SAN DIEGO STATE UNIVERSITY AND

UNIVERSITY OF CALIFORNIA

Santa Barbara

Improving disaster response with aerial imagery through UAS-based image acquisition and analysis, artificial intelligence, and timeliness assessment

A Dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Geography

by

Andrew Christopher Loerch

Committee in charge:

Professor Douglas Stow, Co-chair

Professor David Siegel, Co-chair

Professor Atsushi Nara

Professor James Frew

September 2022

The dissertation of Andrew Christopher Loerch is approved.

_____

James Frew

_____

Atsushi Nara

_____

David Siegel, Committee Co-Chair

_____

Douglas Stow, Committee Co-Chair

September 2022

Improving disaster response with aerial imagery through UAS-based image acquisition and assessment, artificial intelligence, and timeliness assessment

ACKNOWLEDGMENTS

Drs. Doug Stow (Committee Chair), David Siegel, James Frew and Atsushi Nara - For providing guidance and feedback in the proposal and dissertation, and supporting my academics despite the time, distance, and Covid.

San Diego State University's Geography Department – The administrative staff, leadership, and many graduate students through the years provided academic, financial (as a TA), Finch Scholarship and social support.

San Diego State University's Research Foundation – Provided financial support when my TA contract ended, to improve and commercialize (through the university's Technology Transfer Office) the software I developed for this research

ChangeAerial, LLC – For purchasing the DJI Matrice 300 and making it available to the study

Pete Coulter – For piloting the UAS and providing additional support throughout the study

Garret Joe – My loving, caring, and supportive husband of 19 years. Through countless hours of classes, research activities, teaching responsibilities, and missed anniversaries, Garret continued to support and push for my success.

VITA OF ANDREW CHRISTOPHER LOERCH
September 2022

EDUCATION

Bachelor of Science in Geographic Information Science, University of New Mexico, May 2013 (summa cum laude)
Master of Science in Geography and Environmental Management, University of New Mexico, August 2016
Doctor of Philosophy in Geography, University of California, Santa Barbara, September 2022 (expected)

PROFESSIONAL EMPLOYMENT

2014-2016: Research Assistant, Department of Geography, University of New Mexico
2016-2020: Teaching Associate, Department of Geography, San Diego State University
2021-Present: Research Analyst, San Diego State University Research Foundation

PUBLICATIONS

Zhang, Su, Christopher D. Lippitt, Susan M. Bogus, Andrew C. Loerch, and Jennie O. Sturm. "The accuracy of aerial triangulation products automatically generated from hyper-spatial resolution digital aerial photography." Remote Sensing Letters 7, no. 2 (2016): 160-169.

Loerch, Andrew. "Modeling the Timeliness of Airborne Remote Sensing Data." Unpublished thesis submitted in partial fulfillment of the requirements for the Masters of Science degree in Geography and Environmental Management, University of New Mexico (2016).

Stow, Douglas A., Christopher D. Lippitt, Lloyd L. Coulter, and Andrew C. Loerch. "Towards an end-to-end airborne remote-sensing system for post-hazard assessment of damage to hyper-critical infrastructure: research progress and needs." International Journal of Remote Sensing 39, no. 5 (2018): 1441-1458.

Loerch, Andrew C., Gernot Paulus, and Christopher D. Lippitt. "Volumetric Change Detection with using Structure from Motion–The Impact of Repeat Station Imaging." GI_Forum 2018, 6: 135-151.

Uyeda, Kellie A., Kelsey K. Warkentin, Douglas A. Stow, John F. O'Leary, Rachel A. Snavely, Julie Lambert, Leslie A. Bolick, Kimberly O'Connor, Bryan Munson, and Andrew C. Loerch. "Vegetation mapping using hierarchical object-based image analysis applied to aerial imagery and lidar data." Applied vegetation science 23, no. 1 (2020): 80-93.

Warkentin, Kelsey, Douglas Stow, Kellie Uyeda, John O'Leary, Julie Lambert, Andrew Loerch, and Lloyd Coulter. "Shrub Fractional Cover Estimation and Mapping of San

Clemente Island Shrubland Based on Airborne Multispectral Imagery and Lidar Data." Remote Sensing 12, no. 21 (2020): 3608.
Plummer, Matthew, Douglas Stow, Emanuel Storey, Lloyd Coulter, Nicholas Zamora, and Andrew Loerch. "Reducing Shadow Effects on the Co-Registration of Aerial Image Pairs." Photogrammetric Engineering & Remote Sensing 86, no. 3 (2020): 177-186.

Mirka, Blair, Douglas A. Stow, Gernot Paulus, Andrew C. Loerch, Lloyd L. Coulter, Li An, Rebecca L. Lewison, and Lena S. Pflüger. "Evaluation of thermal infrared imaging from uninhabited aerial vehicles for arboreal wildlife surveillance." Environmental Monitoring and Assessment 194, no. 7 (2022): 1-15.

AWARDS

Academic Scholarship, American Society of Photogrammetry and Remote Sensing – Rocky Mountain Region, 2014

Outstanding Graduate Research, Department of Geography, University of New Mexico, 2016

Fellowship, Marshall Plan Foundation, 2016

William and Vivian Finch Scholarship in Remote Sensing, Department of Geography, San Diego State University, 2019

FIELDS OF STUDY

Major Field: Geographic Information Science

Studies in Remote Sensing with Professors Douglas Stow, Christopher Lippitt

ABSTRACT

Improving disaster response with aerial imagery through UAS-based image acquisition and assessment, artificial intelligence, and timeliness assessment

by

Andrew Christopher Loerch

Aerial imagery, as a useful tool for emergency management and response, still poses challenges to its effective (Joyce, Wright, et al. 2009). The challenges include information quality, accuracy, and the timeliness of information delivery (Joyce, Wright, et al. 2009). This research utilizes and updates the Remote Sensing Communication Model (RSCM) to configure time-sensitive remote-sensing-systems based on unpiloted aerial systems (UAS) and machine-learning-based damage detection. The UAS (DJI Mavic 1 and DJI Matrice 300) configurations' navigation accuracies are tested with repeat station imaging (RSI) and traditional imaging. Utilizing real-time-kinematic corrections, the DJI Matrice 300 flown with the RSI method at six sites representing critical infrastructure at San Diego State University is shown to have more accurate repeated navigation to camera stations (0.16m vs. 0.21m for traditional imaging) and multi-date image pairs with the RSI method are shown to have better image co-registration mean absolute error (MAE) accuracy (2.2 pixels MAE vs. 4.4 pixels MAE). The Mask R-CNN machine-learning model (He et al. 2017) evaluated on bitemporal layer-stacked images detected damage (cracks) with a better mean intersection over union (mIoU) when used with the RSI images (83.7% mIoU vs. 72.5% mIoU). A

customized convolutional neural network (CNN) and recurrent neural network (RNN) were evaluated using the RSI images, with the RNN having the higher accuracy (98.4% overall accuracy vs. 96.9% overall accuracy). Using the acquisition and newly demonstrated method of analyst capacity within the RSCM, the traditional imaging method and DJI Matrice 300 configuration had the highest timeliness capacity (1.02×10^7 bits m^{-1} s^{-1} vs. 7.40×10^6 bits m^{-1} s^{-1}) of acquisition, and image co-registration based on the traditional imaging method had the highest co-registration analyst capacity (1.30×10^7 bits m^{-1} s^{-1} vs. 1.16×10^7 bits m^{-1} s^{-1}). Of the three machine-learning models, the CNN had the highest analyst capacity and the RNN had the next highest (9.00×10^7 bits s^{-1} vs. 6.68×10^7 bits s^{-1}).

TABLE OF CONTENTS

I. Introduction

Aerial imaging and digital image processing are critical disaster response tools used throughout the disaster management cycle for saving lives and identifying damaged roads, bridges and buildings (Joyce, Wright, et al. 2009). The application of remote sensing for post-hazard damage assessment can provide valuable and timely information at high spatial resolutions for areas subject to disasters, where direct observation may be hazardous or untimely. Still, the use of aerial imagery for disaster response can be challenging. Some requirements for the use of aerial imagery by emergency managers are: 1) making the correct choice of imagery and data collection methods (i.e., piloted systems, unmanned systems, traditional photogrammetry collections versus repeat station imaging, standard red/green/blue imagery versus band ratios and other transformations), 2) determining appropriate image processing methods, affecting accuracy and the breadth of information that can be gleaned about the detected damage, and 3) minimizing the amount of time required from requesting imagery to the production of maps with actionable information. This research exploits unmanned aerial systems (UAS), repeat station imaging (RSI), and convolutional and recurrent neural networks as solutions to these challenges.

The research addresses the three aforementioned requirements using geographic theory as it pertains to emergency management through the application of remote sensing and computer science. Based on feedback from emergency managers, previously interviewed as part of National Science Foundation award CMMI-1360041, areas representing critical infrastructure (i.e., bridges, roads, dams, hospital buildings, prison complexes) comprise the features of interest. This research focuses on signs of damage symptomatic of earthquakes

(i.e., cracking, chipping, buckling, subsidence and collapse). Due to limitations in data availability for the features of interest, manifestations of damage are artificially added or removed from imagery using photography editing software, and features are chosen that resemble critical infrastructure. The following technical and methodological objectives of this research are:

1. Development of relatively easy to use image processing tools for pre-processing imagery, detecting and assessing damage and generating maps;
2. Comparisons of the accuracies of damage detection and assessment from different acquisition methods and artificial intelligence algorithms; and
3. Comparisons of rapid response options for acquiring and assessing images and maps having high accuracy and timeliness.

A. Considerations for Remote Sensing of Damage after Extreme Events

Most formal disaster management institutions recognize and use a structured framework called the Disaster Management Cycle (DMC) (Cutter 2003; Laben 2002; Taubenböck and Post 2008) . This cycle consists of six phases, as illustrated in Figure I.1. The pre-disaster and intra-disaster phases are: reconstruction, mitigation and preparedness. Th duration of these phases range from months to years in duration and emergency managers can prioritize higher spatial and spectral resolution imagery with greater accuracies, at the expense of information timeliness (Cutter 2003; Joyce, Belliss, et al. 2009; Lippitt, Stow, and Clarke 2014). The immediate intra-disaster phases are rescue, relief and recovery. These phases range from minutes to days in duration and while higher spatial and spectral resolution imagery with high data accuracy may be desirable in supporting rescue and relief objectives, compromises

are commonly made in order to minimize the time to information availability (Cutter 2003; Lippitt et al. 2014).

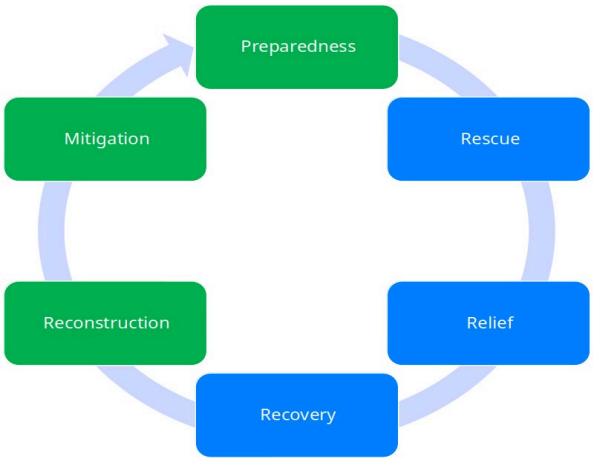

Figure I.1. Phases of the Disaster Management Cycle

As noted, the requirements of a remote sensing system for emergency management are heavily dependent upon the DMC phase. Specific needs stem from the type of disaster event and the expected manifestations and severity of damage (Cutter 2003; Joyce, Wright, et al. 2009; Lippitt and Stow 2015; Pham et al. 2014). Primary damage resulting from earthquakes, such as cracking, subsidence and collapsing of buildings, is better detected with high spatial resolution imagery (ground sample distance (GSD) < 5 m) and the inclusion of building height data (Ehrlich et al. 2009; Joyce, Belliss, et al. 2009; Pham et al. 2014). Similar requirements exist for detecting damage caused by landslides and volcanic activity (Joyce, Belliss, et al. 2009; Metternicht, Hurni, and Gogu 2005; Niethammer et al. 2012). Damage caused by flooding and wildfires can be detected with medium or higher spatial resolutions (GSD > 30 m) and spectral bands such as near infrared and thermal infrared (possibly in addition to red, green and blue bands). For the detection of damage to critical infrastructure,

where fine-scale damage detection is necessary, spatial resolutions of < 1 m may be required (Pham et al. 2014).

