway impacts due to landslides and floods may be prevented if land movement in the surrounding area is detected early and counter measures are put in place, such as retaining walls, slope regrading, or drainage implementation [24].

Oil spills, especially on the shoreline, can have significant negative impacts on the ecosystem [6]. The U.S. shoreline alone is over 95 thousand miles long [25], not all of this shoreline is easily accessible, and human monitoring for oil spills over such a great area would be extremely time consuming. Another area that is very large and difficult to monitor by humans on the ground is forested areas. Illegal logging is one of the major causes of deforestation and can also cause drought, increased runoff, loss of soil fertility, as well as economic losses [7].

The events described above have the potential to cause serious harm to the environment, the economy, infrastructure, and even human life. The negative consequences of these events may be mitigated, at least in part, by the ability to continuously monitor those areas which at the present time is difficult to do by conventional ground methods.

2.3 Change Detection

One major challenge that arises from trying to monitor a specific area is determining where a change has occurred between two images, making sure to only detect changes deemed significant. Many methods exist with varying applications and levels of complexity, a few relevant change detection methods are discussed below.

Changes in Earth topography and ground movement can be monitored using InSAR interferograms [26]. The interferograms are images of the topography of the Earth, when images in a series are compared the difference in height at each geographic location is easy to discern. Another method to identify topographic changes is using light detection and ranging (LiDAR) sensors. LiDAR sensors determine the distance of objects by emitting a laser and measuring the return time, when an area is surveyed using LiDAR sensors a 3D point cloud can be generated giving a model of the area [27]. Multiple point clouds generated at different times can be aligned and differences in topography can be seen [27]. 3D point clouds can also be generated from uncrewed aerial vehicle (UAV) images and these point clouds can similarly be used to discern topographical changes by aligning the point clouds and taking the difference [28].

Motion in a series of images can be detected using the frame difference method in which two images are differenced and compared to a threshold, if the difference for a specific pixel exceeds the threshold it is marked as a motion pixel [29]. Another method for tracking motion through a series of images is the Kanade-Lucas-Tomasi (KLT) algorithm. The KLT algorithm first identifies trackable features in an image, such as locations of sharp contrast, like corners, it then looks for these same features in subsequent images and determines the

displacement of the feature by minimizing the sum of differences [30]. To use the KLT method only small changes can occur between images and several key features must exist in the image to track.

In a more general sense there are a series of steps that can be taken to determine regions of change between two images [31]. First, the two images must be oriented in the same way with stable background objects aligned within both images. Next, radiometric and intensity adjustments must be made to ensure that changes are not detected due to variations in illumination location or intensity. Once these two pre-processing steps have been taken the images are ready for change detection. The simplest method for change detection is a simple difference in which a new image is created by subtracting each pixel of the first image from the same pixel in the second image. The absolute value of each pixel in this difference image is then compared to a set threshold, if it exceeds this threshold a value of 1 is given to that pixel location in the mask image, if it is less than the threshold a value of 0 is given. This creates a simple black and white image mask that shows the regions of change. This is a simple method but, the threshold must be carefully chosen for each application [31].

2.4 Corridor Detection

A common problem in computer vision is identifying roadways, thus, researchers have explored many ways to tackle this issue. One study looked at mapping new roadways using satellite imagery by first aligning the satellite image geographically with an outdated map of roadways [32]. Next, new roads are identified by searching the image for lines connecting to current roads and labels these branches as roads as well, this works under the assumption

that all new roads will be connected to old roads [32]. While this is a good method it relies on having a base map of roads to work off of.

Another study worked to detect roads from aerial imagery using a histogram based thresholding method followed by detecting local line segments and finally used point clustering to classify the final road region [33]. This process even allowed for detection of desert roads which can be difficult to separate from the background; success, however, was more limited in images that contained multiple or curved roads [33]. Yet another study worked on road detection by using an artificial neural network to identify roads [34]. The inputs to the network included spectral values from the image as well as normalized distances of pixels in the surrounding window which increased accuracy as the roads were all homogeneous areas in the images [34].

Many of the methods for road detection rely on the uniformity of the road, its sharp contrast to the background area, and its generally straight nature. To detect nature trails or other paths through the woods from from imagery, slightly different methods are required since these trails do not have all the same attributes as a conventional paved road. One method that has been used to help detect more contoured edges are snakes, a type of active contour model.

Snakes are energy minimizing splines that can be influenced by features in an image, pulling it towards edges or other specified features [35]. A variation of the original snakes, Ziplock snakes, were developed allowing for a more precise contour to be created from a smaller set of initial points reducing the manual initialization time required [36]. Another version of snakes, GPS snakes, can be used to correct GPS trail data and map it to an aerial image of

the trail allowing for proper trail identification in an image [37].

Chapter 3

Software Design

Continuous monitoring of an area for changes, especially due to natural disasters, poses many challenges. This section attempts to develop a software program that is easy to use and can run autonomously after initial setup that can detect and notify the user when significant changes are detected. The four main components of this software are specifying an area of interest, creating a request for satellite images, analyzing the satellite images, and finally automation for continuous monitoring. Each of these main components of the software are described in detail in this section.

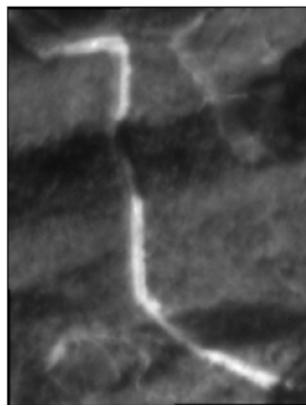
3.1 Specifying the Area of Interest

The first step to providing useful satellite monitoring information is to ensure that the captured images cover the desired AOI. How large an AOI is and how much extra area is included in it that is not relevant for monitoring can greatly impact computation time as well as data limits. Including irrelevant area in an AOI also increases the potential of detecting insignificant changes. For these purposes it is an extremely important first step to specify an AOI that covers just the actual area that needs to be monitored.

3.1.1 Identifying a Corridor Using a Snaking Method

The first attempt at specifying an AOI was completed semi-autonomously. Given a satellite image that included a path, in this test case from a buried pipeline, software was designed to analyze the image and detect the path using a snaking method.

To detect the path the satellite image was first converted from a red-green-blue (RGB) color scale to a gray color scale image shown in Figure 3.1a. Next, a contrast threshold was applied to the gray-scale image. Each pixel in the image was compared to the threshold value, if it exceeded the threshold the pixel was given a value of 1, if it was less than the threshold it was given a value of 0. This contrast threshold resulted in a black and white image, shown in Figure 3.1b, in which the white areas were above the threshold and represented the brightest areas in the image which correlates to the empty dirt trails which reflect more sunlight.



(a) Gray Scale Image



(b) Contrast Threshold Image

Figure 3.1: Gray-scale satellite image of pipeline trail compared to the black and white image that resulted from applying a contrast threshold.

The contrast threshold image was able to identify many sections of the path that were not obscured by shadow, but, it also identified other small bright spots on the image that were

not part of the desired path. To remove the undesired regions, the area of each white spot in the contrast threshold image was calculated. If the area of the region was less than 20% of the area of the greatest region, then that region would be converted to black. This resulted in an image, shown in Figure 3.2, that only contained regions of the desired path.



Figure 3.2: Contrast threshold image with region areas less than 20% of the greatest area removed.

Next, the regions of the path need to be connected to represent the entire trail. The region closest to the top left corner of the image was selected as the starting region of the trail. The region closest to this initial start region was also selected with all remaining regions hidden as shown in Figure 3.3a. An active contour snake was then produced on the image, shown in blue in Figure 3.3b, with an initial condition of a circle surrounding the image, shown as the dashed red line in Figure 3.3b. The region surrounded by the snake was then changed to white to fill the gaps in the trail as shown in Figure 3.3c. This process was repeated by using the new combined region and connecting it to the next closest region, shown in Figure 3.3d - 3.3f, until all regions were connected.

Once all regions were connected this mask was applied to the RGB satellite image, the full RGB image is shown in Figure 3.4a and the path is highlighted in blue in Figure 3.4b. This

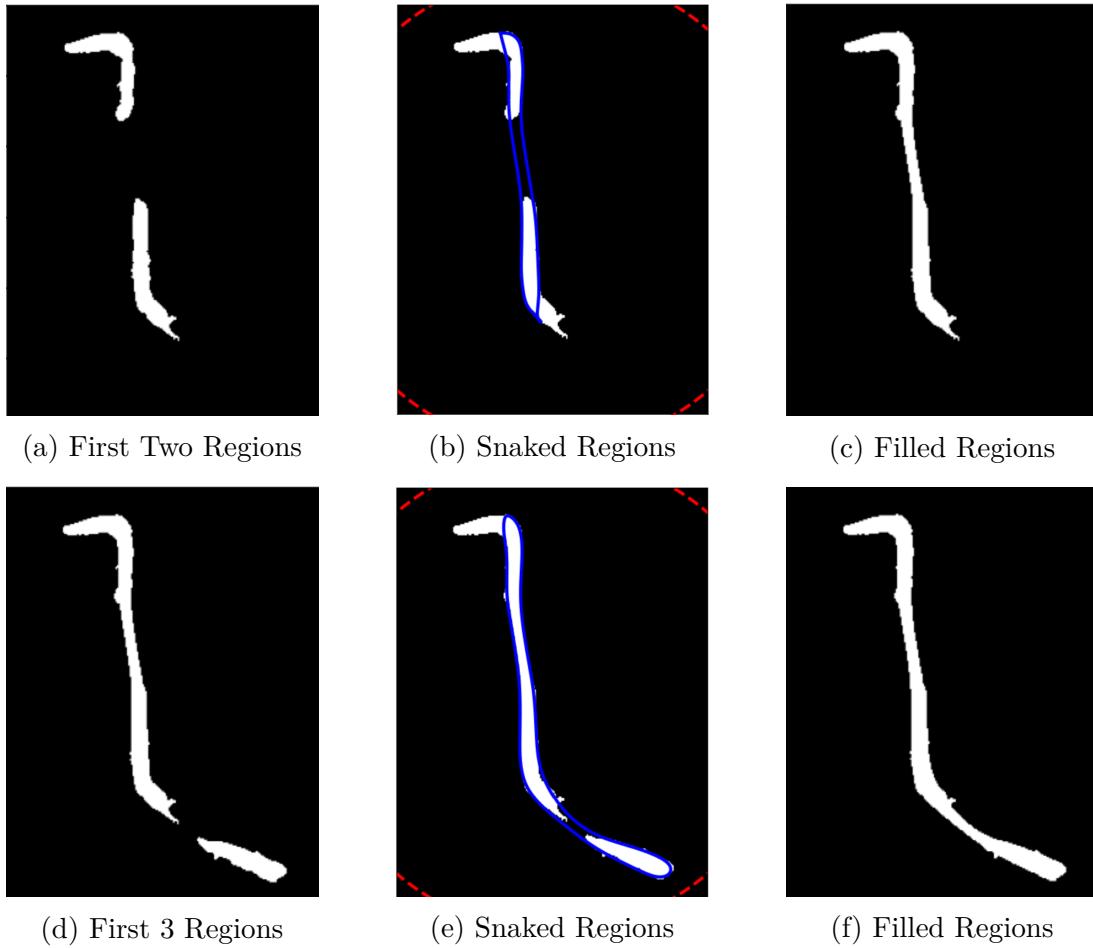
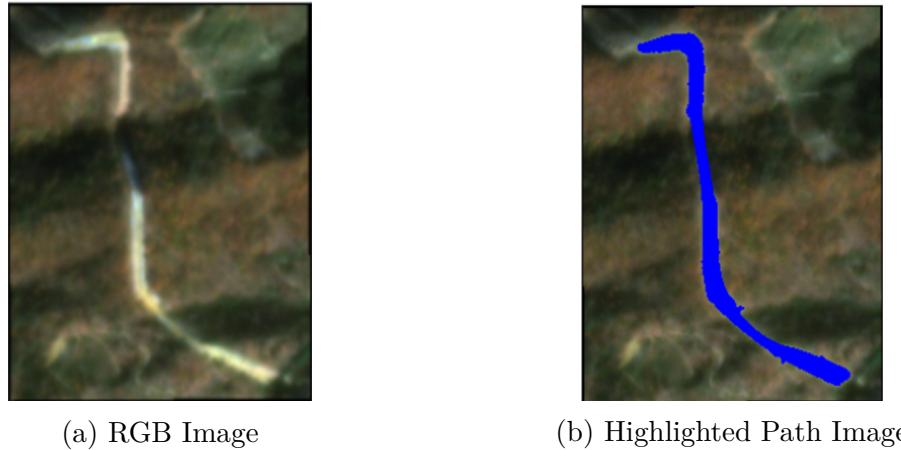


Figure 3.3: Using snakes to fill in gaps in a trail. a-c shows the snaking of the first two regions in the image, d-f shows the snaking of the combined first two regions with the addition of a third region.