In addition to the spatial, spectral, and radiometric requirements, emergency management institutions have specific needs regarding timeliness for information availability (Cutter 2003; Lippitt et al. 2015). At the most fundamental level, emergency managers should have a method to estimate how long it will take from the moment a request for remote sensing data is issued to when a disaster response decision can be made using the information gathered (Lippitt et al. 2015). The Remote Sensing Communication Model describes this need as the sum time required for each of three capacities of a Time Sensitive Remote Sensing System (TSRSS), as originally developed by Lippitt et al. (2014). The "Acquisition Capacity" of a TSRSS is the time required to maneuver a platform and sensor into place and acquire the imagery of a scene. For satellite systems, this would be the "revisit" time (based on orbit characteristics and pointability of the sensor) and for aircraft it would be take-off, flight, and possibly landing time (Lippitt et al. 2014). As part of its estimation, the Acquisition Capacity also calculates the volume of data that is to be acquired by the sensor(s). The "Transmission Capacity" is the time required to transfer the data from the sensor(s) to an analyst and is impacted by the data channel(s)' transfer rates and data volume (Lippitt et al. 2014). The "Analyst Capacity" is the time required to process the data and create a presentation of the information upon which an emergency manager can act (Lippitt et al. 2014). The Analyst Capacity is also impacted by data volume, and the efficiency of the processing algorithm and available processing hardware (Lippitt et al. 2014). The timeliness of a TS-RSS, as modeled by the RSCM, is clearly impacted by the volume of data acquired. The data volume is a factor of the emergency management decision on the required spatial, spectral, and

radiometric needs (Lippitt et al. 2014). Increased resolution requirements for any of these remote sensing characteristics will result in increased data volume (Lippitt et al. 2014). As the RSCM illustrates, increased data volume leads to increased time to availability and decreased timeliness for decision making.

The cited literature for the DMC helps in understanding the spatial and temporal resolution requirements of emergency managers, while the RSCM places these requirements in a context that is useful for predicting timeliness of information delivery. Combined, these frameworks provide much of the structure necessary to this research. The technical objective (1) of the research fits into the Analyst Capacity of the RSCM. The technical and methodological objective (2) fits into the Acquisition Capacity and Analyst Capacity. The technical and methodological objective (3) uses the RSCM to assess the timeliness of RSS configurations.

B. Acquisition Methods for Aerial Imagery

Unmanned aerial systems (UAS) are currently being studied and used operationally for image acquisitions to address problems related to the timeliness of damage detection (Niethammer et al. 2012). For example, UAS are being used to monitor the Super-Sauze landslide progression, as "they can be rapidly deployed", and digital surface models with GSD $\geq$ 0.03 m can be generated (Niethammer et al. 2012).

Small UAS (sUAS) are capable of being deployed at lower-cost and with less technical training than manned aircraft or their larger unmanned counterparts, are capable of multi-spectral imaging, and their low-altitudes make them ideal for capturing data at hyper-spatial resolutions (Westoby et al. 2012; Zhang et al. 2015). UAS remote sensing for both rapid (within 24 hours) and real-time (imagery is processed and delivered as a disaster is

occurring) hazard response improves upon the manned and satellite platform timeliness delays of up to 72 hours. Following the Fukushima nuclear disaster, Tokai and Nihon universities in Japan began a project to demonstrate the capability of UAS based remote sensing in conjunction with satellite imagery to provide continuous (daytime) data about the disaster (Baltsavias et al. 2013).

sUAS generally can only lift light-weight imaging sensors, and the trend of increasing sensor spatial resolutions while simultaneously steadily decreasing in physical size and weight has meant that hyper-spatial ground resolutions can be achieved using consumer-off-the-shelf digital cameras (Colomina and Molina 2014). These systems, largely due to the altitudes at which they are safely flown, are capable of capturing images with spatial resolutions in the millimeter range (Smith, Chandler, and Rose 2009; Turner, Lucieer, and Watson 2012). The size and imaging sensor capabilities of these systems are expected to result in their increased use for scientific research and infrastructure and emergency management applications (Lippitt 2015).

The traditionally employed method for acquiring imagery (further referred to as 'traditional imaging' or non-RSI) is the use of flight lines with a frame-based sensor designed to capture images based upon either time or distance intervals (Jensen and Im 2007). This method of acquiring images ensures sufficient forward-lap and side-lap to ensure features are viewed from multiple perspectives. While differential parallax between imagery in the same observation permits aerial triangulation, it makes co-registration between subsequent observations a challenge (Coulter, Stow, and Baer 2003; Stow 1999). Parallax introduces variable relief displacement between observations (Slama, Theurer, and Henriksen 1980). The distortions in vertical features and co-registration errors between time sequential images

resulting from varied view geometries reduces the accuracy of change detection results (Coulter et al. 2003; Stow 1999; Stow et al. 2016).

RSI is an alternative method for acquiring imagery for use in change detection. RSI uses flight paths designed with global navigation satellite systems (GNSS) waypoints (i.e., x-y-z positions programmed for UAS flight navigation), and the imaging sensor is triggered at camera stations with specified angles based on these waypoints (Coulter et al. 2003; Lippitt et al. 2015). This acquisition method has been demonstrated to improve the co-registration accuracy of time-sequential images, a critical factor in achieving reliable image-based change detection (Coulter et al. 2003; Lee et al. 2008; Stow 1999). Provided the same sensor is used across acquisitions, the result is "multitemporal imagery with matched view geometry" (Lippitt et al. 2015). This method of image acquisition has been shown to result in horizontal spatial co-registrations between multitemporal image sets of between one to two pixels, even at high spatial resolutions (Coulter et al. 2003; Lippitt et al. 2015). Horizontal changes between image pairs and between two-dimensional orthomosaics has been the focus of existing literature that employs RSI (Coulter et al. 2003; Lippitt et al. 2015). Evidence from RSI literature suggests that damage detection can be performed more accurately using RSI than with traditional acquisition methods, even where damage occurs in three-dimensions (Loerch, Paulus, and Lippitt 2018). While piloted aircraft can perform RSI acquisitions, given either a suitable auto-pilot system or skilled pilot, UAS are specifically designed to perform these acquisitions (Loerch 2016).

Chapter 2 explores sUAS as an acquisition platform using both RSI and non-RSI acquisition methods. The assessment criteria are 1) accuracy of image co-registration for

multi-temporal image pairs based on RSI and non-RSI acquisitions, 2) accuracy of machine learning-based post-classification damage detection for the RSI versus non-RSI acquisitions.

C. Artificial Intelligence for Damage Detection and Assessment

Convolutional neural networks (CNN) are an emergent technology with potential for advancements in image segmentation, pattern recognition, and change detection (Long et al., 2015; Lv et al., 2015). CNNs and related machine-learning algorithms are shown to provide improved accuracies over other image segmentation and classification techniques such as object-based change detection and various supervised and unsupervised classifiers (Lv et al., 2015). CNNs utilize a fundamental digital image processing technique called "convolution". Images are convolved into their most basic components, and subsequently reconstructed with a set of automatically-generated and estimated parameters (Dumoulin and Visin, 2016). Parameters that lead to closer matching reconstructions get preferential weight and are transferred to the next convolution as starting parameters. Each subsequent convolution attempts to perform less smoothing and thus learn parameters for reconstructing higher-level, finer-scale features.

Most CNN literature focuses on single-date image classification, with only a few recent examples of their application to change detection. CNN-based change detection can be performed with post-classification analysis across multi-temporal datasets, as direct change detection with co-registered and stacked bi-temporal image pairs, and through Recurrent Neural Networks (RNNs). RNNs are designed for assessing changes in sequential data inputs, for example images over time (Lyu et al., 2016). Change objects and rulesets with the flexibility of being transferred to other purposes/types of change can be developed using RNNs and have been previously shown successful with satellite-derived imagery (Lyu, Lu,

and Mou 2016). Using CNNs and RNNs, changes can be detected and labeled semantically by type, and in the case of damage, severity (Lyu et al. 2016).

D. Study areas and general approach

The primary study area is the San Diego State University (SDSU) campus, with specific study sites selected to contain multiple features of interest representative of critical infrastructure. Table I.1 provides details on site-selection criteria.

Table I.1. Feature, damage, and view perspective criteria used for image collection

Feature Types and Descriptions	Damage Type	View Perspective
Building - Fowler Athletics Center - School of Music and Dance	roof & side cracks, subsidence, partial collapse	oblique
Bridge - Parking Structure 12 pedestrian crossing - San Diego Metropolitan Transit System light-rail track	surface & support structure cracks, subsidence, partial collapse	oblique
Road - Canyon Crest Drive - Aztec Circle Drive - East Campus Drive	surface cracks	nadir

Given the need for substantial quantities of data to be collected and established relationships with administrative and law-enforcement officials, all sites are located on the SDSU campus. Figure I.2 is a map showing the listed features.

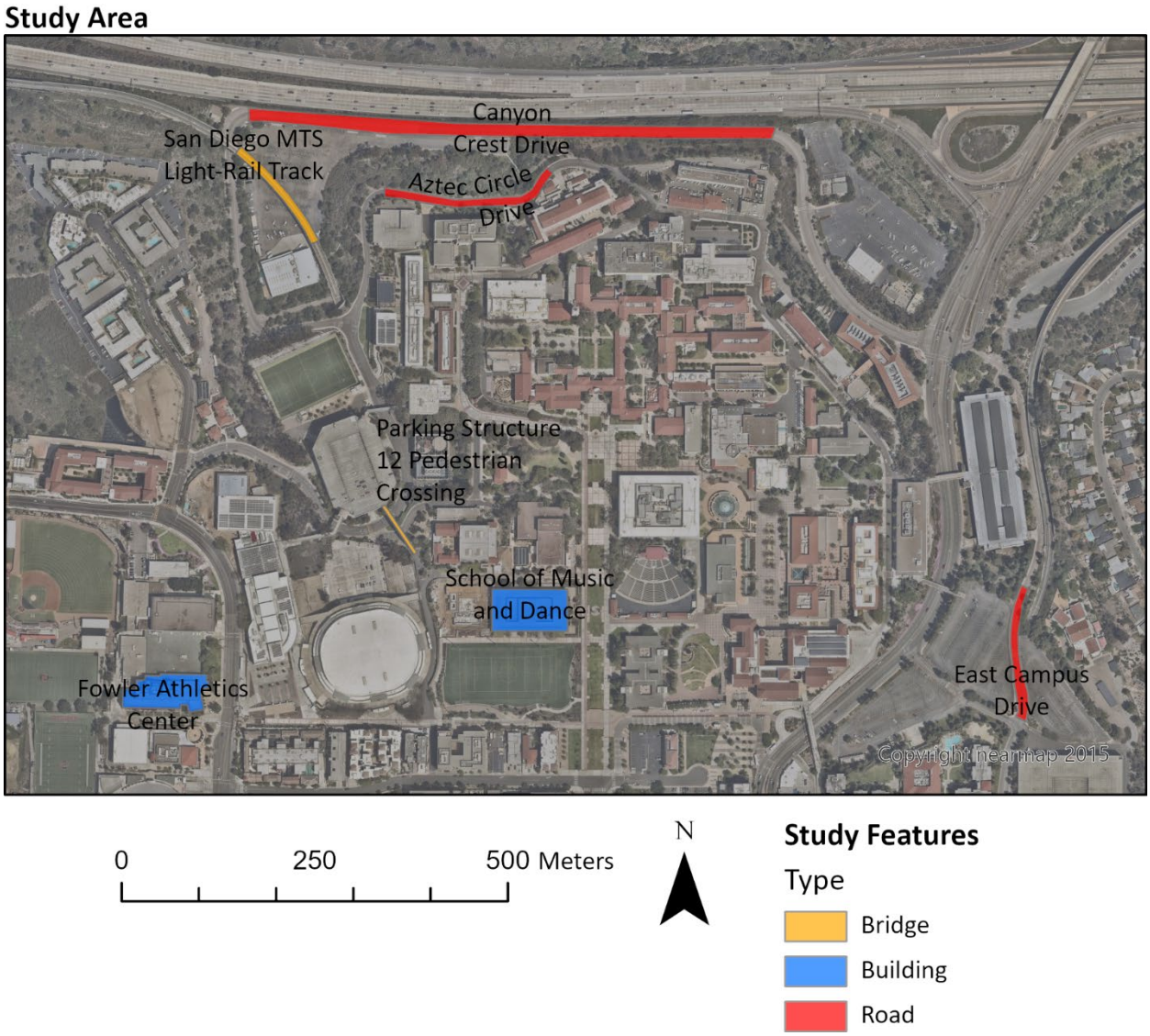

Figure I.2. An image map showing the chosen study sites within the SDSU campus.

The collection and processing of images requires specialized hardware components. Accurately repeated imaging of features requires a real-time-kinematic (RTK) global navigation satellite system (GNSS) with a ground-based unit placed on a surveyed control marker and a mobile RTK unit on the sUAS. Additionally, gimbal mounts with the capability of pointing a camera at specified angles are required for repeating oblique view geometries. The equipment meeting the above criteria and used in this research were an Emlid Reach RS2 RTK base unit and the DJI Matrice 300 sUAS with a Zenmuse H20 camera/gimbal

combination. A DJI Mavic Pro sUAS without RTK capability was also used in additional assessments of non-RSi acquisitions.

Image processing requires substantial computing resources; high-end graphics cards and memory availability are critical. Many image processing applications facilitate co-registering and transforming images, including options from ERDAS, ESRI, and Adobe. There are no software resources available, however, for performing the needed automatic co-registrations and coordinate transformations in-bulk while simultaneously providing accurately georeferenced image outputs for use with neural network classification. Furthermore, currently available implementations of neural networks in pyTorch, Keros, Caffe, and Tensorflow require substantial case-by-case edits of python-language scripts to account for changes in data types, classes and numbers of features, and the presentation of classification results. There are no available CNN/RNN packages that maintain spatial coordinates "out-of-the-box" to enable spatial feature locations and mapping. Because the timeliness of information delivery is a fundamental requirement of emergency management, any solution that is not highly automated or that requires the use of multiple software applications at different steps of processing will result in increases to processing and information delivery times.