Showed the ability to use snakes to detect a non linear dirt path through the forest and through shadows.

While this path detection system shows great promise it was ultimately not used in the specification of the AOI in the final software package. This was due to the fact that the path had to be detected from a satellite image which needed to have a specified AOI bounded by coordinates to begin with to download the image. This method could still be used in future



(a) RGB Image

(b) Highlighted Path Image

Figure 3.4: RGB satellite image compared to the same RGB satellite image with the detected path highlighted in blue.

applications when trying to identify dirt paths from given satellite or aerial imagery data.

3.1.2 Specifying a Corridor Using Coordinates

Another way to specify an AOI is by using coordinates. A geojson file can be created which can be used to specify the geometry of any polygon as a list of GPS coordinates. A great tool for generating geojson files is geojson.io [1]. This website allows the user to draw any polygon of up to 200 points on a map and will generate a properly formatted geojson file that specifies the drawn area of interest.

This is a very easy to use visual system for specifying an AOI and allows for very exact polygons to be created, an example AOI drawn with geojson.io over Lane Stadium is shown in Figure 3.5. This method would also help ensure that undesired change detections do not happen outside the desired monitoring zone. The process may be slightly tedious depending on how exact of a boundary is drawn and over what area but in the end specifying the AOI

using coordinates was determined as the best course of action in this software system.

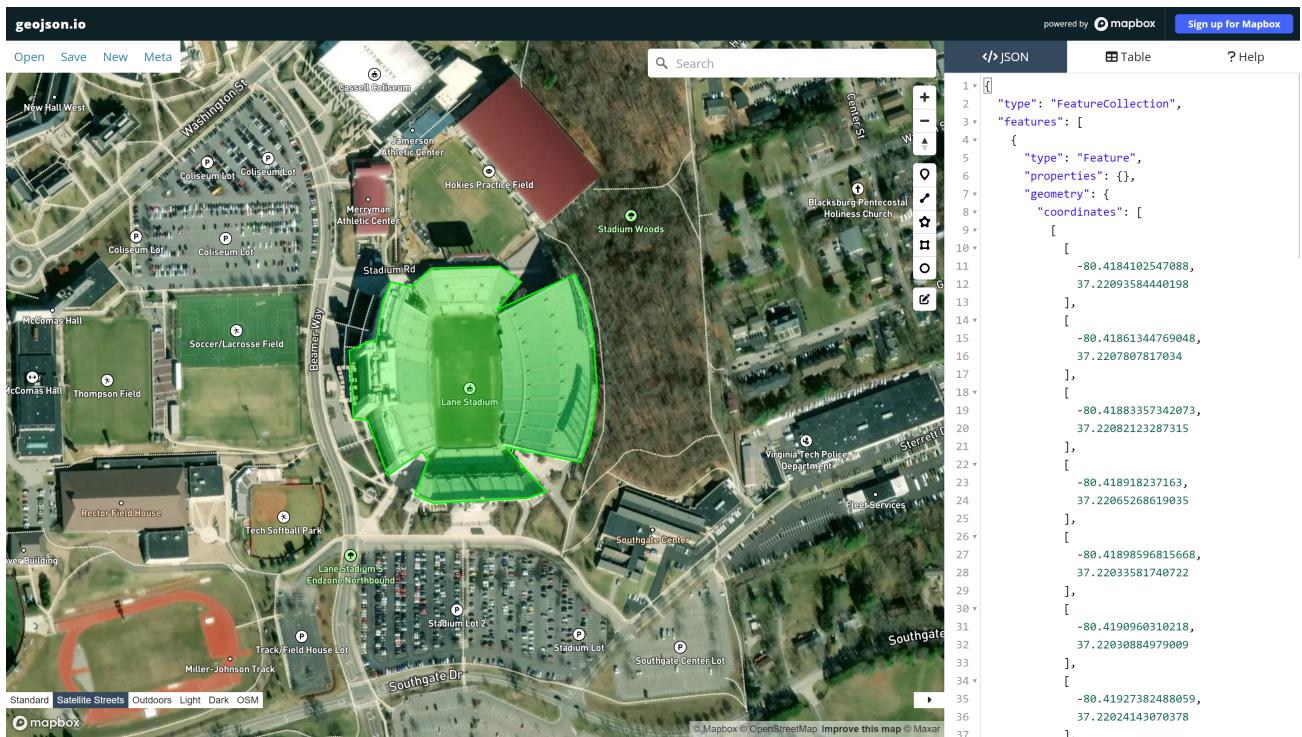


Figure 3.5: Screenshot of geojson.io [1] being used to create an AOI polygon covering Lane Stadium.

3.2 Creating a Request for Satellite Images

Planet Labs was chosen as the source of all satellite imagery in this project. It was chosen due to its great features including 3m resolution, near daily imagery, and a great subscriptions tool for setting up access to imagery on a regular basis. A python script was created to submit a subscriptions request that would signal Planet Labs to send a copy of all data that met the subscription requirements each time new data became available [38]. This script only needs to be run once when specifying an area to monitor then will continue to deliver images as specified. Two main components go in to making a subscription request, creating

image filters and delivering the data, these two components will be explained in depth in this section.

3.2.1 Creating Image Filters

Planet Labs offers a wide range of imagery data types [9]. Creating image filters allows the user to specify exactly what types of imagery data should be received as well as when and where the images should be taken from. The first and most important filter is the geometry filter. The geometry filter takes an input of the geojson file that was created to specify an AOI in Section 3.1.2 and only looks for imagery that overlaps this AOI. The subscription request is also set to clip images to the AOI which minimizes data storage costs.

Two more main filters are used in the request, a date range filter and a cloud cover filter. The date range filter is used to specify the length of monitoring by entering a start date and an end date, the start date can be set in the past if it is desired to have more historical data. The cloud cover filter is currently set to only retrieve images that have less than 50% cloud cover but this value can be changed based on the user's specifications.

In addition to the content filters listed above (geometry, date range, and cloud cover) types of imagery data can also be specified. In this program the subscriptions were set to retrieve Ortho Analytic 4B SR, Ortho UDM2, and Ortho Analytic 4B XML data types. The Ortho Analytic 4B SR data type includes the main image data covering the RGB and near infrared (NIR) bands, this data has been orthorectified for geometric distortions and corrected due to surface reflectance and atmospheric changes. The Ortho UDM2 data type contains mask

data for the image, with information on the classification of each pixel type (clear, cloud, shadow, snow, etc.) and the confidence that a particular pixel was classified correctly. Finally, the Ortho Analytic 4B XML contains background data that allows for the images to be harmonized into a consistent color range.

The combination of these filters allows for the most relevant imagery data to be retrieved without wasting valuable storage space on less effective data. For an external user to submit a subscription request the only update to the existing code required for the filters section would be to include a file path to the geojson file created in Section [3.1.2](#).

3.2.2 Data Delivery

Imagery from Planet Labs subscriptions can only be uploaded to a GCS space. A GCS project and sub bucket was created to store the satellite images. The json credentials file associated with the GCS bucket was downloaded and stored in the same folder as the python script to create the subscription request. The delivery location for the subscription was then set as the GCS bucket with the credentials linked to the decoded base64 version of the json credentials file. This setup allows for Planet Labs to automatically upload all data meeting the filter requirements to the specified GCS location with no additional user inputs.

3.2.3 Submitting the Request

Once the filters have been created and a delivery location has been specified the subscription request can be made. The request is made by first combining the filters and delivery location

information into a json style file. This file is then posted to the Planet Labs Subscriptions Application Programming Interface (API) URL and is authenticated with the API key that can be retrieved from the user's Planet Labs account. Once the request has been submitted it will continue to upload data to the GCS location until the specified end date passes, with no additional input needed from the user. If a change needs to be made to the subscription an update can be made by sending an updated json file to the specific subscription URL.

This process should be relatively easy for users as the the only inputs required are the AOI, date range, GCS location and credentials, and the Planet Labs API key. These inputs are easily created or accessible and only need to be entered once unless changes to the AOI are needed.

3.3 Analyzing the Satellite Images

Once satellite imagery data has been requested, as described in Section 3.2, this data must be analyzed and processed to generated usable results. This section details the process of downloading relevant data, pre-processing it, detecting changes, and finally sending a report to the user.

3.3.1 Downloading Most Recent Images

Before imagery data can be analyzed it must first be downloaded locally so that the information is accessible. Previously in Section 3.2.2 imagery data was uploaded to a GCS

bucket. The same credentials used to upload to the GCS bucket were also used to request downloads from the bucket. Before any downloads were made a local folder to store the data was created if one did not already exists for the particular subscription, once this was checked downloading could begin.

Next, each file from the GCS bucket was cycled through to be downloaded. Since this script was designed to be run daily or even multiple times a day it would waste space and computation time to repeatedly download the same image data to a local folder rewriting it each time the script was run. To prevent this the file path of each image was checked to see if it already existed before it was downloaded, if it did already exist it was skipped. Additionally, sometimes multiple images are captured just minutes apart on the same day which takes away some comparison value. To prevent this from occurring the date of the image and file type was extracted from each image name before it was downloaded. If there was already a file of the same type from the same day downloaded the image would be skipped. This process ensured that daily imagery was captured and prevented repeat downloads.

While Planet Labs provides imagery nearly everyday sometimes a day is missed, either due to a satellite not passing over the area or due to excessive cloud cover resulting in an unusable image. When this occurs there is no new imagery data to make a comparison to and try to detect a change. Therefore, the script is set up to check if a new image has been downloaded, if a new image has not been downloaded then the script will end without proceeding any further to prevent repeat comparisons.

If a new image has been downloaded the program checks to see if there are at least six days worth of satellite imagery data, if there is not the script will end. If there is at least six

days worth of data then the current image is loaded as well as the five most recent previous images which will be used for comparison.

The RGB data for each day is loaded and a raw and a visualized image is created for each day. The raw images are generally quite dark, as shown in Figure 3.6a. To adjust the images to a more visual range each RGB band was normalized resulting in an easier to visualize image as shown in Figure 3.6b.

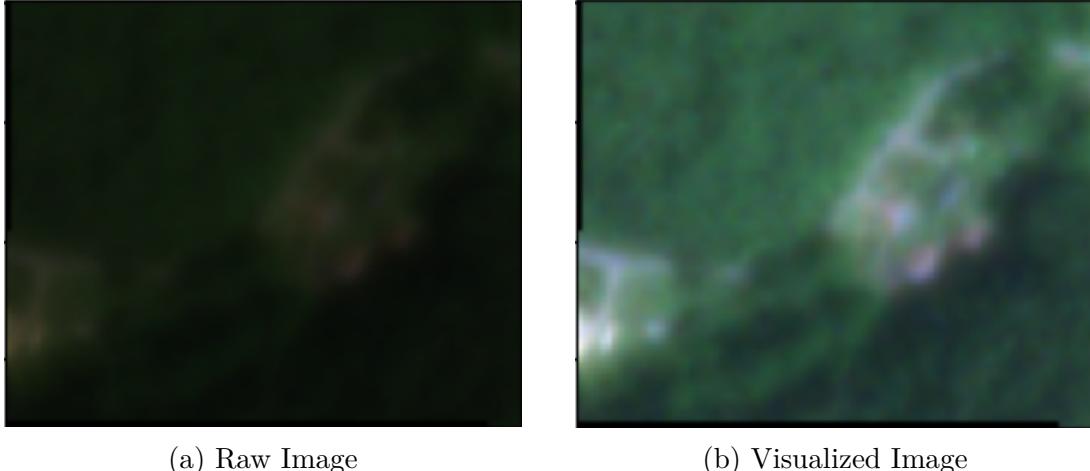


Figure 3.6: Raw RGB satellite image data vs a visualized version of the data in which the RGB bands have been normalized.