The software application “Image Pre-Processing” (IMPP) was developed for this research to provide bulk image transformations, automatic co-registration of image pairs based on geographic proximity, and the selection of CNN/RNN models for damage detection, as shown in Figure I.3.

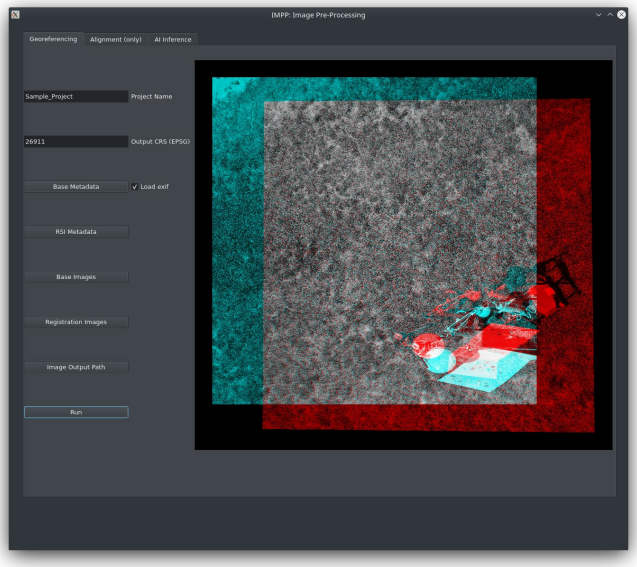

Figure I.3. The image co-registration and neural network damage detection software IMPP

Data pre-processing was performed on all collected images with two primary goals: 1) simulate damage, 2) co-register time-sequential image pairs. Damage simulation was performed using Adobe Photoshop for two-dimensional representations of damage and both Agisoft MetaShape and Blue Marble Global Mapper for three-dimensional damage representations. Geometric and radiometric image transformations was performed using IMPP. Additional details on the simulation of damage and transformation of images are provided in Chapter 2.

II. Evaluating the accuracy of sUAS navigation, image co-registration and CNN-based damage detection in the context of repeat station imaging

A. Background

Research on image change detection consistently shows image co-registration accuracy directly impacts the accuracy of change detection results (Stow 1999; Stow et al. 2016). RSI, which is shown to yield higher image co-registration accuracy than conventional implicit image registration approaches, when employed with image differencing, should be useful in CNN-based change detection (Stow 1999). Traditionally, feature detection, localization, and change detection have been performed in the remote sensing context using methods such as image differencing, object-based image analysis, supervised- and unsupervised-classifications with support vector machines, and earlier versions of artificial neural network algorithms (Atkinson and Tatnall 1997). More recently, researchers using remote sensing for environmental sciences and engineering problems have developed CNN models as means for achieving higher classification accuracies (Lyu et al. 2016; Pinheiro and Com 2014).

In the domain of remote sensing, CNNs have been applied to spaceborne and airborne imagery, with pre-trained and fine-tuned models yielding accurate results. Most published work in this domain has focused on the scene labeling problem, although Lyu, Lu, & Mou (2016) used an RNN to classify change features from high spatial resolution satellite imagery. The scene labeling problem, also referred to in literature as semantic segmentation and labeling, refers to accurate image classification by pixel or object (Chen et al. 2018; Long, Shelhamer, and Darrell 2015). Mimicking the geographic-based image change analysis (GEOBICA) approach with CNN learning, where co-registered bi-temporal image pairs are used to find change objects, should result in improvements to change (e.g., damage)

detection accuracy over post-classification change detection approaches (Stow 2009). This research employs a novel use of the Mask R-CNN model, designed for object detection and location in imagery and video frames, to bi-temporal imagery for the purpose of change detection (He et al. 2017).

B. Objectives

This chapter encompasses the following four objectives: 1) evaluate the capacity of low-altitude sUAS systems to repeat the positioning of image stations with high-precision and accuracy, 2) evaluate the impact of RSI versus non-RSI (i.e., traditional imaging) image acquisition techniques on co-registration of sUAS images acquired with nadir and oblique view perspectives, 3) develop a software application for automatic image co-registration and CNN-based detection of damage to critical infrastructure, and 4) compare the damage detection accuracy of a CNN employed with bi-temporal image pairs acquired with both RSI and non-RSI approaches. These objectives provide the basis for addressing the following research questions:

1. How accurately can an RTK GNSS repeatedly navigate a sUAS platform and trigger a camera at a specified waypoint (i.e., imaging station)?
2. How does the co-registration accuracy vary for RSI versus non-RSI acquisitions of sUAS imagery captured with nadir and oblique views?
3. What difference in classification accuracy of bi-temporal change objects is observed with a CNN for RSI and non-RSI co-registered images?

C. Methods

Features of interest for the SDSU study site were selected to represent critical infrastructure and include roads, buildings, and bridges. Two sUAS with different sized

sensors and focal length settings were used to acquire images of these features, using both RSI and non-RSI acquisition methods. The types of damage of interest, cracks, were simulated for the collected imagery. Subsequent-time (i.e., repeat-pass) images with the simulated damage were co-registered to baseline images. Some co-registered image pairs were used to train the CNN, and others used to detect damage. The accuracies of platform navigation, image co-registration, and resultant CNN damage detection were assessed for the RSI and non-RSI acquisition methods.

1. Data Requirements and Acquisition

The primary data acquired are repeated sUAS images of the study area and image-related metadata. This section outlines the required parameters for the imagery and the image acquisition methodology.

With surface cracks as the primary damage of interest, a ground sample distance (GSD) of 0.5 cm was chosen for all images. The flight plans for nadir and oblique imaging with the Matrice 300 and Mavic Pro sUAS used camera stations designed to achieve the 0.5 cm GSD by setting the appropriate altitudes and distances from the features to be imaged. Terrain following by the sUAS was not a consistently available option due to coarse resolution elevation models in the flight-planning software, so where terrain varies in elevation the GSD deviates by +/- 0.2 cm. Flight plans for oblique (i.e., off-nadir) imaging were based on a set gimbal angle of 45-degrees.

Mavic Pro flight planning was performed with the Litchi application and the Matrice 300 with the UgCS application. The specific sensor information for both platforms is shown in

Table II.1. The Zenmuse H20 incorporates a zoom lens; the focal length was fixed to the same parameter for all flights.

Table II.1. sUAS sensor specifications by platform

	Sensor Make/Model	Image Dimensions	Sensor Dimensions	Focal Length
Mavic Pro	DJI FC220	4000 x 3000	6.16 mm x 4.55 mm	4.74 mm
Matrice 300	Zenmuse H20	5184 x 3888	9.50 mm x 5.70 mm	25.4 mm

The RSI acquisition method requires repeating the view geometries by triggering the camera at the same locations and angles, with the same sensor (Stow et al. 2016). RTK GNSS plays a vital role in this process, as the manufacturer's stated positional accuracy of the DJI and Emlid RTK systems is 1.0 cm; the stated standard GNSS accuracy (without RTK) of the Mavic Pro is 3.0 m, and 1.0 m for the Matrice 300. Achieving < 1.0 m navigational accuracy, and therefore camera station repeatability, with the Matrice 300 requires use of the onboard RTK capabilities and an Emlid Reach RS2 base station mounted on a fixed location for all flights. Achieving reasonable RSI collections (< 0.5 m camera station repeatability) with the Mavic Pro is not possible due to its GNSS limitations. As each platform was programmed to fly at 1 $m \cdot s^{-1}$, waypoints were configured for a 2-second delay prior to image capture to improve platform stability; it is uncertain whether this delay is necessary, but was set to better simulate a 'stop-and-go' image capture rather than a fixed-wing constant-motion platform image capture and reduce the likelihood of the platform overshooting a camera station. The baseline flights and all RSI-based flights were repeated with this setup.

The non-RSI method is the default behavior of the Mavic Pro and Matrice 300. The Mavic Pro does not have RTK capability and for the Matrice 300 the RTK functionality was disabled. As previously stated, GNSS positional accuracy without RTK for the Mavic Pro is 3.0 m and is 1.0 m for the Matrice 300. The 2-second delay was removed from the Matrice 300 waypoints for the non-RSI flights, increasing the likelihood of over-shooting camera stations. These changes are useful in limiting the navigational accuracy and repeatability of waypoints for the platforms and their GNSS positions are more susceptible to changes in satellites' ephemeris. The baseline images for each feature are based on and identical to the RSI method, and repeated flights use these non-RSI parameters.

For each feature in the study area, the number and locations of waypoints were based on the sizes of the features and the 0.5 cm GSD requirement. With the Matrice 300, a baseline flight was conducted for each feature, and repeated nine times with the RSI method and two times with the non-RSI method. With the Mavic Pro, a baseline flight was flown for each feature and repeated nine times with the non-RSI method. Each repetition was flown sequentially with up to 3 hours passing between the baseline and final repeat being flown. The number of baseline waypoints and images collected for each feature by platform is shown in Table II.2.

Table II.2. Number of baseline waypoints (WP) and images (IM) by feature and platform

	Bridge 1		Bridge 2		Building 1		Building 2		Road 1		Road 2		Road 3	
	WP	IM	WP	IM	WP	IM	WP	IM	WP	IM	WP	IM	WP	IM
Mavic Pro	14	128	11	71	11	66	15	49	28	354	33	331	n/a	n/a
Matrice 300	8	103	8	101	11	110	10	132	n/a	n/a	12	176	12	145

2. IMPP Software Development

To accurately and automatically co-register and georeference sUAS image pairs in bulk and perform CNN-based damage detection it was necessary for me to develop a custom software solution. IMPP is a Python 3 application with a graphical user interface. It utilizes several open-source image processing and geographic information systems modules: OpenCV, GDAL, NumPy and SciPy.

The graphic interface of IMPP divides the application into two tabs. The Georeferencing tab is used to automatically co-register and georeference image pairs in bulk. The AI Inference tab allows for the selection of a trained neural network model and the automatic classification of damage in images or image pairs. Designing and training neural network models is performed by modifying python scripts included in IMPP but is not part of the graphic interface.

When using the georeferencing functionality, baseline imagery and repeated imagery's geographic coordinates and sensor information were loaded and used to automatically find bi-temporal image pairs and their georeferenced spatial extents based on the user-specified EPSG geographic coordinate projection. This information was saved in a project GeoPackage as point layers.

Oriented FAST and Rotated BRIEF (ORB) is an open-source feature detector in OpenCV that is used to detect and assign identifiers to unique features in each image within an image pair. ORB operates on grayscale images only (Karami, Prasad, and Shehata 2017). Prior to using ORB with the grayscale versions of the images, a contrast limited area histogram equalization (CLAHE) is applied to improve the contrast in areas of an image that are over-

or under-saturated (Ganesh and Ramesh 2017). ORB is used with the CLAHE enhanced images. With the features identified, IMPP then searches for matched features by counting the number of positions in the feature identifiers that differ using Hamming Distance (Norouzi, Fleet, and Salakhutdinov 2012). IMPP requires a minimum of 20 matched features to continue the image co-registration process.

Matched features from the two images were passed to a planar-homography algorithm that finds the 3x3 transformation matrix between the images with 8 degrees of freedom. This transformation matrix was used with a perspective transformation to align the time-*n* image to the baseline image and apply padding to the aligned image.

The local image coordinates for the baseline and time-*n* images were then converted to the specified geographic coordinate system and projection using GDAL with bi-linear interpolation of the pixel values for the full-color images. At this point the images were exported to a folder as aligned and georeferenced files.

In addition to the co-registered image outputs, IMPP was designed to output useful processing logs. Co-registration accuracy is automatically assessed by performing the ORB feature detection and Hamming Distance matching on the already co-registered image pairs and the results for each image pair are output as comma-separated files. The processing times for image loading, alignment, georeferencing and exporting were saved to the logs. Additionally, the percent of time-*n* image pixel overlap with the baseline image was calculated and saved to the logs.

3. Damage Simulation

Due to limitations in data availability for actual post-hazard damage features, manifestations of damage (cracks) were artificially added or removed from imagery using photography editing software, to generate bi-temporal image pairs containing no damage then damage for portions of structures. Large-scale geometric changes were not simulated in this study, in part because 1) the assumption is that small-scale damage would be a more challenging use-case for machine-learning than large scale, and 2) the ability to realistically simulate large-scale damage of infrastructure in imagery is challenging and 3) no literature was located that does this. The structural features were chosen to resemble critical infrastructure. The imaged structural features either already contained damage (cracks) or were undamaged and a method for simulating an undamaged and a damaged state was employed. When a feature in the baseline imagery already contained cracks, (i.e., a road, bridge, or building that has surface cracks), the Adobe Photoshop healing tool was used to remove some but not all cracks. When a feature in the subsequent (non-baseline) imagery did not contain cracks, Photoshop's clone, rotation, and skew tools were used to transfer cracks from other damaged features with similar background surface types to the undamaged features.

This approach to damage simulation yielded bi-temporal images containing radiometrically, spatially and contextually realistic changes, where the pre-disaster imagery portrays features in a pre-disaster state and post-disaster imagery with new damage. Figure II.1 shows an example of an original baseline image containing existing cracks, the result

from removing the damage and a subsequent-time image where cracks were artificially added.

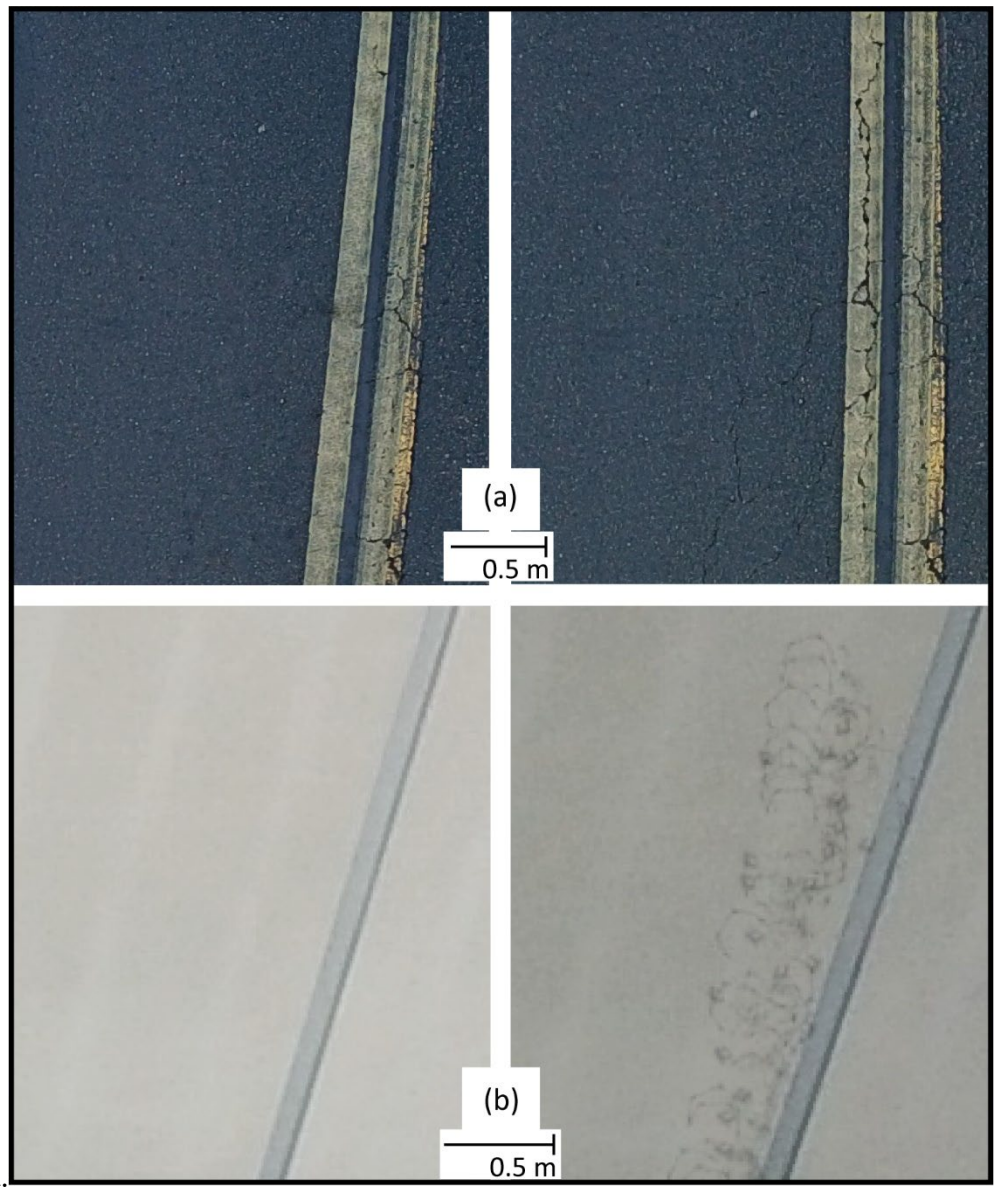

Figure II.1. Examples of damage simulation in baseline (left images, (a) and (b)) and time-*n* (right images, (a) and (b)) pairs. Removal of cracks in baseline images is shown (top) and addition of cracks to time-*n* images is shown (bottom).

4. Image Pre-processing

The image pre-processing steps accomplished three goals following damage simulation: 1) co-register and georeferenced bi-temporal image pairs (described above), 2) prepare

imagery for use in neural network training and 3) prepare imagery for use in neural network-based damage detection. For each of the study area's features, two projects were created in IMPP to co-register and georeference the Mavic Pro and Matrice 300 images, following procedures detailed in the IMPP Software Development section above. The first step performed for the entire image dataset was the damage simulation. Next, for each of the study area's features, two projects were created in IMPP to co-register and georeference the Mavic Pro and Matrice 300 images.

The georeferenced image pairs (training data) were then loaded into a GIS application, QGIS, and image pixels were labeled using a binary mask of 0 (no damage) or 1 (damage). All images for all features and both drone platforms were then split into two datasets, a training dataset and a classification dataset. The training dataset contained 70% of the images and labels, and the classification dataset contained the other 30% of images and labels. An example of a baseline image (a), post-hazard image with simulated damage (b), bitemporal layer-stack image (c), and damage labels where the damage extents are shown in red (d) for a pedestrian bridge, taken with the Matrice 300 are shown in Figure II.2.

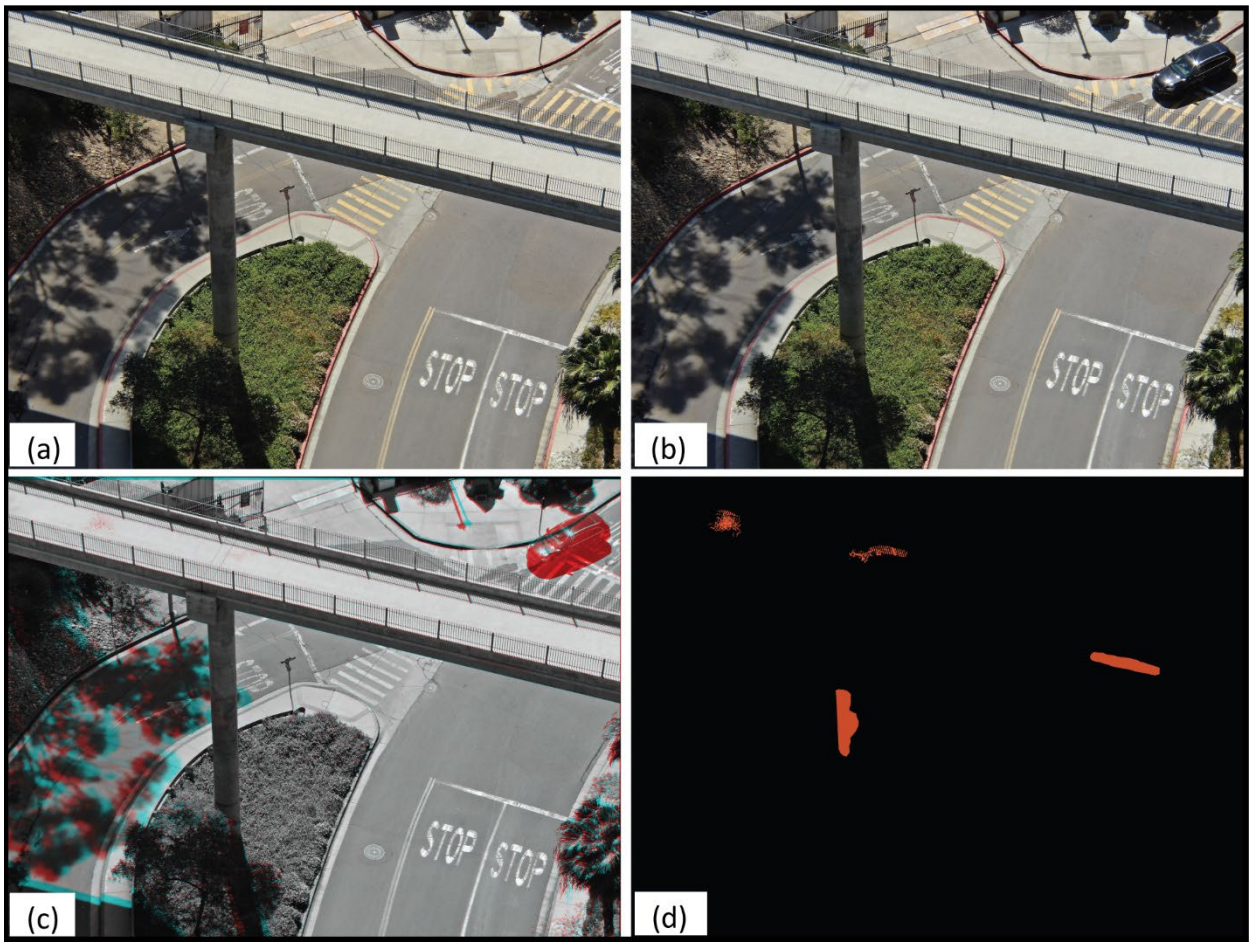

Figure II.2. Example of pre-processing inputs ((a) natural color image and (b) natural color image with damage) and outputs (layer-stack (c) where non-gray tones represent scene changes between images, and labels (d)) where the extents of the damage are shown in red.

5. Neural Network Training and Classification

The CNN model used for training is based on Mask R-CNN, which is suitable for semantic segmentation and object detection (He et al. 2017). Pretraining of the model was previously performed by its developers on a large image dataset of more than 1.5 million images with 100 classes (Chen et al. 2018). The process of re-training a model on new data and new classes is called fine-tuning and can be performed with hundreds rather than millions of images (Long et al. 2015). The training and fine-tuning process requires that

images be loaded as smaller 1024 x 1024 pixel image regions due to computer memory limitations. The Mask R-CNN model training performs best with a large batch size (16 images per batch). A single image from the Matrice 300 yields 20 images for training, and a single Mavic Pro image can yield 6 images.

Images were provided to the model as three-band bi-temporal layer stacks with the baseline grayscale image in band 1, the grayscale time-*n* image in band 2, and a simple image-difference product in band 3. As the images were split into smaller regions for training, the split regions underwent randomized flipping and rotation to increase the overall number of images available for training. Training was performed for 300 epochs where an epoch represents a number of training iterations equal to the number of images divided by the batch size.

Three versions of the model were trained. The first used the non-RSI image pairs, the second the Matrice 300 non-RSI image pairs, and the third the Matrice 300 RSI image pairs. Once the training was completed, the Mask R-CNN algorithm was used to classify the remaining 30% of bi-temporal image pairs and detect damage to the features. The resultant classification products are binary-mask raster in GeoTiff format showing new post-disaster damage and no damage.

D. Analytical Procedures

The analysis focuses on assessing the navigational accuracy and available precision of two sUAS for conducting RSI image acquisitions, the effects of RSI and non-RSI acquisition methods on co-registration accuracy at low altitudes (<122 m), and the accuracy of a CNN trained on RSI and non-RSI bitemporal change (i.e., simulated damage) objects.

1. RSI and non-RSI Navigational Accuracy

The RSI and non-RSI navigational accuracies of the platforms were assessed with two metrics. The first is a measurement of distance from the designated camera stations and where the GNSS RTK onboard the platforms indicate the images were actually acquired. Mean absolute error (MAE) and root-mean-square error (RMSE) metrics were generated from the offset values for all flights, per platform. This approach provides navigational and repeatability assessments based on the recorded GNSS RTK positions. Another metric for assessing the repeatability of image stations is the percent of overlap between baseline and repeated images. This metric uses a summation of the percent of non-overlapping pixels for each image pair and divides the percent-sum by the number of images, providing a mean percent error (MPE) measurement.

2. Image Co-registration Accuracy

The co-registration accuracy of images was calculated for RSI and non-RSI pairs after co-registration was performed in IMPP. Automatically generated and matched points within previously co-registered image pairs were used to determine the distances between matched pixels, providing the co-registration error. The automatically generated points were selected from a subset of those used in the co-registration process and are independent of that process (i.e., they were not used to perform co-registration). RMSE and MAE metrics of accuracy were then used to assess the co-registration accuracy.

3. Neural Network Classification Accuracy

CNN classification accuracy was quantified by based on known damage features on image pairs that were not used for training the model. As with the pairs used in training, these images have ground reference labels, where pixels representing damage have a class

value pre-assigned and non-damaged areas have a background class assigned. Intersection over union (IoU), a metric for calculating the percent of an object that falls inside and outside the area predicted by the CNN is used to score individual detections of damage. The mean IoU was assessed for the outputs from both RSI and non-RSI based CNN models.

E. Results

The results of the navigational and co-registration accuracy assessments clearly demonstrate that 1) RSI resulted in greater repeatability of camera station waypoints, 2) RSI resulted in greater overlap of image pairs, and 3) RSI resulted in higher image co-registration accuracy. The Mask R-CNN neural network classifier yielded highest accuracies when applied to the bi-temporal images based on the RSI acquisition approach.