3.3.2 Addressing the Cloud Problem

When trying to discern significant changes between two images it is important to remove unwanted sources of difference. In the case of satellite image monitoring some things that may cause irrelevant differences between images are clouds, shadows from clouds, and fog. These can cause a significant change between images to be falsely detected. These difference sources also obscure the desired ground data making those pixels useless. Since the pixels

identified as clouds, shadows, fog, etc. do not provide any useful information and result in false change detection masking was used to remove these pixels to prevent detection due to irrelevant changes.

The UDM2 files were used to create a mask for each of the six current images. The first band in the UDM2 file is a boolean operator for each pixel location on the satellite image; the pixel is given a true value if it is clear and a value of false if it is obscured in any way whether it is by cloud, shadow, fog, or reflection. The seventh band in the UDM2 file is a confidence value from 0-100 for each pixel location. This confidence value represents the confidence that the pixel type has been correctly classified.

For each of the six current images, a mask of the same size was created and initialized to zeros (all black). Then, each pixel location was checked to see if it was labeled as clear and if the confidence value was greater than or equal to 95. If both of these were true there is a good chance that the pixel truly is clear and the corresponding pixel in the mask was given a value of 1 (white). Once all pixels were checked a mask was generated in which white areas represented clear regions and black areas represented obscured regions, an example of this mask is shown in Figure 3.7b. The clear percent of each image was also calculated by dividing the number of clear pixels in each image by the total number of pixels in each image.

This mask could then be multiplied by the RGB image to get a final image that only contains clear values, an example is shown in Figure 3.7c. This process ensures that only clear values will be used in calculating the baseline variance, described in Section 3.3.3, and in comparing the current image, described in Section 3.3.4.

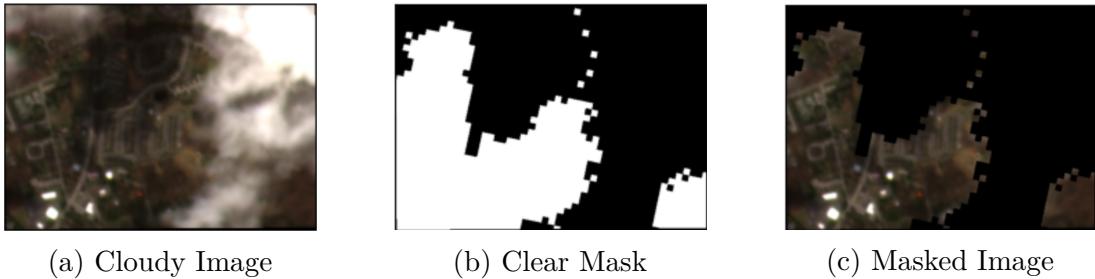


Figure 3.7: Satellite image of a residential area in Blacksburg, VA obscured by clouds and shadows (a) followed by a mask of the clear pixels in the image (b) and the image when multiplied by the clear mask (c).

3.3.3 Creating the Baseline Variance

No two images of the same outdoor area will look exactly the same, even on consecutive days absent of major Earth movement. These slight variations in color can be due to lighting changes, vegetation health, or even blooming flowers. While these changes are often small if a pixel by pixel difference comparison is conducted between two images these differences can show up. To prevent small changes from being detected a threshold is often established.

Difference thresholds often require precise and intensive tuning for each application case; to monitor a wide range of areas with satellite imagery this would pose a serious issue and take an immense amount of time. As an alternative to a fixed difference threshold a moving threshold for each individual pixel will be calculated.

The moving threshold for each individual pixel was set by looking at the pixel values across the five previous images. First the maximum pixel value in each of the three bands (red, green, and blue) was determined from the five previous images at each pixel location, explained in pseudo-code in Figure 3.8. Next, the minimum value was determined in each band at every pixel location.

```

FOR each row of pixels in the image
    FOR each column of pixels in the image
        R1 = red pixel value at this row and column location in image 1
        R2 = red pixel value at this row and column location in image 2
        R3 = red pixel value at this row and column location in image 3
        R4 = red pixel value at this row and column location in image 4
        R5 = red pixel value at this row and column location in image 5

        G1 = green pixel value at this row and column location in image 1
        G2 = green pixel value at this row and column location in image 2
        G3 = green pixel value at this row and column location in image 3
        G4 = green pixel value at this row and column location in image 4
        G5 = green pixel value at this row and column location in image 5

        B1 = blue pixel value at this row and column location in image 1
        B2 = blue pixel value at this row and column location in image 2
        B3 = blue pixel value at this row and column location in image 3
        B4 = blue pixel value at this row and column location in image 4
        B5 = blue pixel value at this row and column location in image 5

        Maximum red value = maximum value out of (R1, R2, R3, R4, R5)
        Maximum green value = maximum value out of (G1, G2, G3, G4, G5)
        Maximum blue value = maximum value out of (B1, B2, B3, B4, B5)
    
```

Figure 3.8: Pseudo-code to describe how to determine the maximum pixel value.

Determining the minimum value was more difficult than finding the maximum value due to the fact that the masked images were being examined. This meant that if a pixel was obscured its value was set to zero in all three color bands. Finding a minimum value of zero due to an obscured value was not desired. To combat this issue the maximum value was first checked, if the maximum value was equal to zero across all images at a specific pixel value then the minimum was also set to zero, as described in the pseudo-code in Figure 3.9. If the maximum did not equal zero the minimum value was determined by selecting the minimum value that did not equal zero.

Next, the maximum difference for each pixel value was computed for each pixel location. This was calculated by simply subtracting the minimum value from the maximum value at

```

FOR each row of pixels in the image
    FOR each column of pixels in the image
        IF maximum red value = 0
            minimum red value = 0
        ELSE
            minimum red value = minimum value that does not equal zero out of (R1, R2, R3, R4, R5)

        IF maximum green value = 0
            minimum green value = 0
        ELSE
            minimum green value = minimum value that does not equal zero out of (G1, G2, G3, G4, G5)

        IF maximum red value = 0
            minimum red value = 0
        ELSE
            minimum red value = minimum value that does not equal zero out of (G1, G2, G3, G4, G5)

Maximum difference red = maximum red value - minimum red value
Maximum difference green = maximum green value - minimum green value
Maximum difference blue = maximum blue value - minimum blue value

```

Figure 3.9: Pseudo-code to describe how to determine the minimum pixel value.

each location. The maximum, minimum, and maximum difference was calculated for both the raw images and the normalized visual images. Much greater maximum differences were calculated using the visual images, likely due to the artificial normalization process that brought the image onto a better viewing scale. This difference is shown in Figure 3.10 which shows an example of the maximum differences of the raw image, which is almost completely black, versus the maximum differences of the visual images, which shows much more color even though it is still minimal. The raw images were therefore chosen to make the comparison as they did not include errors due to artificial adjustment.

The maximum difference image provides a good variable threshold that can adapt through changing seasons and provide more cushion in areas of an image that tend to vary more than others. Five was chosen as the number of previous images to compute this maximum difference image as it allowed time for slight variations to be registered but was short enough

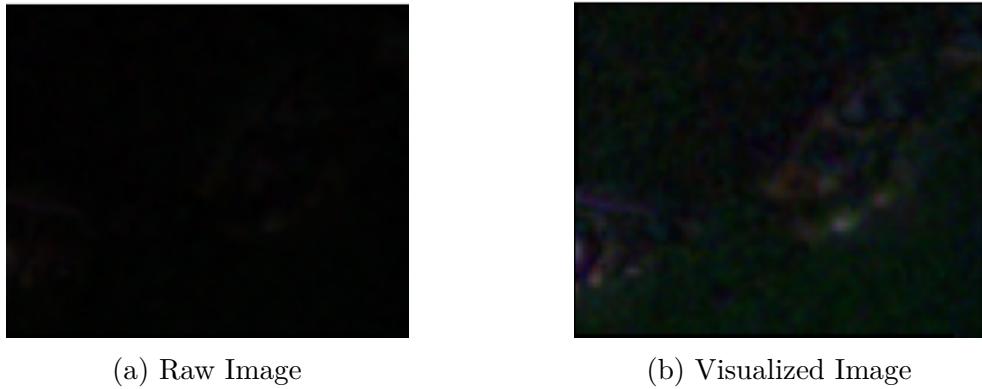


Figure 3.10: Maximum difference of a raw RGB satellite image vs the maximum difference of a visualized version of the data in which the RGB bands have been normalized.

to not span major changes such as a change of seasons. While using the maximum difference image as the threshold worked well there were still cases in which only one image was clear on a specific pixel resulting in a maximum difference of zero. This would mean that any slight variation in that pixel could be registered as a change. To combat this a final threshold matrix was developed that compared the maximum difference to a pre-set minimum allowable difference. The final threshold matrix was then filled in pixel by pixel with the maximum threshold difference, but, in locations where the calculated maximum threshold difference was less than then pre-set minimum allowable difference, the minimum allowable difference was entered as the final threshold, this is explained in pseudo-code in Figure 3.11. The minimum allowable difference was set to 5% of the color scale allowing for slight variations.

This combination variable threshold is robust and can easily adapt to small seasonal changes in the environment to help prevent insignificant difference detections as described in the following Section 3.3.4.

```

FOR each row of pixels in the image
    FOR each column of pixels in the image
        Minimum allowable difference = 5% of pixel color scale range

        IF minimum allowable difference > maximum difference red
            Final threshold for red at this row and column location = minimum allowable threshold
        ELSE
            Final threshold for red at this row and column location = maximum difference red

        IF minimum allowable difference > maximum difference green
            Final threshold for green at this row and column location = minimum allowable threshold
        ELSE
            Final threshold for green at this row and column location = maximum difference green

        IF minimum allowable difference > maximum difference blue
            Final threshold for blue at this row and column location = minimum allowable threshold
        ELSE
            Final threshold for blue at this row and column location = maximum difference blue

```

Figure 3.11: Pseudo-code to describe how to determine the variable threshold.

3.3.4 Comparing the Current Image

To determine if any significant changes have occurred in an image it must be compared to some previous baseline image. Using the most recent previous image is the obvious choice to make a comparison as the least time has past so there is less of a chance that an event causing a significant change has occurred. However, this is not always the best choice as the previous image may have obscured areas that overlap with the current image's clear areas and vice versa. This could lead to a very small viable comparison window even with relatively small obscured regions in each individual image. Therefore, the choice of comparison image needs to be addressed rather than simply using the previous image.

To select a comparison image the current image is checked against each of the five previous images to determine the clear overlap percent. To calculate the clear overlap percent the clear mask for the current image, whose creation is described in Section 3.3.2, and the clear mask for the selected previous image are multiplied together. When multiplied together,

another matrix is created filled with zeros in all pixel locations that were obscured in either image. The clear overlap percent is then calculated by first counting each pixel that does not equal zero and dividing that by the total number of pixels.

The clear overlap percent is then compared to the set minimum overlap percentage, which was set to 75%, if the clear overlap percent is greater than the minimum then that image is selected for comparison. The most recent previous image is checked first for an adequate clear overlap percent to ensure that the most recent usable data is used, if an image fails to have a high enough overlap then the next previous image is checked. If none of the five previous images have enough clear overlap with the current image then the comparison is not run.

Once a previous image with adequate clear overlap with the current image is selected, the two images can be compared. The absolute value of the difference of the current clear masked raw image and the selected comparison clear masked raw image is calculated. This difference was then compared to the final threshold matrix, described in Section 3.3.3. Each pixel location was checked to see if the difference in any of the RGB bands exceeded the value in the final threshold matrix at the same location. If the difference did exceed the value then the difference mask image value was set to 1 (displayed as white), if the difference did not exceed the value the difference mask image value was set to 0 (displayed as black). This would create a black and white image with black indicating no change and white indicating a detected change. This image, however, would indicate no change detected in areas that could not be compared due to obscured pixels which is not necessarily true. To represent the data more accurately obscured pixels that could not be compared were given a value of 0.5 (displayed as gray). This tri-color image was able to better represent each region to the user, an example tri-color difference mask depicting Minden, IA following a tornado strike

is shown in Figure 3.12.