1. Navigation and Image Co-registration Accuracy

The navigational accuracies of the M1 and M300 sUAS are dependent on the ability of their GNSS to accurately determine their geographic positions. The M1 does not utilize RTK for accurate positioning while the M300 can be operated with or without RTK positioning.

Table II.3 shows the navigation, overlap, and co-registration results, separated by whether images were of nadir or oblique angles and by sUAS platform. The results are given in both mean absolute error (MAE) and root-mean-square-error (RMSE) with emphasis on MAE due to its lower sensitivity to outliers; RMSE follows the MAE trends in this study. The analysis shows that the M1's navigational accuracy was highly variable, from < 20 cm to > 9 m. The M300 with RTK disabled had a navigational accuracy ranging from 13 cm to 30 cm and with RTK enabled the range was 14 cm to 18 cm.

The image-pair-overlap assessment also shows that RSI yielded greater overlap, which is dependent on platform navigation and sensor pointing accuracy. Image overlap was

consistently lower for the M300 with RTK enabled (8.6 – 10.1 %) versus disabled (12.5 – 18.0 %), and in most circumstances was much lower than the M1 (7.6 – 27.8 %).

Image co-registration results show a similar pattern, with the RSI method yielding image pairs with the least amount of error. Images from the M300 with RTK-enabled have an MAE of 2.1 (nadir images) to 2.3 (oblique images) pixels. With RTK-disabled, the M300 images have MAE of 3.8 (nadir) and 5.0 (oblique) pixels. Images from the M1 yielded the highest MAEs (nadir: 11.0 pixels, oblique: 139.2 pixels).

Table II.3. Navigational and image co-registration accuracies of the M300 and M1 with RSI (RTK-enabled) and non-RSI (RTK-disabled), presented for nadir (roads) and oblique (buildings and bridges) image collections. n = number of image pair samples used for accuracy assessment.

Dataset	Navigation MAE (m)	Navigation RMSE (m)	Image Overlap (MPE)	Co-registration MAE (pixels)	Co-registration RMSE (pixels)
Nadir, M300 RSI (n = 161)	0.174	0.191	10.1	2.1	4.9
Nadir, M300 non-RSI (n = 160)	0.277	0.292	12.5	3.8	8.7
Nadir, M1 non-RSI (n = 685)	9.131	9.770	27.8	11.0	53.3
Oblique, M300 RSI (n = 223)	0.144	0.173	8.6	2.3	5.5
Oblique, M300 non-RSI (n = 223)	0.137	0.150	18.0	5.0	10.7
Oblique, M1 non-RSI (n = 314)	0.184	0.508	7.6	139.2	195.9

2. Neural Network Classification Results

Results from the Mask R-CNN neural network classification show clear variations between acquisition methods (RSI vs. non-RSI), view perspectives (nadir vs. oblique) and feature types (roads, buildings, bridges). Differences in accuracy are also observed by platform type in the case of non-RSI acquisitions.

Table II.4 contains the neural network classification accuracy results presented as mIoU (%). The CNN classifier trained and tested on RSI-acquired imagery outperformed the non-RSI classifiers by an average of 20.5% mIoU, with an overall accuracy of 83.7% mIoU.

Table II.4. Mask R-CNN classification accuracy in mIoU percent, by sUAS platform type, acquisition method, feature type, and view perspective.

	Roads (nadir)	Buildings (oblique)	Bridges (oblique)	Overall Accuracy
M1 (non-RSI)	88.9%	71.4%	37.5%	72.5%
M300 (non-RSI)	64.3%	64.3%	40.9%	54.0%
M300 (RSI)	88.2%	92.3%	69.2%	83.7%

A plot showing the relationship between the nadir and oblique RSI and non-RSI Mask R-CNN mIoU results against their associated image co-registration MAE values is shown in Figure II.3. From this, an inverse linear trend is evident, such that the Mask R-CNN detections were more accurate when the co-registration error was lower.

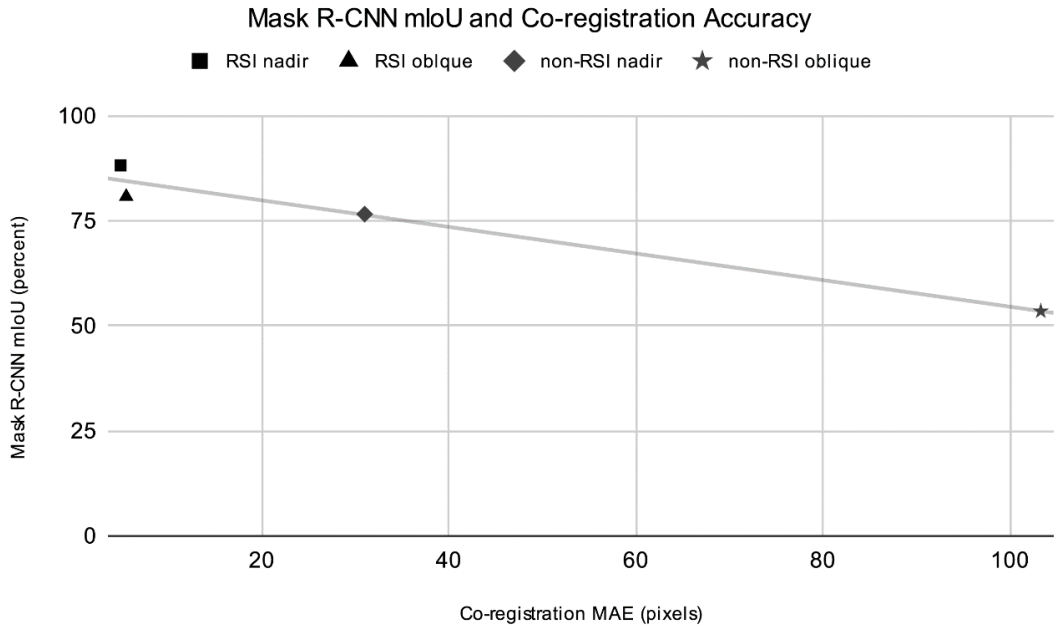

Figure II.3. Relationship between the Mask R-CNN (mIoU) and co-registration (MAE) accuracies by RSI, non-RSI acquisition methods and nadir, oblique view perspectives.

F. Discussion

The findings and related discussion are structured by addressing the three research questions and provide a synthesis of the results. Challenges of the study and recommendations for follow-on research are also discussed.

How accurately can an RTK GNSS repeatedly navigate a sUAS platform and trigger a camera at a specified waypoint (i.e., imaging station)?

The basic principle of RSI is that collecting images from the same locations and view perspectives, with the same sensors over time should result in improved image-to-image co-registration (Stow et al. 2016). In practice, no prior studies provide evidence that sUAS are currently able to accomplish this with a precision that fully negates the effects of parallax. However, this research has demonstrated that with the aid of RTK GNSS for navigation, a state-of-the-art sUAS, the DJI Matrice 300, is able to repeat waypoints and capture images to within 14 cm (MAE) horizontally and vertically (Table 3).

The accuracy of image co-registration and CNN-classification of damage are in-part dependent on the ability of a sUAS platform to navigate to camera stations accurately. RTK-based navigation of sUAS poses some challenges: 1) setting up a local RTK base receiver or remotely accessing an RTK base is required, 2) remote access of an RTK base receiver requires internet data availability, 3) environmental conditions like wind and radio interference can lead to less-accurate navigation, and 4) proprietary sUAS autopilot systems like those from DJI may handle horizontal and vertical navigation by RTK differently. Technical advancements in the delivery of RTK corrections to GNSS receivers and the

implementation of RTK on sUAS are expected to increase overall navigation accuracy and convenience.

How does the co-registration accuracy vary for RSI versus non-RSI acquisitions of sUAS imagery captured with nadir and oblique views?

With the use of automated co-registration software (IMPP), the navigation precision of the DJI Matrice 300 combined with RSI is sufficient to co-register bi-temporal images to within 2 pixels (MAE) (Table 3). By contrast, the same co-registration approach results in much higher error for images collected further apart spatially with either the Matrice 300 or DJI Mavic 1.

While the navigation capabilities of sUAS platforms and available GNSS solutions such as RTK may continue to improve, other factors can affect automated image co-registration and could be explored further. Detection and elimination of image match points along the boundaries of transient shadows (i.e., shadows which shift position in bi-temporal image pairs) could result in lower co-registration errors (Storey et al. 2017). Additionally, accurate image alignment and generation of photogrammetric points clouds relies on the identification and correction of lens distortions (Turner et al. 2012). It is reasonable to conclude that uncorrected distortions in the imagery may reduce the overall co-registration accuracy.

What difference in classification accuracy of bi-temporal change objects is observed with a CNN for RSI and non-RSI co-registered images?

A method for data preparation based upon the object-based image change analysis (OBICA) approach was implemented in this study; bi-temporal images of a scene are co-registered and change objects are detected (Dronova, Gong, and Wang 2011; Stow 2009). The process for acquiring the images (RSI vs. non-RSI) was shown to directly affect the co-

registration accuracy of bi-temporal image pairs. Despite the high accuracy of co-registration that was quantified, evidence of image misregistration is visually apparent in the bi-temporal images, and higher error in co-registration was expected to lead to higher error in change detection.

Results from the CNN analysis showed that the RSI-based images, with the lower co-registration errors, had an overall higher change detection accuracy than the non-RSI images with higher co-registration errors. Of particular interest in these results are the differences between the accuracy of CNN detections for nadir versus oblique views. The nadir views, focused on roads, contained less apparent parallax and had the lowest co-registration errors for both RSI and non-RSI images. Nadir captured images yielded higher change detection accuracies, with the Matrice 300 RSI and Mavic 1 non-RSI performing similarly (88.2% versus 88.9%) (Table II.4). The damage in the oblique views of buildings, with the buildings filling the majority of the image frames, was classified with 92.3% accuracy for the Matrice 300 RSI and 71.4% for the Mavic 1 non-RSI (Table II.4). The classification accuracies for the oblique views of bridges, that filled less of the image frames and contained the most parallax (and associated higher co-registration error), were much lower at 69.2% (Matrice 300 RSI) and 37.5% (Mavic 1 non-RSI) (Table II.4). The classification accuracy of Matrice 300 non-RSI images was lower than the Matrice 300 RSI for all features and view perspectives, and only slightly higher than the Mavic 1 non-RSI for the bridges (40.9% versus 37.5%) (Table II.4).

Beyond the impact of co-registration accuracy on the Mask R-CNN classification accuracy, other factors are worthy of further research to improve damage detection performance. The most obvious of these is additional images with simulated damage, as the

accuracy of any CNN is likely to improve with additional data (Dieleman, de Fauw, and Kavukcuoglu 2016; He et al. 2017; Voulodimos et al. 2018). The Mask R-CNN model is also only one example of numerous available algorithms, and other models could be explored to determine whether they are more or less robust to co-registration errors.

G. Conclusions

The results from this study indicate that achieving higher co-registration accuracies via RSI results in higher Mask R-CNN change detection accuracy and that sUAS combined with RSI and automated damage detection is useful in detecting fine-scale damage, such as cracks, to critical infrastructure. The application of Mask R-CNN to co-registered bitemporal image pairs for change detection is novel. Additional methods for improving the co-registration accuracy of images (i.e., improving platform navigation precision, transient shadow masking, lens distortion corrections) should lead to more accurate and reliable damage-detection results.

For emergency response identification of damage such as cracks to critical infrastructure, the outlined approach in this paper demonstrates the utility of RSI and Mask R-CNN -based damage detection. Both RSI data acquisition and analysis with Mask R-CNN have specific pre- and post-hazard-event requirements. RSI requires that both pre-event and post-event imagery be collected using GNSS RTK for sUAS navigation and a point-able camera. This allows for the greatest accuracy in image pair co-registration. The Mask R-CNN must also be pre-trained on co-registered image pairs containing undamaged and damaged features with damage labels.

Post-hazard damage assessment to critical infrastructure with sUAS, RSI and machine learning has the potential for benefits in the rescue and relief phases (image acquisition and

damage detection accuracy and timeliness) of the DMC but requires some considerations to achieve well. For RSI-based bitemporal damage assessments to be effective, pre-damage imagery of the infrastructure must exist and have been collected with the same sensor at the same view angles. Accomplishing this relies on the use of a real-time-kinematic GNSS for sUAS navigation and camera triggering. The Mask R-CNN algorithm used in this research requires training on bitemporal examples of no-damage (baseline) and damage (time-n). This research focused on fine-scale cracking of concrete and asphalt as representations of damage to critical infrastructure with the belief that coarser-scale damage would likely be a lesser challenge.

III. Comparison of CNN- and RNN-based damage detection methods for critical infrastructure

A. Background

In this Chapter, I develop and test a novel approach to exploiting recurrent neural networks (RNN) for detecting post-hazard damage (cracks) to critical infrastructure and compare the results to those from a similarly structured convolutional neural network (CNN) without recurrent nodes. In Chapter 2, the repeat station imaging (RSI) datasets collected with the DJI M300 using real-time-kinematic (RSI) navigation had the lowest mean absolute error (MAE) for co-registration of time-n image pairs (2.3 pixels MAE) versus the DJI M300 without RTK (5.0 pixels MAE) and the DJI Mavic 1 without RTK (139.2 pixels MAE). The results from Chapter 2 further show the positive relationship between co-registration accuracy and the Mask R-CNN performance for detecting change objects (damage). In this Chapter I compare the performance of RNN and CNN crack damage detectors using the DJI M300 RTK dataset for training, validation and evaluation.

This section covers foundational topics on 1) critical infrastructure and damage, 2) basic structure of a CNN for image classification tasks, and 3) basic structure of an RNN for image classification tasks. More thorough descriptions of CNN and RNN principles and designs can be found in the cited literature for each section.

1. Critical Infrastructure and Damage

Substantial debate persists within academic research, federal management agencies, and the organizations responsible for managing infrastructure, regarding a common definition of "critical infrastructure" (Cutter 2003; Lippitt, Stow, and Coulter 2015; Pescaroli and

Alexander 2016). Although no commonly agreed upon definition of what constitutes specific critical infrastructure features can be found across all domains, the category of critical infrastructure can be summarized as any resource or feature that is necessary for the functioning of society (Cvetkovi and Akade 2013). Built features are generally critical infrastructure when they are any physical component of a vital societal resource that would be hampered or cease functioning without the feature, particularly immediately following a major hazard event (Cutter 2003; Lippitt et al. 2015). The societal resources making use of critical infrastructure are emergency response, transportation, energy, and governmental (Lippitt et al. 2015). Examples of built features found in these societal resources are: government and medical buildings, bridges, roads, railways, airports, and the physical components of an energy grid.

Damage from earthquakes is the primary application context of this component, which emphasizes machine learning for detecting damage to critical infrastructure based on sUAS-RSI data sets. Manifestations of damage of critical infrastructure that could result following earthquakes are taken from a review of several damage indices and observations of damage from remote sensing systems (Blong 2003; Cutter 2003; Gerke and Kerle 2011; Joyce, Belliss, et al. 2009). These built feature and damage associations are:

1. Buildings - subsidence, tilt, general structural damage, and collapse, resulting from cracking of walls and foundations and the liquefaction of soil;
2. Electrical grids – failures resulting from damaged buildings, tilting or toppled utility poles and towers;
3. Roads – severe cracking from shifting, faulting, and/or liquefied soil, debris from collapsed buildings and landslides; and

4. Bridges – surface cracking, tilting, or collapse from movement of support structures during the shaking and/or liquefaction of soil around pylons.

Buildings, bridges and roads are the focus of this chapter, as these features are less complex than electrical grid components (for sUAS navigation) and less fragile (in the event of an sUAS platform failure). Cracks were chosen as the focus for damage simulation and detection as 1) the three chosen feature types share this damage manifestation, 2) cracking is less complex to simulate in imagery than other manifestations (i.e., tilting, collapse, and shifting), and 3) as small-scale features in very high-resolution imagery (1 cm ground sample distance), cracks should be more challenging to detect in multi-temporal image pairs as change objects than large-scale damage types.

2. Basic CNN Architectures

As this research evaluates the accuracy of a customized CNN for change object detection (Section C, 2.), background information on the functional components that comprise a basic CNN (single-date images for feature identification) are provided here.

CNN models are automated algorithms suitable for pixel- and object-based classification (He et al. 2017; Long et al. 2015). Supervised models, those where images used for training contain labels, are capable of achieving highly accurate results in object detection and identification (Cheng and Han 2016; Lyndon et al. 2015). A basic CNN consists of a series of connected layers, with each layer containing connected nodes. Typical node types are 1) inputs, matching the dataset structure with each input node connected to a matching convolution node in the next layer, 2) convolution, used in feature extraction from the input

nodes, 3) pooling, used for dimension reduction of features extracted through convolution, and 4) fully-connected, used for identifying feature relationships and classification decisions (Voulodimos et al. 2018). Figure III.1 shows a simple CNN structure with each of these node types. Each convolution, fully-connected, and classifier node contains an 'activation function' (an algorithm for determining what values the node should output based on the application of the weights to the input values, described in further detail later).

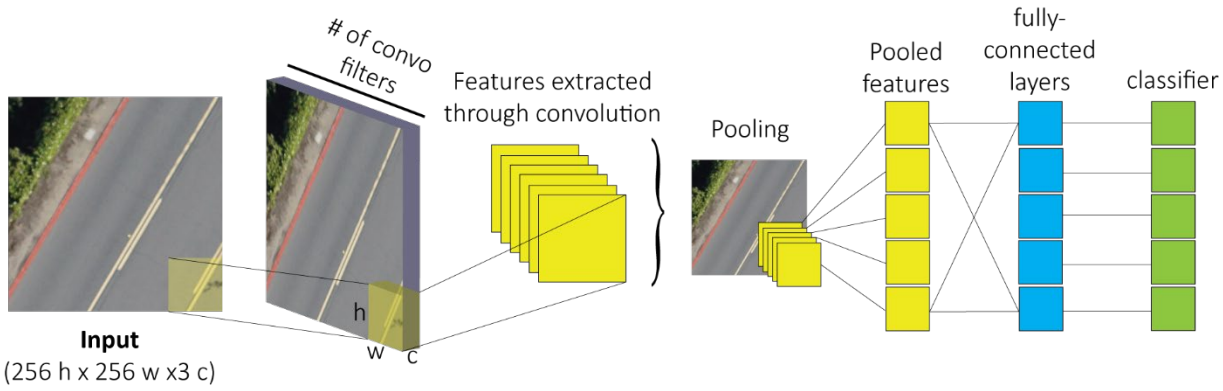

Figure III.1. Example of a simple CNN using a small natural color image as inputs. Convolution filters learn to identify and extract features and are passed to the pooling layer for dimension reduction. The fully-connected layers separate and pass the features to the classifiers.

The first layer for a CNN, the input layer, takes a training dataset with a matrix shape compatible with the input layer. This CNN input layer is typically a placeholder, and compatibility with the training dataset means it contains a number of static (non-learning) nodes equal to the number of images (or batches, see below), and node shapes equal to the shapes of the images (or image 'tiles', see below). This input layer passes the training dataset onto the first convolution layer. The training dataset, so long as it is compatible with the input layer, can contain images of any size (rows (m) x columns (n)) and dimensions (color channels (c)). A typical training approach for a 'deep' network (one with many connected nodes and millions of trainable parameters) is to feed groups of images (b batches) into the

input layer as an array of shape (*b* x *m* x *n* x *c*). This process of 'batching' combined with the depth of the network and available hardware processing resources (graphic processing units, central processing units and random-access memory) typically necessitates that training dataset images be either resized or split into smaller image 'tiles' for training. For example, a single image from the DJI M300 using the Zenmuse H20 camera has dimensions of 3888 (*m*) x 5184 (*n*) x 3 (*c*). Splitting this image into smaller tiles of 256 x 256 pixels creates 336 images of shape 256 (*m*) x 256 (*n*) x 3 (*c*). A batch of 20 image tiles, then, would be an array with a shape of 20 (*b*) x 256 (*m*) x 256 (*n*) x 3 (*c*), and the input layer of the CNN would need to match this shape (i.e., 20 nodes, of size 256 (*m*) x 256 (*n*) x 3 (*c*)).

Convolution layers in the feature extraction process are designed to learn the spatial and spectral representations of the image classes from the input layer's nodes. These layers are not 'fully-connected', meaning the convolution nodes in these layers do not share inputs and outputs with other nodes in the same layer. Each node in the initial layer of convolution kernels will accept an input image shape equal to (1 x *m* x *n* x *c*). Subsequent layers of convolution nodes and fully-connected layers are designed to accept input shapes equal to the output shapes of their immediate predecessor. Figure 2 shows the structures of convolution kernels used in CNNs. For demonstration purposes, the convolution kernel (a) is assigned to a single node in a layer and the kernel has a shape of 3 (*m*) x 3 (*n*). The values in the convolution kernels are among the weights the model learns, though during the initial training step these weights are typically either randomized or set to single values (i.e., 0.5). As the kernel, (a) moves across the image (b) and (c) in Figure III.2, it computes results that are added to the output array for that node (the output arrays for all nodes in a layer become the input arrays to the nodes in the next layer).

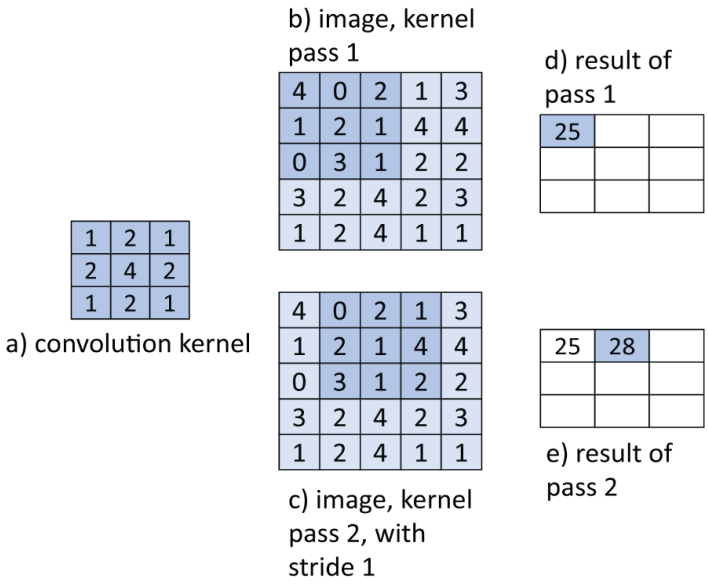

Figure III.2. Example of how a convolution kernel works. In a CNN, the values in a) are the weights learned by the model. Both b) and c) demonstrate how the kernel moves along an image. Both d) and e) show the new pixel values of the output image as the kernel moves, and the values are the result of some activation function (i.e., sigmoidal, hyperbolic-tangent, rectified linear unit, gaussian, etc.).

Activation functions are the arithmetic operations that generate the output values of each convolution node, fully-connected node, and classifier in a CNN. These activation functions evaluate the values created by a node's application of its weights to its input values, and outputs a new value based on the activation function. Three common activation functions are rectified linear unit (ReLU), hyperbolic tangent, and sigmoid. The ReLU activation evaluates values generated by the input data and weights in a node, and outputs the generated values if they are greater-than zero; the node's output values are zeros if the generated values are less-than or equal-to zero (Nwankpa et al. 2018). This function is useful for binary and non-binary classification problems as its output range can match the input data and it does not approach a limit of one or zero (the vanishing gradient problem)(Tan and Lim 2019). The hyperbolic tangent function evaluates the input data and weights and outputs values of -1 to 1, while the sigmoid function outputs values from 0 to 1 (Nwankpa et al. 2018). Both the

hyperbolic tangent and sigmoid functions are useful in binary classification problems and RNNs make use of hyperbolic tangent (Hochreiter and Schmidhuber 1997).

A layer with pooling nodes takes the spatial/spectral features found (i.e., examples of classes) during convolution and decreases their height and width dimensions. This is achieved by replacing the pixel values in a moving matrix (kernel) of some height and width with the maximum pixel value. The number of columns and rows the kernel moves before its next operation is referred to as the stride; for image reduction the stride is equal to the kernel size. For example, a 20-pixel by 20-pixel image could be reduced with a maximum value kernel of 2 by 2 pixels, and a stride of 2, to a new image of size 10-pixels by 10-pixels. Equation III.1 shows how the reduction in image size is calculated with a pooling node of a given size. Pooling layers are used to reduce the number of parameters needed by a model to represent a feature.

$$h_o = \frac{(h - K_h)}{s_h} + 1, \qquad w_o = \frac{(w - K_w)}{s_w} + 1$$

where $h = \text{image height}$, $w = \text{image width}$, $K_h = \text{kernel height}$, $K_w = \text{kernel width}$, $s_h = \text{row stride}$, $s_w = \text{column stride}$, $h_o = \text{output height}$, $w_o = \text{output width}$

(III.1)

Unlike convolution layers, fully-connected layers share their input and output data within a layer and with the following layer. The fully-connected layers take the output pixel values from convolution nodes as inputs, perform an activation function on the values, and compares their output values (predictions) with the image label. The degree of error between the classification prediction and the label is measured as “loss.” A CNN ‘learns’ using this

loss metric to move backwards through the layers and determine whether the model weights' values (fully-connected nodes' weights, convolution kernel values/weights) should be increased or decreased in a process called back-propagation. A common method of back-propagation, 'gradient descent' uses the derivatives of the loss function with each node's output value to determine whether that nodes weights should be increased or decreased.

3. RNN Architectures

Unlike CNNs, containing no specific temporal components, RNNs are intended to track temporal patterns through specifically designed nodes. The initial applications and traction of RNNs were in natural language and signal processing (Lawrence, Giles, and Fong 2000; Yu et al. 2019). Given a sequence of data inputs, an RNN learns patterns to infer the next sequence. The basic recurrent network cell consists of input values and labels, and an operation that finds the standard deviation of the weights and biases against those inputs over time (Yu et al. 2019).