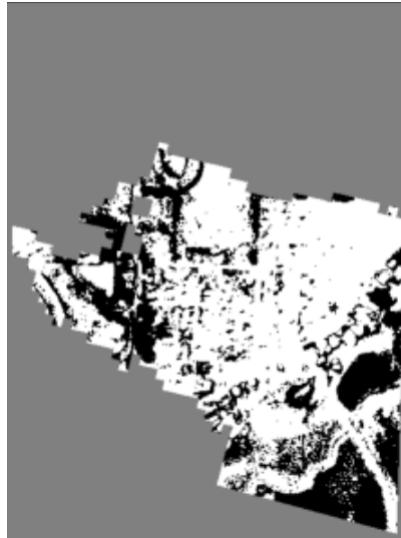


Figure 3.12: Difference mask of Minden, IA after a tornado. White areas indicate a significant change, black areas indicate no change, and gray areas indicate that a comparison could not be made.

Once the tri-color difference mask has been made the total affected area in the image is calculated. The number of white pixels in the image are counted, this number is divided by the total number of pixels in the image to get the total affected percent. The total affected area is then calculated in square meters by multiplying the number of white pixels by 9, since the resolution of the images are 3m. This affected area is then converted to acres by dividing the number of square meters by 4047. Finally, the number of pixels in the largest white region is found and the largest affected area percent is calculated along with area in square meters and acres in the same way as for the total area.

3.3.5 Sending the Results

A tri-color difference image was generated and the affected percent and affected area for the total and largest regions of difference were calculated as described in Section 3.3.4. This is great and important data but it needs to be delivered to the user in a manner that is easy to read and access. A composite of all the relevant images and data, therefore, need to be combined into a single figure.

To compose this figure many pieces were combined, first, it was titled with the date of the current image and "Difference Analysis". Four images were included, the visual version of the previous image that was used for comparison (titled with the date captured), the visual version of the current image, the tri-color difference threshold, and an RGB image showing only the affected areas. A table is also needed to summarize the statistics on the affected percent and affected area. This combination of the table and images shows all of the important information relevant to the comparison, an example of this combined figure showing tornado damage to Minden, IA is shown in Figure 3.13. Finally, this figure is saved to the same local directory as the satellite images named for the date the difference analysis occurred so that the file can be accessed later regardless of whether or not a significant change was detected.

Once the summary figure has been created a decision needs to be made on whether or not a significant change was detected. Various metrics can be set up to determine if a significant change has occurred, in this case values were chosen for maximum change percent and a maximum change area in square meters. The maximum change was set to 5% of the total area or 100 square meters, if either of these values were exceeded a significant change was said to occur, pseudo-code explaining this determination is shown in Figure 3.14. If no

04-29-2024 Difference Analysis

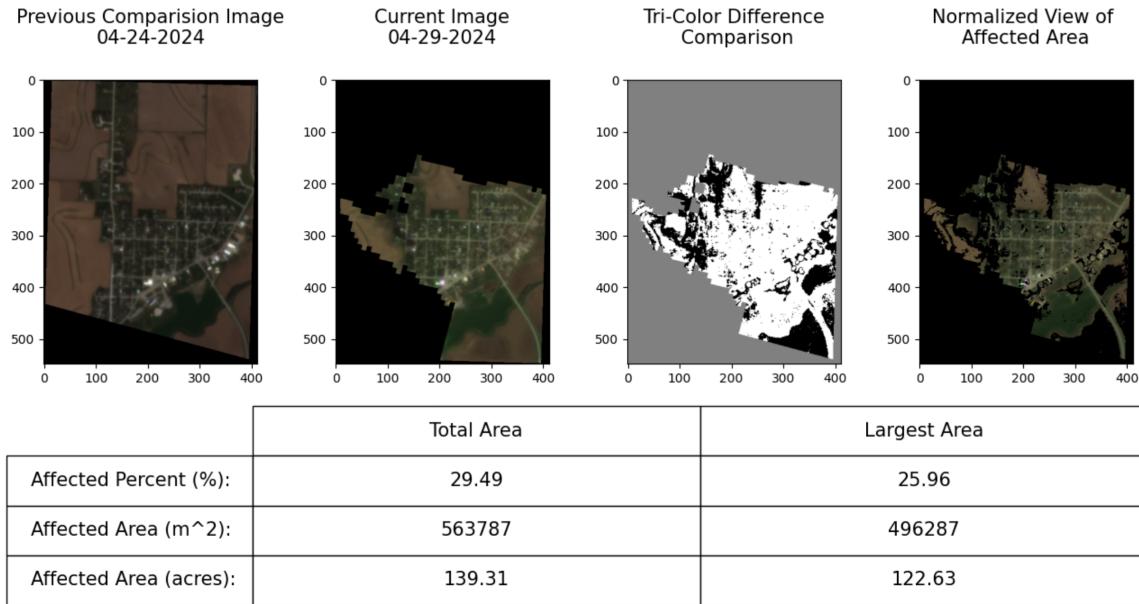


Figure 3.13: Summary figure of Minden, IA following a tornado strike depicting the comparison image, the current image, the tri-color difference, the RGB filled difference, and a summary table of the difference statistics.

significant change occurred then the program would end, if a change was detected the user needed to be notified.

```

Maximum change based on percent area = 0.05 * total image area (square meters)
Maximum change based on total area = 100 (square meters)

IF total affected area > maximum change based on percent area OR affected area > maximum change based
on total area
    Significant change detected = TRUE
  
```

Figure 3.14: Pseudo-code to describe how to determine if a significant change has occurred.

To notify the user of a change an email was automatically generated by logging into a secure email server using the sender's specified email address and secure access password. The receiver's email address was also specified, with the option of including multiple recipients. The email was automatically generated with a subject line indicating that a change was

detected on the specific date. The body of the email describes when the change was detected, the affected percentage, the affected area, and suggests that the user views the area to determine if action is necessary. The summary figure is also attached to the email to give the user a full picture of what occurred by simply checking their email. A sample email that was generated by the program is shown in Figure 3.15.

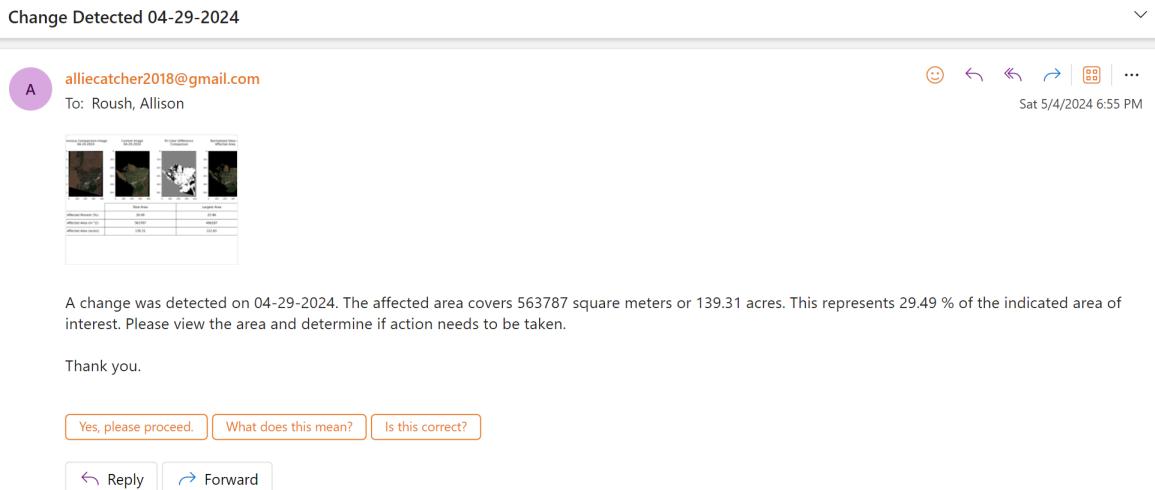


Figure 3.15: Screenshot of email generated notifying the user that a significant change was detected in Minden, IA following a tornado strike.

3.4 Automation for Continuous Monitoring

Ideally a program could be set up to continuously monitor satellite imagery for changes with no input from a user and would be able to notify the user only if significant changes were detected. The software developed for this project and described throughout Chapter 3 can be set to run autonomously with minimal user input at the start of the monitoring period. The first user input required is a geojson file specifying the AOI which can easily be generated using the steps described in Section 3.1.2. Next, the user will need to update the

subscription script to include their specific GCS bucket and credential information, Planet Labs API key, desired date range, and local path to the geojson file. Once this information has been updated the user must manually run the script once to generate the subscription.

The script to analyze the images, described in Section 3.3, needs only minor user updates to specify the GCS bucket and credentials, the email address and password for the account that will be sending the notification emails, and the email address of the account that wishes to receive the notifications. Changes can also be made to the thresholds for change notification such as maximum affected percent or maximum affected area, but these can also be left at the default values. Once these minor updates have been made to the script it is ready to be set up to run autonomously.

To run the script to analyze satellite images for changes autonomously Windows Task Scheduler was used. In Task Scheduler a new task was created and named "Satellite Imagery Monitoring". The task triggers were set to run the script everyday at 8am as well as every time a user logged into the computer, a screenshot of the task triggers is shown in Figure 3.16a. Next, the task action was specified. The action was set to run a program and the selected program was set to the local path to pythonw.exe. This python execution application allows a python script to be run in the background without showing anything on the screen. A sub-path to the analyzing satellite imagery python script was also specified in this tab as shown in Figure 3.16b.

Once these steps are taken and the task is saved the program will autonomously run each day and on every computer login. This is all done in the background with no additional user input required, but it will still send an email to notify the user of significant changes.

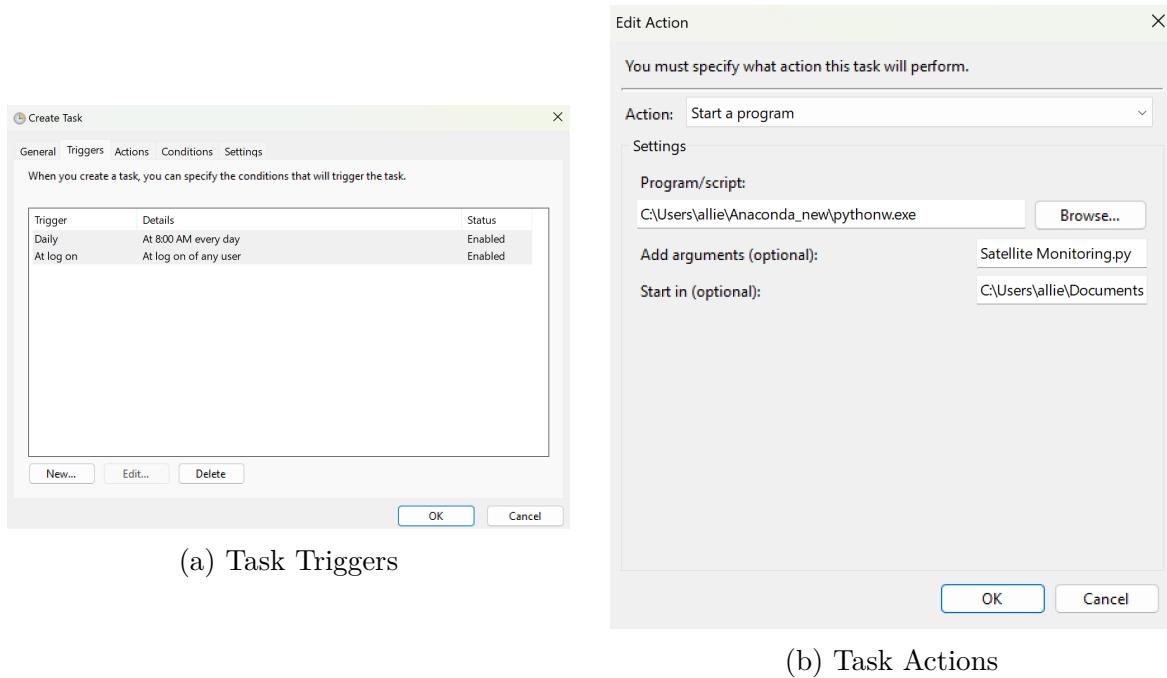


Figure 3.16: Screenshots of Windows Task Scheduler being used to specify triggers and actions to configure the analyzing satellite imagery python script to run autonomously.

Chapter 4

Results

Validation tests were conducted on each component of the system described in this paper. Simple tests were completed on the software processes used to specify an AOI (described in Section 3.1), to submit a request for satellite imagery (described in Section 3.2), and to automate the process for continuous monitoring (described in Section 3.4). The bulk of the validation testing was on the software used to analyze the satellite images (described in Section 3.3). Four test cases were examined to determine the success of identifying change areas in satellite imagery in locations where various natural disasters occurred.