Long-short-term-memory (LSTM) nodes address an issue with the basic RNN cell, where the cell's predictions become less accurate as time (the number of data sequences) increases. It accomplishes this by introducing a 'forget gate', 'input gate', and an 'output gate'. Input data enters the 'forget gate' first, and the cell determines whether to 'keep' or 'forget' the data (Hochreiter and Schmidhuber 1997; Yu et al. 2019). The 'input gate' component determines what data can be saved in the LSTM node, and the 'output gate' determines the node's data output (Hochreiter and Schmidhuber 1997; Yu et al. 2019). This process of forgetting and determining the data the node should store makes RNNs with LSTM more

robust than basic RNN nodes over time (Hochreiter and Schmidhuber 1997; Lawrence et al. 2000).

An RNN designed for change detection of multi-temporal imagery uses co-registered multi-temporal images that are first passed through convolution filters similar to how a CNN functions (Lyu et al. 2016). A sequence of time-1 and time-*n* images separately and simultaneously undergo convolutions to extract features. The RNN computes weights for the image sequence and saves them in LSTM nodes (Lyu et al. 2016; Mou, Bruzzone, and Zhu 2019). The ability of an RNN to learn the spectral-radiometric, spatial, and temporal properties of features makes them attractive options for image change detection.

The co-registration process and the images in RNN change and anomaly detection from the cited literature have been both simple and made use of best-case scenarios. Images in the literature were either collected from repeat-pass satellites (i.e., Landsat) or fixed-mounted video cameras, greatly limiting and/or eliminating image misregistration (Lyu et al. 2016; Mou et al. 2019). The RNN classification aims of Lyu et al. (2016) and Mou et al. (2019) pertain to large-scale land-cover and land-use changes. The structures of these RNNs, furthermore, treat the time-sequential images as individual dates, merging them into the RNN nodes only after the features were extracted from the individual date images. My research develops and tests an RNN using more challenging imagery (higher spatial-resolutions with greater misregistration relative to the size of a pixel) using a truly time-sequential structure to detect small-scale changes (cracks).

B. Objectives

This chapter describes the development of an RNN architecture and an assessment of damage detection accuracy of RNN-derived products and those from a comparable CNN.

The accuracy and processing-time of damage detection are assessed for each routine; the goals are to determine which routine yields the most accurate damage detection and highest speed of processing.

A novel aspect of this research is its application to multi-temporal images, such that co-registration effects on change detection are minimized. The following research questions are addressed:

1. How accurate are post-hazard damage detection products generated using RNN and CNN models?
2. In the context of rapid response for emergency management, how do the damage detection models compare in processing time from the instance that co-registration is completed to the point in time that all features of interest have been assessed?

C. Methods

To meet the objectives and address the questions outlined in this chapter, the dataset with the lowest co-registration MAE (2.3 pixels) from Chapter 2 (DJI M300 with RTK and RSI collection) was used as input to the CNN and RNN models. The dataset structure and damage simulation method were modified from Chapter 2 to fit the input requirements for the RNN learning process. RNN and CNN training and evaluation were performed, and the accuracy results were compared against the RNN results. The processing times for both models were measured during the classification tasks to derive performance benchmarks.

1. CNN and RNN Dataset Structures

The methods of learning for the Mask R-CNN (from Chapter 2), and the CNN and RNN in this chapter are very different. Each used identical imagery from the RSI co-registered

images, split into smaller image tiles. The Mask R-CNN required image labels containing the spatial extents of the manually-simulated damage. Target labeling for the CNN and RNN is simpler, as either an image tile contains damage and the class label for the tile is 1, or it does not and the label is 0. Unlike the Mask R-CNN, where fewer examples of damage were required to fully-train, the CNN and RNN requires significantly more examples. To achieve this, the CNN and RNN were trained using automatically generated cracks to represent damage, rather than manually simulated damage as in Chapter 2.

Pre-processing of the CNN images in this chapter involved combining co-registered time-1 and time-*n* images into single, two-channel, bi-temporal images. The RNN pre-processing required each camera station containing co-registered time-1 and time-*n* images to be treated as image sequences, similar to image frames in a viewing-stationary video.

The images for both the CNN and RNN were further processed prior to training by splitting them into tiled batches. The batch dimensions for the CNN tiles were 20 (*b*) x 256 (*m*) x 256 (*n*) x 2 (*c*) and for the RNN 20 (*b*) x 2 (*t*) x 256 (*m*) x 256 (*n*) x 1 (*c*). For the RNN, *t* represents the time-sequence length.

Cracks were automatically simulated with the time-*n* image tiles during the batching process. To automatically generate the cracks, a semi-randomized process was used. Half of the time-*n* tiles were randomly selected from the batch to receive simulated damage. The region location for placement of the simulated cracks was randomly determined using the 'numpy random' function (Harris et al. 2020). The region size for crack placement was randomly selected from a list ranging from 24 (*m*) x 24 (*n*) pixels to 128 (*m*) x 128 (*n*) pixels. Within the selected region location and size, a random number of cracks was placed, ranging from 1 to 10. Crack widths were randomly chosen from 1 to 2 pixels, and crack lengths were

randomly chosen from 4 pixels up to the region size. The starting locations of the cracks within a region were randomly chosen, and the brightness values of the cracks were randomly selected from lowest 20% of the available brightness range (0-50). Brightness values for cracks were allowed to vary randomly within the brightness range. A gaussian smoothing function was applied to 80% of the cracks generated. During this process, any image tile selected to receive cracks was given a training label of 1, and those selected to remain unaltered were labeled 0. Figure III.3 shows the comparison of the dataset structures between the CNN and RNN for a single camera station, including tiling, damage simulation, and labeling examples.

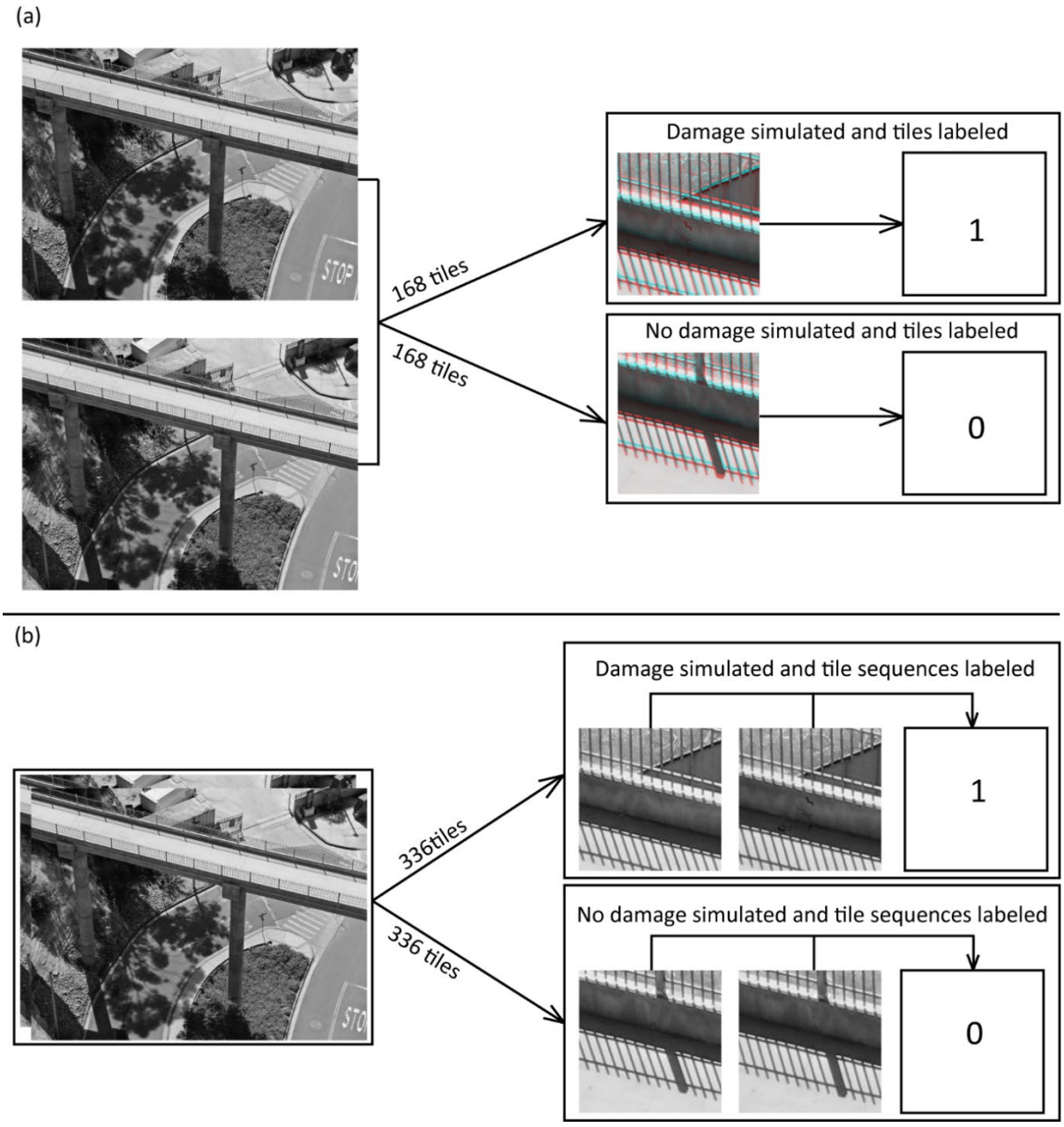

Figure III.3. Difference in dataset structures from (a) the CNN and (b) RNN. The CNN time-1 and time-*n* image pair is stacked into a 2-channel image and split into two sets of 168 image tiles of 256 (*m*) x 256 (*n*) x 2 (*c*); a set with simulated damage, and a set of without simulated damage. Each tile is labeled '1' if damage was simulated, or '0' if damage was not simulated. The RNN time-1 and time-*n* images are kept single-channel and tiled separately. Sequences of spatially corresponding time-1 and time-*n* tiles of size 2 (t) x 256 (m) x 256 (n) x 1 (c) are split into two sets of 336 tiles each; one set for

damage simulation and one without damage simulation. A single label is applied to each sequence of tiles; '1' for damage, and '0' for no damage.

2. CNN Architecture and Training

The CNN developed for this chapter consists of blocks of convolution layers, separated by pooling layers, that feeds into a network of fully-connected layers that are inputs to the output classifier. Layers consisting of convolution and fully-connected nodes use ReLU activations, and the output classifier uses sigmoid. The structure of the CNN is shown in Figure III.4 and differs from the Mask R-CNN tested in Chapter 2 by performing whole-image rather than region-based convolution.

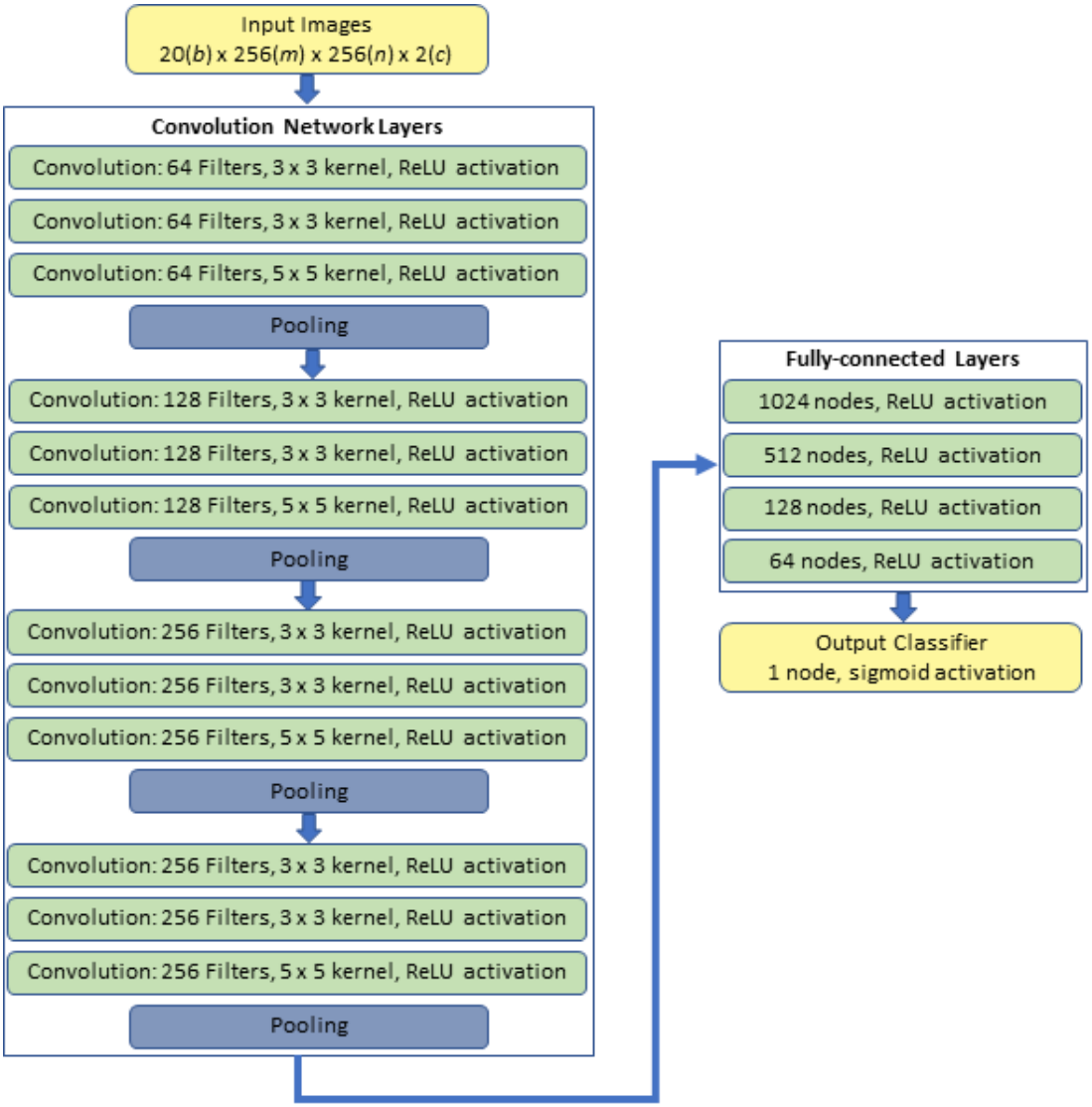

Figure III.4. Structure of the CNN, comprised of input images, blocks of convolution and pooling layers, fully-connected layers, and the output classifier.