4.1 Simple Tests for Processes

The first process that was tested was creating the AOI using coordinates. The AOI was specified by creating a polygon with selected points on a map using geojson.io which generated a geojson file with all the necessary points, the full process is explained in detail in Section 3.1.2. This process was completed multiple times to ensure that polygons with many points and complex geometries could be captured, a screenshot of geojson.io in use to select an AOI is shown in Figure 4.1. All AOIs and associated geojson files were generated properly, validating the AOI specification process.

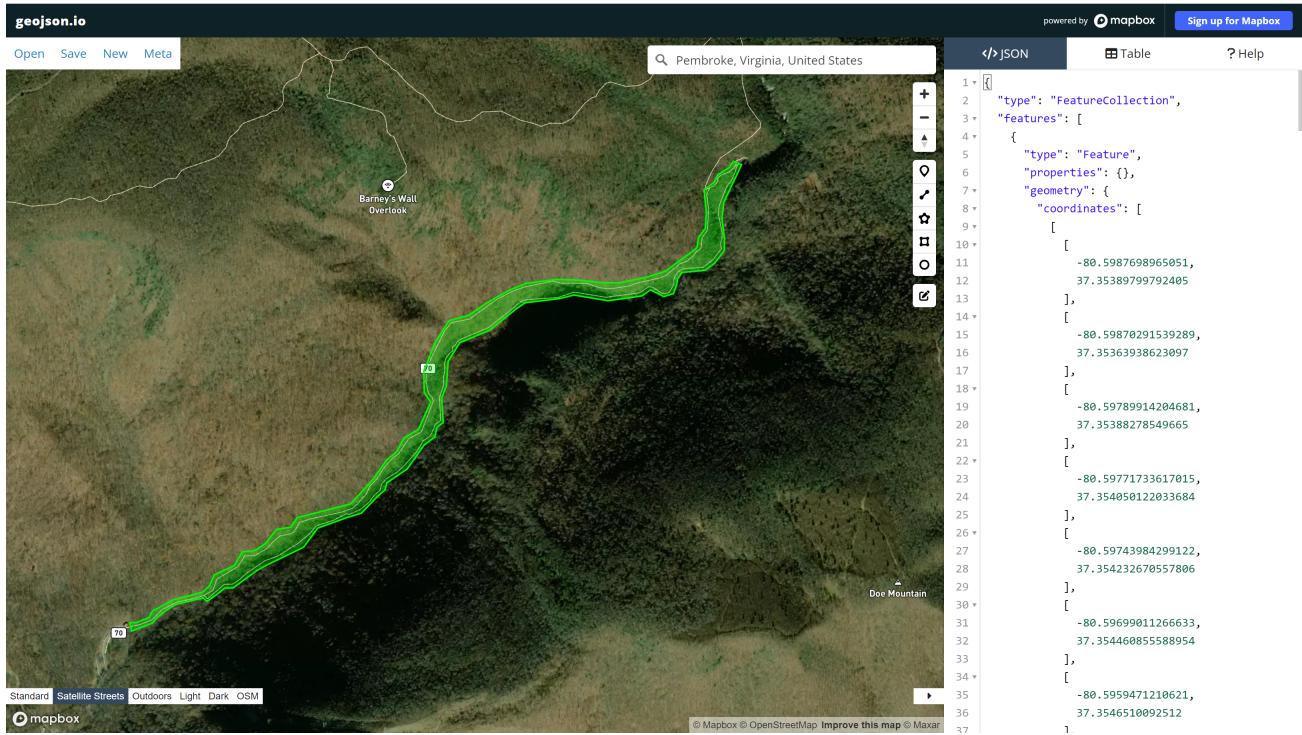


Figure 4.1: Screenshot of geojson.io [1] in use to select a complicated geometry following the Cascades hiking trail.

The next process that was tested was the Planet Labs subscription software, detailed in Section 3.2. Multiple subscription requests were tested with various AOIs selected as well as different date ranges. All requests were successfully submitted and all relevant images were automatically uploaded to the designated GCS location. The GCS storage location was manually checked daily against the Planet Labs website to verify that images were uploaded to GCS on the days when imagery meeting the requirements was captured. All requested imagery was successfully uploaded to GCS, validating the Planet Labs subscription process.

The process to set up automation for continuous monitoring, described in Section 3.4, was tested using a task scheduler test script. This python test script was set to create a test image, save it in a local folder, and finally send an email to the user with the saved image

included as an attachment. The functionality of this test script was set up to mimic the satellite imagery analyzing process without actually retrieving the satellite imagery data. This script was then scheduled to run automatically with Windows Task Scheduler on a daily basis and on user login. An email notification with an attachment was successfully generated in the background at login without user input, screenshot shown in Figure 4.2, validating that the process could be run autonomously. However, it is important to note that the scheduled daily run time could only be completed if the computer was turned on at the time of the scheduled event, if it was off the event was performed the next time the computer was turned on.

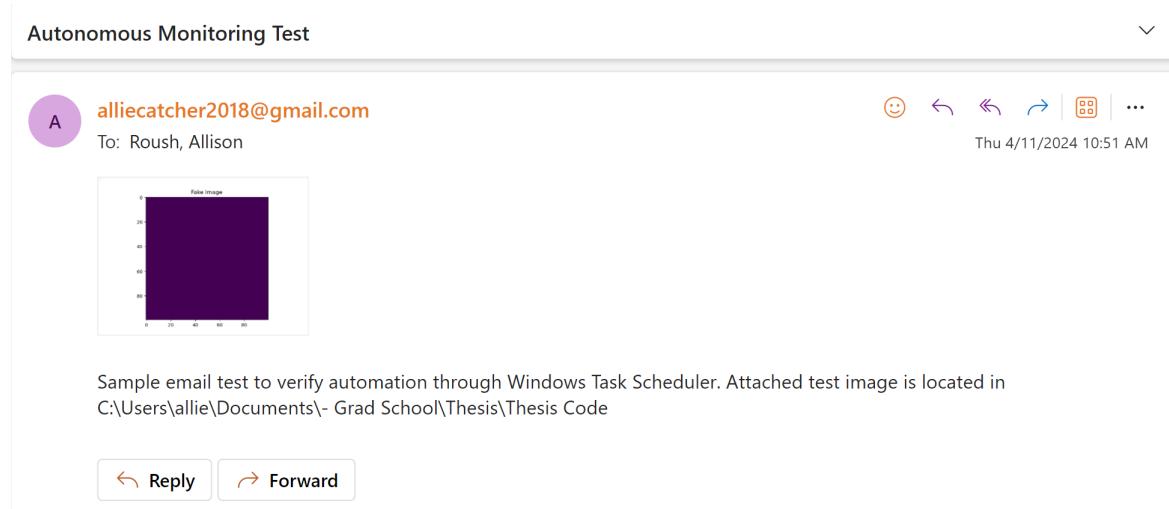


Figure 4.2: Screenshot of an email that was automatically generated at computer login by the Windows Task Scheduler running the test script.

4.2 Change Detection Results

To determine if the software developed to analyze the satellite images, described in Section 3.3, was successful at identifying significant regions of change between images the program

was run using images from four test cases. The first two test case locations were chosen as areas where prior knowledge of a significant change occurred due to a natural disaster. Five days of imagery data before the event was collected as a baseline and compared to the first imagery data from after the event. The detected changes were then compared to documented areas of change from the event. The third test case was chosen as a baseline comparison. A location was chosen that showed no signs of change to validate that the program would not identify changes where none had occurred. The final test case location was chosen as a region with known change due to natural disaster that was successfully identified using HazMapper and compared to ground truth surveying in a previous study [19]. Selection of this region allowed for a comparison of change detection success between the software described in Section 3.3 and the closest current work, HazMapper.

4.2.1 Hurley VA Flood Example

The first location that was evaluated was the area surrounding a section of Guesses Fork Road in Hurley, Virginia. Heavy rainfall on August 30th, 2021 caused significant flooding and landslides in the area. The software to analyze satellite images (described in Section 3.3) was used to try and identify areas with significant changes due to the weather event.

Satellite imagery from the most recent five days prior to the flooding event were collected to create the baseline variance, as described in Section 3.3.3. Next, the first clear image post flood was retrieved and compared to the previous image and thresholded against the baseline variance, as described in Section 3.3.4. The results of this comparison are shown in Figure 4.3. This shows the previous image (from before the flood), the current image

(after the flood), the difference between the images, as well as the current image masked to show only the areas of change. The comparison also shows the numerical statistics on the percentage and area where significant changes occurred.

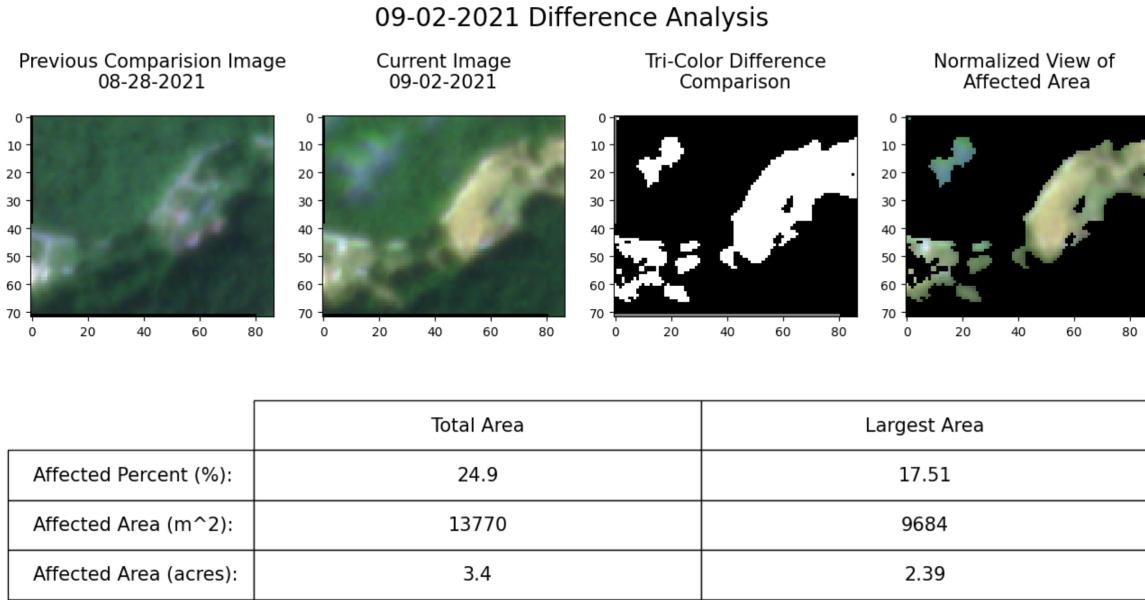


Figure 4.3: Summary figure generated to explain the comparison conducted on the Hurley, VA images.

Once the difference comparison was completed the resulting difference mask was compared to ground truth data of the affected flood areas. A UAV mission to image the Hurley Flood areas was conducted in November 2021 by Daniel Whitehurst, an ortho photo generated from the UAV images is shown in Figure 4.4a [2]. This ortho photo was then aligned, and overlaid with the post flood satellite image, shown in Figure 4.4b.

Next, to make a better comparison between the images the difference mask was placed on top of the ortho photo and the opacity was decreased so that the ground truth ortho photo could be seen through the difference mask, this is shown in Figure 4.5.

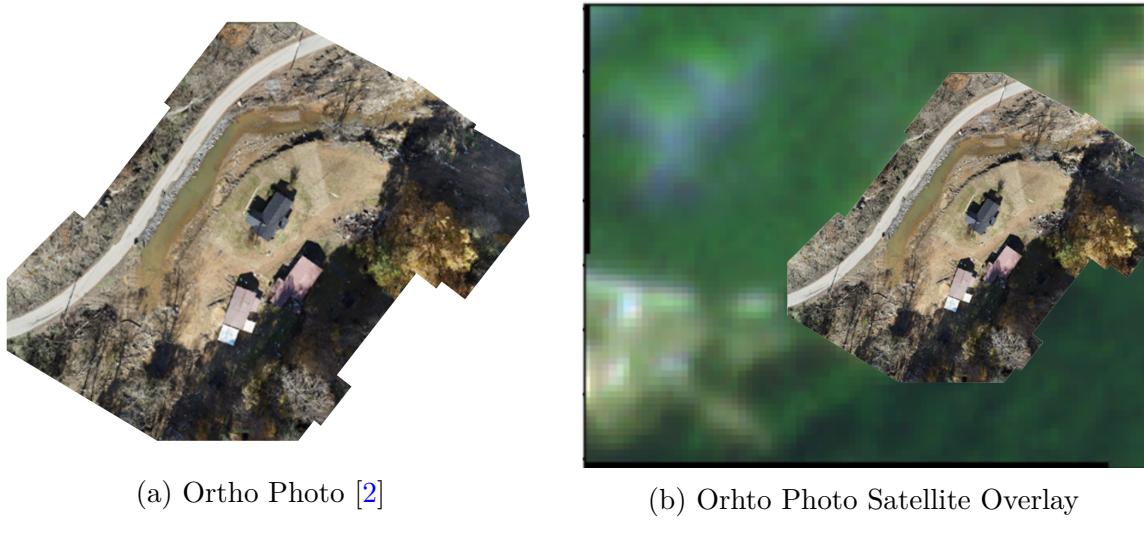


Figure 4.4: Ortho photo generated from UAV images taken in Hurley Virginia (photo copied from Whitehurst et al. [2]) and overlaid with a satellite image.

Some key observations can be discerned from Figure 4.5. First, the red outlines indicate sturdy structures such as a house or a garage. These structures were not impacted by the flood and, correctly, no changes were detected in these areas as they are black in the difference mask. Additionally, the changes are shown to extend over the road and throughout the rest of the imaged area which is consistent with reports from the flood.

This comparison between the UAV aerial images and the changes detected through the satellite image analysis software shows great promise as changes were detected in the correct areas and no change was detected over sturdy structures in this example. Additional areas of detected change follow along the stream bed and were also likely areas of flood damage. However, one region of change was detected near the upper left hand corner of the satellite image was not along the stream corridor. Upon visual inspection it appears to be a cloud or reflection of some kind. If it was a cloud or reflection then that means that those pixels

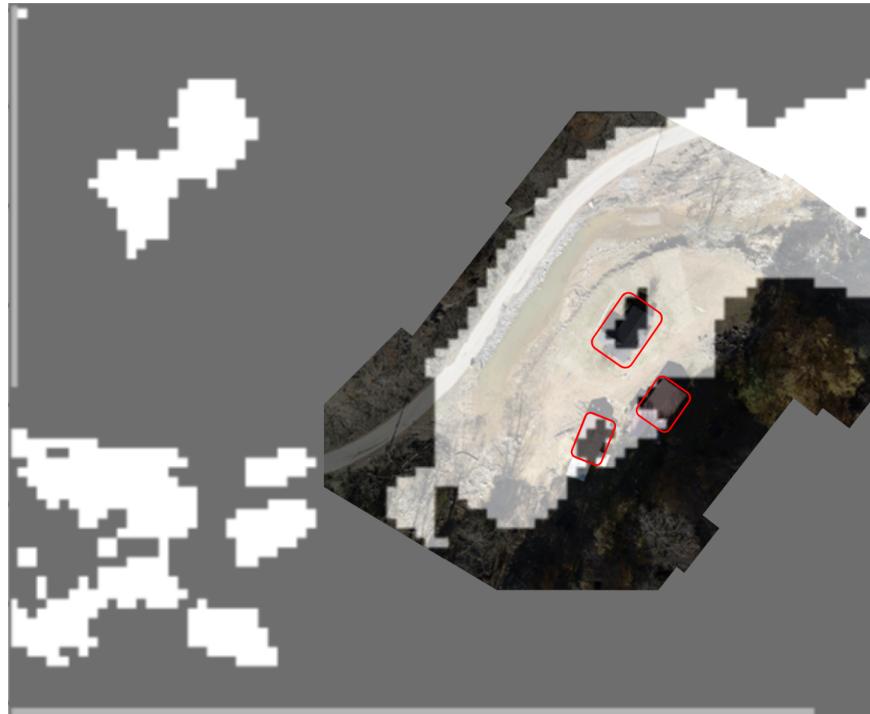


Figure 4.5: Difference mask overlaid with UAV ortho photo, areas outlined in red mark the location of sturdy structures.

were marked clear incorrectly. It is impossible to confirm this suspicion though as no UAV imagery data is available over that section of land. Despite this, the program worked well overall to detect significant changes.

4.2.2 Wheeling WV Debris Flow Example

The second change detection validation test was conducted over an area in Wheeling West Virginia. This location was chosen due to a debris flow recently occurring in this region. Several days of heavy rain eventually led to multiple landslides and debris flows in Wheeling, with a major debris flow occurring on April 3rd, 2024 disrupting the Mt. Zion Cemetery. The software to analyze satellite images (described in Section 3.3) was used to try and iden-

tify areas impacted by the debris flow.

Five days of satellite imagery data prior to the debris flow was collected to create the baseline variance, as described in Section 3.3.3. Due to rain and heavy cloud cover the prior five days of images was collected from March 22nd to March 29th, with the post debris flow image collected on April 7th.

After all satellite imagery data was retrieved the post debris flow image was compared to the previous image and thresholded against the baseline variance, as described in Section 3.3.4. The results of this comparison are shown in Figure 4.6. This shows the previous image (from before the debris flow), the current image (after the debris flow), the difference between the images, as well as the current image masked to show only the areas of change. The comparison also shows the numerical statistics on the percentage and area where significant changes occurred.

To verify the validity of the change detection, ground images of the debris flow were collected from reporters local to the area. One good comparison image taken at the site of the debris flow is shown in Figure 4.7 [3].

The ground truth image of the Wheeling debris flow, shown in Figure 4.7, shows that the debris flow originated uphill near the tree line and proceeded toward the large tree then branched into two debris flows around the tree. Based on reports, both sides of the debris flow stopped prior to reaching the road at the bottom of the hill and the debris flow to the left when looking up hill moved farther than the path on the right [3]. To compare this

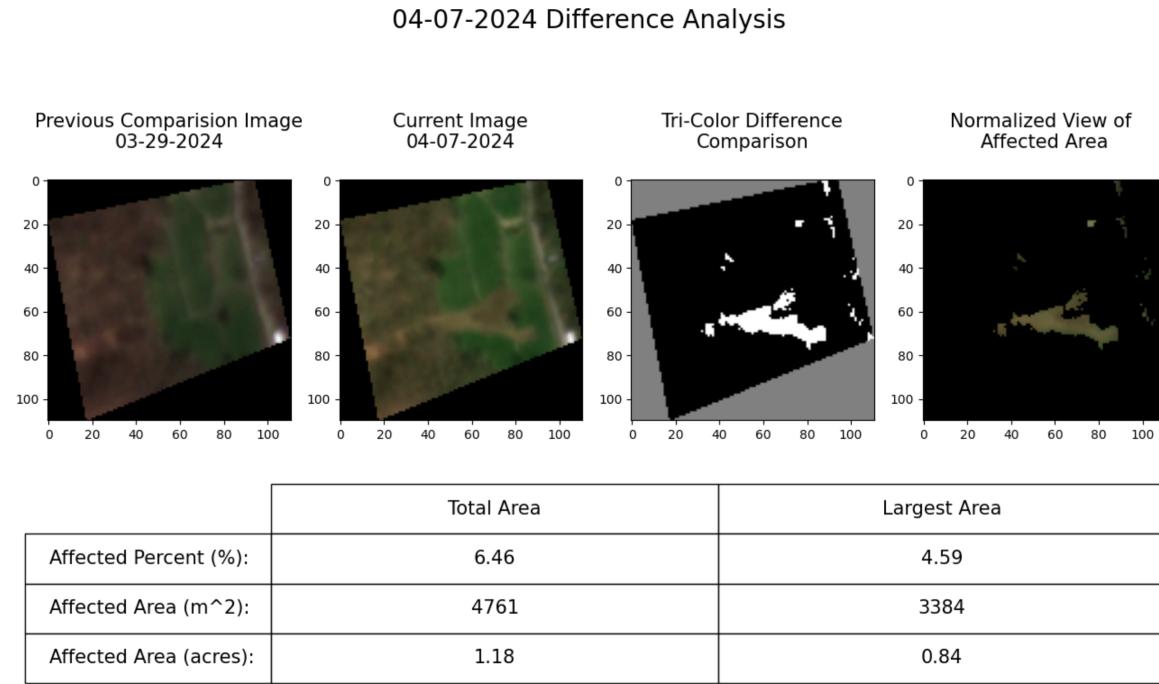


Figure 4.6: Summary figure generated to explain the comparison conducted on the Wheeling, WV debris flow images.

image and description to the satellite imagery analysis an inverse of the image difference was multiplied by the current image resulting in an image that shows debris flow location as black on the RGB satellite image, this is shown in Figure 4.8.

In Figure 4.8 the blue line follows the treeline, the orange line follows the road, and the red circle marks the location of the main tree. From this image it is clear that the main region of change, marked in black, captures the shape of the debris flow. It shows that the debris flow began near the treeline and flowed toward the road and split before the main tree was hit. This matches the picture of the debris flow, Figure 4.7, and the reported description of the debris flow. A portion of the debris slide appears to not have been detected due to its close proximity to the original dirt path, this is difficult to confirm though as the ground truth comparison image does not provide a bird's eye view of the debris flow. Despite this, this



Figure 4.7: Image of debris flow in Wheeling, WV [3].

example provides strong evidence towards the validity of the software to detect significant changes in satellite imagery.

Other extraneous changes were detected in the satellite image in addition to the main debris flow. These were most likely due small land movement or other changes due to the heavy rainfall. However, this cannot be verified as there is no ground truth imaging for these areas.

4.2.3 Baseline Example

To validate that the software was not detecting changes where none occurred, the third and final test location was chosen in an area with no expected change. The area chosen was a slightly hilly field located locally in Blacksburg, Virginia. In-person visual inspection of the field showed no significant signs of land movement or change in general.



Figure 4.8: Satellite image post debris flow with regions of change masked in black. The blue line marks the treeline, the orange line marks the road, and the red circle marks the main tree.

Six days worth of satellite imagery data over the chosen field was collected consisting of five previous images and one current comparison image. These six images were analyzed using the process described in Section 3.3 and the resulting summary image was generated, which is shown in Figure 4.9.

As expected no differences were detected in the image, demonstrating the software's ability to not detect false positives. While difference detection aspect of this test case was successful this test revealed an issue with the visualized version of the image. The visualized image, shown in Figure 4.10a, is extremely bright and white saturated. This is not representative

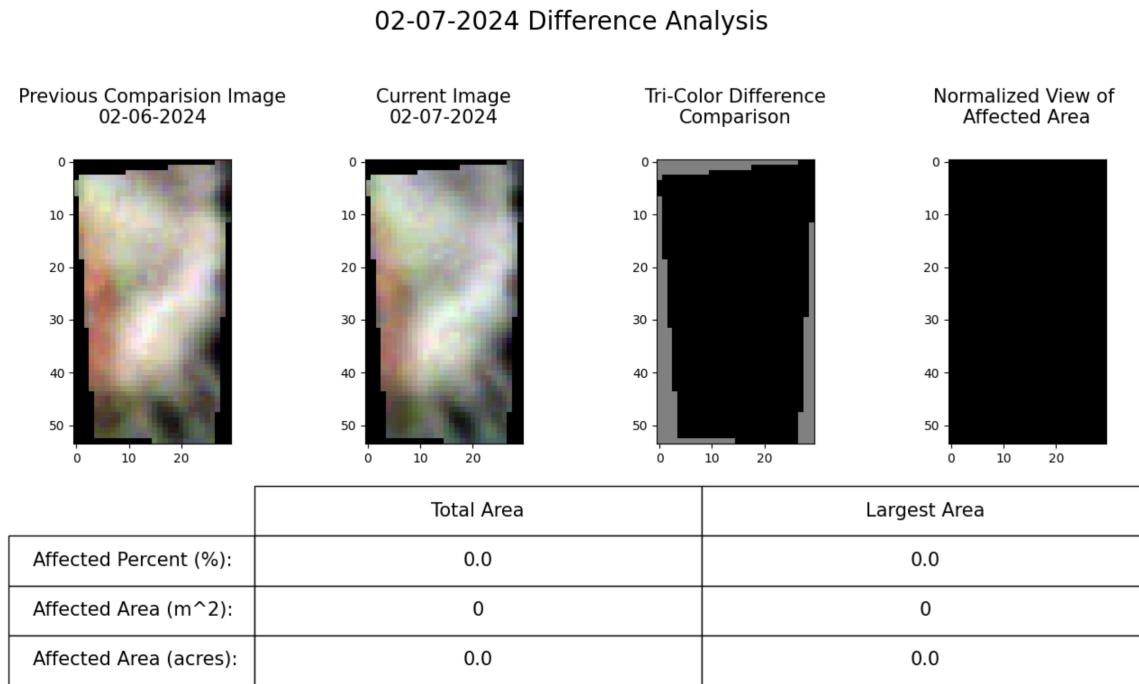


Figure 4.9: Summary figure generated to explain the comparison conducted on the baseline images from a field in Blacksburg, VA.

of the field in-person which is green. The raw satellite image of the field, shown in Figure 4.10b, is closer to depicting the true colors of the field but it is still too dark and too brown.

This particular image is difficult to normalize due to its small area and near uniform color. The visualization code works by normalizing each color band and stacking the norm bands back together. To normalize a band the maximum and minimum values are first found then each pixel is adjusted by computing the true pixel value minus the minimum pixel value all over the maximum pixel value minus the minimum pixel value. This normalization process expands or contracts the given pixels values to a new evenly spread range. This process works well in images that have some significant spread of colors and just need to be generally lightened or darkened. It does not work well when the spread of colors is extremely small as then it will spread the colors much more than would occur naturally.

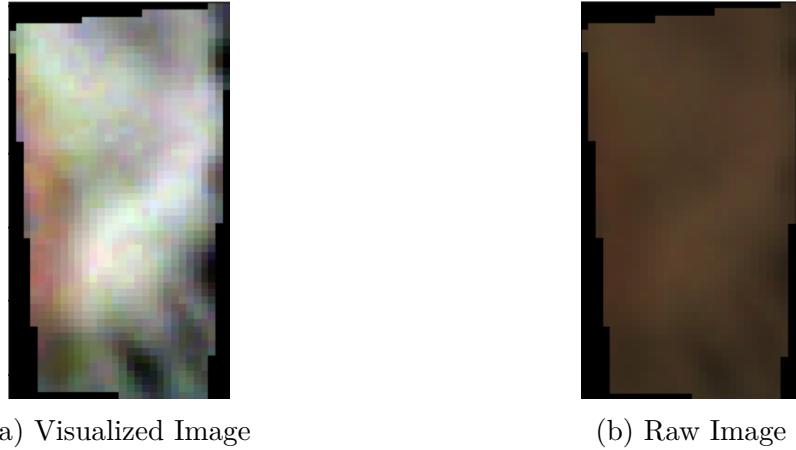


Figure 4.10: Comparison of visualized and raw satellite images for the baseline area.

To prevent normalization errors a larger area with some variety in colors would be best to select, if that is not available looking at the raw image may still give a better idea of what the ground truth colors look like. Regardless of the visualization difficulties found in this example the main purpose, to validate that false positives do not occur in areas where no change is expected, was achieved.

4.2.4 HazMapper Comparison

HazMapper was successfully used to identify the location of a debris flow in Breathitt County, KY post storm in July 2022 [19]. A repeat search of the area post storm using HazMapper is shown in Figure 4.11, with the red box surrounding the debris flow area.

To compare to the success of HazMapper, satellite images over the area of the debris flow were collected to analyze for significant changes. Five pre-storm images as well as the first



Figure 4.11: Screenshot of HazMapper with red box surrounding debris flow area in Breathitt County, KY [4].

post-storm image were collected and the script to analyze satellite images (described in Section 3.3) was run, the result of the comparison is shown in Figure 4.12.

The a comparison of the results from HazMapper and from the newly developed program are shown in Figure 4.13. Figure 4.13a shows the result from HazMapper with the debris flow indicated by the red box and Figure 4.13b shows the tri-color difference comparison generated from the software presented in Section 3.3 with the debris flow indicated by the red box.

To create a better comparison the HazMapper result and the result from the proposed difference analysis method were overlaid with each other, this is shown in Figure 4.14. This shows that both methods were able to accurately detect the debris flow and shows that the results from the software developed in Section 3.3 align with the results generated with HazMapper.

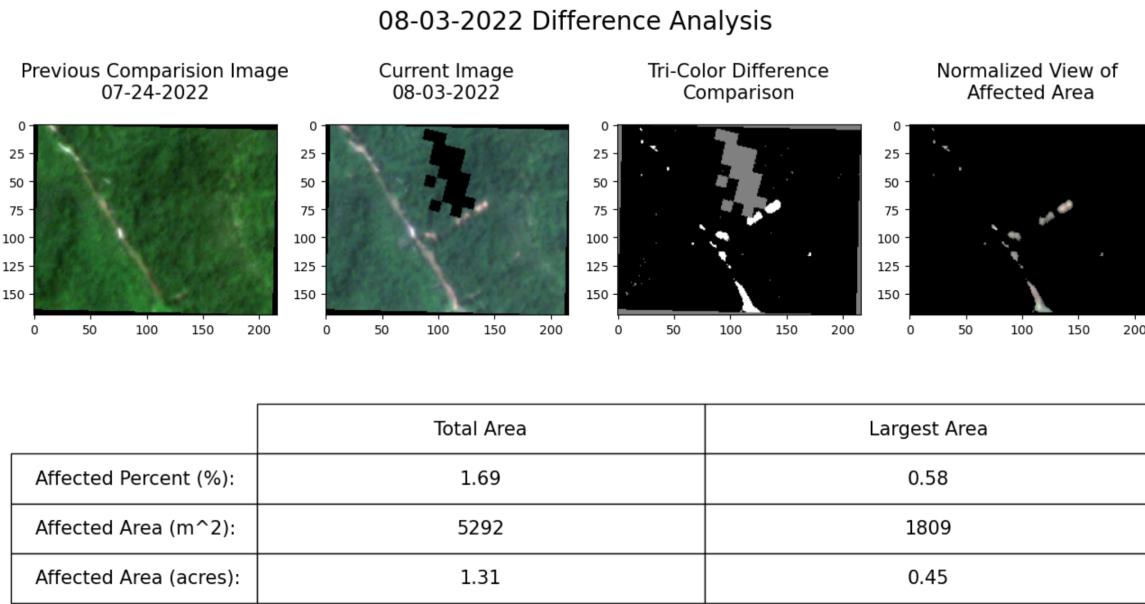
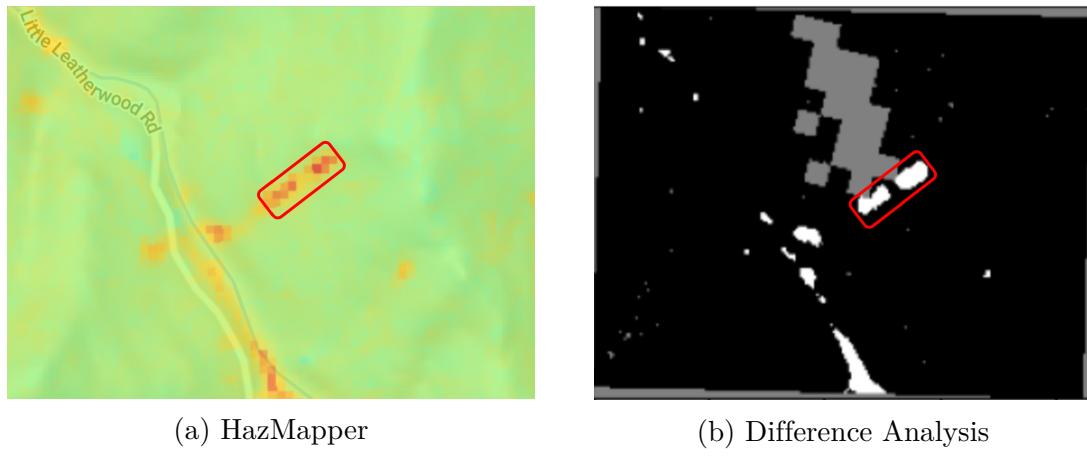


Figure 4.12: Summary figure generated to explain the comparison conducted on the Breathitt County, KY images.

This demonstrates the ability of the developed software to match the performance of the best widely available difference detection source for natural disasters.



(a) HazMapper

(b) Difference Analysis

Figure 4.13: Results from HazMapper and the proposed difference analysis method over a debris flow in Breathitt County, KY. The red boxes indicate the debris flow area.

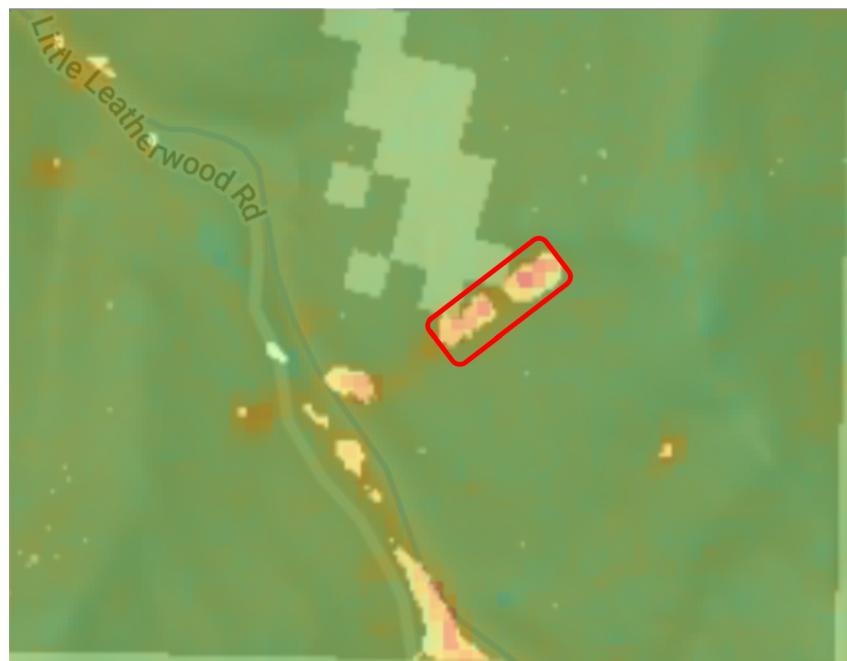


Figure 4.14: Overlay of HazMapper result and proposed difference analysis result on the Breathitt County, KY images, the red box indicates the debris flow area.

Chapter 5

Summary & Conclusions

This paper presented a way to continuously, and autonomously monitor areas for changes using satellite imagery. The system used to complete this process was broken into four main components. The first component specified the desired AOI for monitoring. Two ways to specify the AOI were presented, one used an active contour snake model to identify dirt pathways, the other used geojson.io [1] to select points on a map to create a polygon surrounding the AOI and generated a geojson file to save the relevant AOI information. Ultimately, the geojson.io method was selected to specify the AOI. This method was tested by generating various AOIs with complicated geometries and many points. It proved easy to use and generated the data required for subsequent steps.

The second main component of the software created a request for relevant satellite imagery data. Image filters were created to specify the AOI that needed to be captured in the images, the date range to monitor, and a maximum cloud cover threshold was set. Additionally, the types of satellite imagery data requested were specified and the upload location in GCS was designated. Finally, the request for imagery was submitted using the Planet Labs Subscriptions API [38]. This process was validated through multiple test subscriptions, requested imagery was consistently delivered to the specified upload location on a regular basis as more data became available.

The third main component contained the bulk of the computations, it analyzed the satellite imagery data to determine if any significant changes were detected. First, non-repeat imagery data was downloaded from GCS to a local folder and the six most recent images were selected for processing. Next, these images were masked to ensure that only clear pixels, ones that were not obscured by clouds, shadows, etc., were used in the comparison. A baseline variance was then created by calculating the range of pixel values at each pixel location across the five previous images. This, combined with a minimum allowable difference, was used to create a variable threshold for significant changes. The current image was then differenced with the best previous image and compared to the variable threshold to determine areas of significant change. Finally, a summary figure explaining the comparison was created, it was then emailed to the user if significant changes were detected.

This component was validated through four test cases. The first two test cases were of areas where significant change had occurred, the third test case was a stable area with no expected changes, the fourth test case compared the ability of the developed program to detect changes to the current best natural disaster detection program, HazMapper. In the first test case, significant changes due to flooding were detected in the appropriate areas when compared to UAV imagery of the same location. Additionally, no change was found in areas that covered structures such as houses that had not been affected. The second test case identified the location of a debris flow. The shape and trajectory of the changes detected aligned with ground reports and imagery of the slide. The third test case analyzed a stable field for changes, as expected none were detected. Finally, the fourth test case compared the ability to detect a debris flow between the developed software and HazMapper. The developed software was able to accurately identify the location of the debris flow and matched the results generated from HazMapper.

The final component was to set up the system to autonomously retrieve and analyze the data as well as notify the user when significant changes were detected. To do this a task was created in Windows Task Scheduler, each day and on user login it was set to execute the python script to analyze satellite imagery. It was run with pythonw.exe which allowed the script to be run in the background with no interaction from the user. This process was validated using a test script that simulated the functions of analyzing the satellite images without needing to actually look at data and determine a difference. The test script was successfully scheduled and was able to automatically run and send an email with an attachment.

Overall, the system to autonomously monitor areas for significant change worked as intended and was successfully validated through test cases. The system also demonstrated similar change detection ability when compared to HazMapper but also provided added functionality with autonomous monitoring. Applications for this system include monitoring buried pipelines in landslide prone areas, monitoring rural roads at risk of flooding, monitoring remote shorelines for oil spills, and monitoring protected forest areas for illegal logging. This system could be used for many more applications other than those listed, it could be implemented almost anywhere to look for any type of significant change to the environment. The system is also easy to use, minimal setup is required to initiate the monitoring process, after that the system can theoretically run indefinitely without additional user input.

While the system works well there is room for much improvement in future work. The first suggested area of improvement would be to conduct further research and tuning on the parameters set for the variable threshold matrix. The number of previous images used to establish the threshold as well as the minimum allowable difference percentage of the color scale could be tuned to establish optimized comparison metrics. Additionally, the time of

day of re-imaging could be evaluated to see if it had any impact on the comparison.

Another area of improvement would be to store the downloaded images and run the script to analyze them on a virtual computer or server. This would allow for more data to be downloaded without taking up space on the user's computer. It would also allow the program to run more frequently and not depend on if the computer housing the script is logged in.

Another area of improvement would be to implement a generative adversarial network (GAN) which could be used to help detect changes and classify whether or not it was actually significant. Other machine learning techniques could also be used to classify the detected changes based on their likely cause. This would be a great feature as it could give the user more insight on the issue before reaching the scene. Response would look very different if the detected change was caused by a landslide versus a large parked vehicle. Much more research would be needed to classify the changes but it could provide key information.

A final suggested area of improvement would be to have an interactive difference map with the ability to generate autonomous drone missions. The user could be shown the difference map and select the difference region they want to image. Once the region is selected the coordinates of the affected pixels could be retrieved and used to generate an autonomous drone mission. This mission could then be uploaded to a drone and flown without much required skill. The aerial imagery from the drone could then be inspected to get a better idea of the extent of the damages. This would be particularly useful in areas where the damages have made it unsafe to complete a visual inspection.

These suggested improvements would increase the ability to monitor regions for significant changes and allow for better response to disasters. However, the current system still serves as a useful tool to monitor regions for change without requiring much skill from the user. As the impact of extreme hydrological events increase, monitoring at risk areas areas, such as with the system presented in this paper, becomes evermore important.

Bibliography

- [1] Tom MacWright. geojson.io: powered by mapbox. URL <https://geojson.io/#map=2/0/20>.
- [2] Daniel Whitehurst, Kunal Joshi, Kevin Kochersberger, and James Weeks. Post-flood analysis for damage and restoration assessment using drone imagery. *Remote Sensing*, 14(19):4952, Oct 2022. doi: 10.3390/rs14194952.
- [3] Megan Guza. At a cemetery in west virginia, a massive landslide wiped out hundreds of headstones. what happens next isn't clear., Apr 2024. URL <https://www.post-gazette.com/news/weather-news/2024/04/07/wheeling-west-virginia-mt-zion-cemetery-landslide/stories/202404060063>.
- [4] C. M. Scheip and K. W. Wegmann. Hazmapper: a global open-source natural hazard mapping application in google earth engine. *Natural Hazards and Earth System Sciences*, 21(5):1495–1511, 2021. doi: 10.5194/nhess-21-1495-2021.
- [5] Kevin E Trenberth. The impact of climate change and variability on heavy precipitation, floods, and droughts. *Encyclopedia of hydrological sciences*, 17:1–11, 2005. doi: 10.1002/0470848944.hsa211.
- [6] Zunaira Asif, Zhi Chen, Chunjiang An, and Jinxin Dong. Environmental impacts and challenges associated with oil spills on shorelines. *Journal of Marine Science and Engineering*, 10(6):762, May 2022. doi: 10.3390/jmse10060762.
- [7] Fatima Khalid, Muhammad Babar Taj, Asma Jamil, Huda Kamal, Tahira Afzal, Muhammad Jamshed Iqbal, Tahseenullah Khan, Muhammad Ashiq, Ahmad Raheel,

- Muhammad Sharif, and et al. Multiple impacts of illegal logging: A key to deforestation over the globe. *Biomedical Journal of Scientific & Technical Research*, 20(5), Aug 2019. doi: 10.26717/bjstr.2019.20.003519.
- [8] Arthur P. Cracknell. The development of remote sensing in the last 40 years. *International Journal of Remote Sensing*, 39(23):8387–8427, Dec 2018. doi: 10.1080/01431161.2018.1550919.
- [9] Planet Labs PBC. Planet imagery product specifications, Dec 2023. URL https://assets.planet.com/docs/Planet_Combined_Imagery_Product_Specs_letter_screen.pdf.
- [10] Planet Labs PBC. Education and research program, . URL <https://www.planet.com/markets/education-and-research/#:~:text=Any%20university%2Daffiliated%20student%20faculty,university%20email%20address%20is%20required>.
- [11] Planet Labs PBC. Planet application program interface: In space for life on earth, 2024. URL <https://api.planet.com>.
- [12] Konstantinos N Topouzelis. Oil spill detection by sar images: Dark formation detection, feature extraction and classification algorithms. *Sensors*, 8(10):6642–6659, Oct 2008. doi: 10.3390/s8106642.
- [13] Amber Bonnington, Meisam Amani, and Hamid Ebrahimi. Oil spill detection using satellite imagery. *Advances in Environmental and Engineering Research*, 2(4):1–8, Aug 2021. doi: 10.21926/aeer.2104024.
- [14] Paweł Tysiąc, Tatiana Strelets, and Weronika Tuszyńska. The application of satellite image analysis in oil spill detection. *Applied Sciences*, 12(8):4016, Apr 2022. doi: 10.3390/app12084016.

- [15] Akanksha Sharma and Kamal Kumar Sharma. A review on satellite image processing for landslides detection. *Artificial Intelligence and Machine Learning in Satellite Data Processing and Services*, 970:123–129, 2023. doi: 10.1007/978-981-19-7698-8_14.
- [16] Devara Meghanadh, Vipin Kumar Maurya, Manish Kumar, and Ramji Dwivedi. Automatic detection of landslides based on machine learning framework. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, page 8460–8463, Jul 2021. doi: 10.1109/igarss47720.2021.9553341.
- [17] David Holden, Daniele Cerin, Karola Mäkitaloavola, Fredrik Ersholm, and Jonny Sjöberg. Who needs resolution anyway? three satellite insar case studies comparing coincident and contemporary datasets. *SSIM 2021: Second International Slope Stability in Mining*, page 211–226, Oct 2021. doi: 10.36487/acg_repo/2135_12.
- [18] David P.S. Bekaert, Alexander L Handwerger, Piyush Agram, and Dalia B Kirschbaum. Insar-based detection method for mapping and monitoring slow-moving landslides in remote regions with steep and mountainous terrain: An application to nepal. *Remote Sensing of Environment*, 249, Nov 2020. doi: 10.1016/j.rse.2020.111983.
- [19] Matt M. Crawford, Zhenming Wang, Seth Carpenter, Jonathan Schmidt, and Hudson J. Koch. Reconnaissance of landslides and debris flows associated with the july 2022 flooding in eastern kentucky. *Kentucky Geological Survey*, 13(5):14, Jan 2023. doi: 10.13023/kgs.ri56.13.
- [20] Adam B Smith. 2023: A historic year of u.s. billion-dollar weather and climate disasters, Jan 2024. URL <https://www.climate.gov/news-features/blogs/beyond-data/2023-historic-year-us-billion-dollar-weather-and-climate-disasters>.
- [21] J.Y. Zheng, B.J. Zhang, P.F. Liu, and L.L. Wu. Failure analysis and safety evaluation

- of buried pipeline due to deflection of landslide process. *Engineering Failure Analysis*, 25:156–168, Oct 2012. doi: 10.1016/j.engfailanal.2012.05.011.
- [22] Vassilis Marinos, Georgios Stoumpos, and Costas Papazachos. Landslide hazard and risk assessment for a natural gas pipeline project: The case of the trans adriatic pipeline, albania section. *Geosciences*, 9(2):61, Jan 2019. doi: 10.3390/geosciences9020061.
- [23] Mike G. Winter, Barbara Shearer, Derek Palmer, David Peeling, Clare Harmer, and Jonathan Sharpe. The economic impact of landslides and floods on the road network. *Procedia Engineering*, 143:1425–1434, 2016. doi: 10.1016/j.proeng.2016.06.168.
- [24] E.T. Bowman. Small landslides – frequent, costly and manageable. *Landslide Hazards, Risks, and Disasters*, page 439–477, 2022. doi: 10.1016/b978-0-12-818464-6.00016-0.
- [25] NOAA. How long is the u.s. shoreline?, Jul 2010. URL <https://oceanservice.noaa.gov/facts/shorelength.html>.
- [26] B. Teodosio, A. Al-Taie, E. Yaghoubi, and P. L. Wasantha. Satellite imaging techniques for ground movement monitoring of a deep pipeline trench backfilled with recycled materials. *Remote Sensing*, 15(1):204, Dec 2022. doi: 10.3390/rs15010204.
- [27] V. Morocho, D. A. España, A. C. Serrano, R. Achig, and J. Crompvoets. Mass movements detection from uav images analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13:461–467, Jun 2019. doi: 10.5194/isprs-archives-xlii-2-w13-461-2019.
- [28] P. C. Pesáñez. Land movement detection from uav images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2022:1071–1076, May 2022. doi: 10.5194/isprs-archives-xliii-b2-2022-1071-2022.

- [29] Nishu Singla. Motion detection based on frame difference method. *International Journal of Information & Computation Technology*, 4(15):1559–1565, 2014.
- [30] M Sushma Sri. Object detection and tracking using klt algorithm. *International Journal of Engineering Development and Research*, 7(2):542–545, 2019.
- [31] Richard J Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: A systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, Mar 2005. doi: 10.1109/tip.2004.838698.
- [32] D. Chaudhuri, N. K. Kushwaha, and A. Samal. Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(5):1538–1544, Jun 2012. doi: 10.1109/jstars.2012.2199085.
- [33] Yucong Lin and Srikanth Saripalli. Road detection from aerial imagery. *2012 IEEE International Conference on Robotics and Automation*, May 2012. doi: 10.1109/icra.2012.6225112.
- [34] M. Mokhtarzade and M.J. Valadan Zoej. Road detection from high-resolution satellite images using artificial neural networks. *International Journal of Applied Earth Observation and Geoinformation*, 9(1):32–40, Feb 2007. doi: 10.1016/j.jag.2006.05.001.
- [35] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, Jan 1988. doi: 10.1007/bf00133570.
- [36] W. Neuenschwander, P. Fua, G. Székely, and O. Kübler. From ziplock snakes to velcrotm surfaces. *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, page 105–114, 1995. doi: 10.1007/978-3-0348-9242-1_11.

- [37] Scott Morris and Kobus Barnard. Finding trails. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2008. doi: 10.1109/cvpr.2008.4587819.
- [38] Planet Labs PBC. Planet subscriptions api, . URL <https://api.planet.com/subscriptions/v1>.

Appendices

Appendix A

GitHub Link for Code

<https://github.com/agr2022/Satellite-Monitoring>