CNNs were trained with the bitemporal, two-channel images and a batch size of 20 images. During training, a selection of 30% of the DJI M300 RTK RSI dataset image pairs was set aside and used exclusively for validation accuracy. The training was stopped at 60 epochs (every batch for all of the images was run 60 times).

3. RNN Architecture and Training

The convolutional network, fully-connected layers, and output layer used for the CNN were included in the design of the RNN. For the CNN, the convolution layers operated on single images with two channels (time-1, time-*n*), the implementation in the RNN operated on two images (time-1, time-*n*) with one channel each, together using a method in TensorFlow called "time-distributed convolution." This distinction is important, as the LSTM nodes in the RNN require time-sequential images as inputs. The RNN structure is shown in Figure III.5.

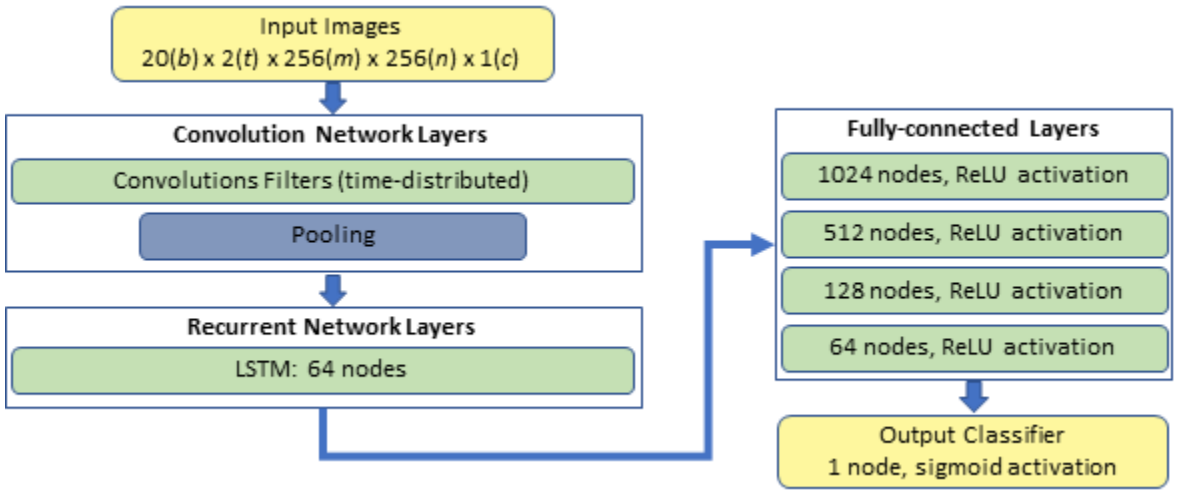

Figure III.5. Structure of the RNN, where the convolution layers are identical to the CNN in Figure 4, but applied to the time-sequential images. The RNN includes the LSTM recurrent network layers, and the fully-connected layers and output classifier are identical to the CNN in Figure 4.

RNN training was performed with time-1 and time-*n* image pairs and a batch size of 20 pairs (40 images). During training, a selection of 30% of the DJI M300 RTK RSI dataset image pairs was set aside and used exclusively for validation accuracy. The training was stopped at 60 epochs.

D. Analytical Procedures

Once again, the analysis focus is on CNN and RNN accuracy, and processing times. The CNN and RNN accuracies and processing times were automatically recorded as a function of custom software that I developed.

1. CNN and RNN Damage Detection Accuracy

The detection accuracy of the neural networks was evaluated by feeding the models selected images of bridges, buildings, and roads that were not used in model training and validation. Because the simulated damage generation was automated and the tiles in an image were predicted by the models to contain either new time-*n* cracks or not, a direct cross-tabulation of the predictions and labels was possible.

These cross-tabulated confusion matrices were generated for the categories of bridges (oblique views), buildings (oblique views), and roads (nadir views). Overall (total percent of correct predictions), users' (compliment to errors of commission) and producers' (compliment to errors of omission) accuracies characterize the metrics of analysis.

2. CNN and RNN Processing Times

Processing times for the loading, training, evaluation and exporting of the images and information products were recorded in hundredths of a second within the custom software using the python 'time' module. The results were normalized by seconds per megabyte (s / Mb) of data. The recorded times inherently reflect the hardware on which the models were run (see Appendix A for hardware specifications); both models had identical data volumes and were run on the same hardware.

E. Results

The results from the RNN and CNN accuracy analysis demonstrate that both models are capable of very accurately identifying regions in multitemporal image pairs containing a trained class (cracks). There is little difference in crack detection accuracy between the two types of deep learning detectors for images of bridges, buildings and roads and view perspectives (oblique and nadir).

1. CNN & RNN Accuracy

Table III.1 (CNN) and Table III.2 (RNN) show the cross-tabulation results, user's, producer's, and overall accuracy for post-event crack damage detection. The mean overall accuracy of the RNN (98.3%) and CNN (96.9%) indicates comparable performance at detecting post-event damage/cracks with the RNN accuracy 1.4% higher than the CNN. The user's and producer's accuracies for the 'damage' class were lower than for the 'no damage' class across all features with both the CNN and RNN, ranging from 93.0% to 97.6%.

Table III.1. CNN cross-tabulation confusion matrix for 2 classes: no damage, damage. Results are presented by primary feature types: bridges, buildings, roads.

	Bridges			Buildings			Roads		
	No Damage	Damage	User's Accuracy	No Damage	Damage	User's Accuracy	No Damage	Damage	User's Accuracy
No Damage (Reference)	4290	64	98.5 %	5176	79	98.5 %	4002	83	98.0 %
Damage (Reference)	119	902	98.3 %	126	1674	93.0 %	75	1215	94.2 %
Producer's Accuracy	97.3 %	93.4 %		97.6 %	95.5 %		98.2 %	93.6 %	
Overall Accuracy			96.6 %			97.1 %			97.1 %

Table III.2. RNN cross-tabulation confusion matrix for 2 classes: no damage, damage. Results are presented by primary feature types: bridges, buildings, roads.

	Bridges			Buildings			Roads		
	No Damage	Damage	User's Accuracy	No Damage	Damage	User's Accuracy	No Damage	Damage	User's Accuracy
No Damage (Reference)	4309	41	99.1 %	5154	81	98.5 %	4027	39	99.0 %
Damage (Reference)	47	979	95.4 %	63	1758	96.5 %	31	1279	97.6 %
Producer's Accuracy	98.9 %	96.0 %		98.8 %	95.6 %		99.2 %	97.0 %	
Overall Accuracy			98.4 %			98.0 %			98.7 %

2. CNN & RNN Processing Times

The CNN model trained faster (4.52 s Mb^{-1}) than the RNN (7.96 s Mb^{-1}) over the same number of steps and evaluated images for damage faster (0.09 s Mb^{-1} versus 0.12 s Mb^{-1}). For both models, the average rate per Mb was slowest during the data loading (0.40 s Mb^{-1} average of both models) and information exporting (0.50 s Mb^{-1} average of both models) steps. Table III.3 shows the data volumes and measured times for both models.

Table III.3. Data volumes and processing rates for different components of the CNN and RNN training and evaluation

	CNN	RNN
Training: data volume per step (Mb)	112.81	112.81
Training: average data loading rate (s Mb^{-1})	0.41	0.38
Training: average training rate per step (s Mb^{-1})	0.19	0.36
Training: average total training rate (s Mb^{-1})	4.52	7.96
Evaluation: data volume (Mb)	478.47	478.47
Evaluation: average damage detection rate (s Mb^{-1})	0.09	0.12
Evaluation: information export rate (s Mb^{-1})	0.49	0.50

F. Discussion

The summary of findings and related discussion are structured by addressing the two research questions and provide a synthesis of the results. Challenges of the study and recommendations for follow-on research are also discussed.

What is the difference in the accuracies of post-hazard damage detection products generated using RNN and CNN models?

CNNs are frequently chosen for image classification tasks; convolution kernels in neural networks can learn the spatial and spectral representations of features (Long et al. 2015; Lyndon et al. 2015). RNNs are useful in identifying and predicting changes in time-sequential data, for example in natural language processing (Yu et al. 2019). For an RNN to learn change objects in multitemporal imagery, the inputs into the LSTM node should be the spatial and spectral representations of those objects (Lyu et al. 2016; Mou et al. 2019); for this reason, convolution layers form the backbone of the RNN model in this chapter.

At least where a binary classification task is required (i.e., cracks, no cracks or damage, no damage), the overall accuracy of the CNN (96.9%, Table III.1) was comparable to that of the RNN (98.3%, Table III.2), with only a 1.4% difference. A close inspection of the results, however, shows that the CNN 2-3% more likely to predict false positives and false negatives for the damage class than the RNN (from Table III.1 and Table III.2).

These accuracy results are representative of the training class; automatically generated cracks in the simulated post-hazard time-*n* imagery. The practical application of these models to real-world post-hazard damage assessments would necessitate fully-realistic (or real) training samples. This is one area of potential future research. Both the CNN and RNN are capable, with a modification to the 'output classifier', of learning and assessing multi-class

change objects, and could be used in follow-on research in such a capacity. Given that cracks are not the only representations of damage of concern, future research testing these models on multiple damage classes could be valuable (Blong 2003).

In the context of rapid response for emergency management, how do the damage detection models compare in processing time?

Results of the processing times for the models fall into two basic categories, 1) training and 2) evaluation. The acquisition of pre-hazard imagery of critical infrastructure, pre- and post-hazard damage samples for training a model, and the model training itself are tasks best associated with 'preparedness' in the DMC. Likewise, the acquisition of post-hazard imagery, co-registration with the pre-hazard images, and the evaluation of the image pairs for damage, appear most suited to the 'relief' and 'recovery' phases of the DMC.

In the above context, where model training occurs in the 'preparedness' or inter-hazard phases of the DMC, the difference in training time of the CNN (4.52 s Mb^{-1} Table III.3) versus the RNN (7.96 s/Mb Table III.3) may be negligible. This is particularly the case when the data volume of available training imagery is small. By contrast, both models performed comparably during evaluation tests, 0.09 s Mb^{-1} (CNN, Table III.3) versus 0.12 s Mb^{-1} (RNN, Table III.3). For context, these results show the CNN capable of performing binary-class damage detection on a time-1, time-*n* pair of images from the DJI Zenmuse H20 camera in 0.52 s, and the RNN in 0.70 s.

The processing-time results presented in this chapter are limited to the specific hardware available and would vary with different hardware and data volumes. Future research into the efficiency of the software algorithms, irrespective of the hardware availability, could lead to insights into the timeliness-suitability of these models for rapid response.

G. Conclusion

Results from this chapter indicate that both the CNN (96.9%, Table III.1) and RNN (98.3%, Table III.2) models achieved overall accuracies that are "state-of-the-art" at the binary classification task of bi-temporal image change detection. Furthermore, these models, with non-stationary image frames as inputs, outperformed similar 'anomaly' detection models applied to fixed-view video frame inputs (Lyu et al. 2016; Mou et al. 2019; Pinheiro and Com 2014). Training on datasets with real-world examples of damage would be necessary to apply either model to a genuine post-hazard scenario.

While the RNN yielded marginally more-accurate damage detection products than the CNN (1.4% higher overall accuracy), the training time for the CNN was nearly half that of the RNN (4.52 s Mb^{-1} versus 7.96 s Mb^{-1}, Table III.3). The fully-trained CNN (0.09 s Mb^{-1}, Table III.3) and RNN (0.12 s Mb^{-1}, Table III.3) perform damage detection with a difference in rate of only 0.03 s Mb^{-1}, however, this difference may become substantial as the number of image pairs to evaluate increases.

IV. Time-sensitive remote sensing with sUAS-imaging and machine-learning damage detection

A. Background

The Remote Sensing Communication Model (RSCM) is a theoretical model supporting emergency managers' configurations of time-sensitive remote sensing systems (TSRSS) for emergency managers and remote sensing scientists, with the potential to address challenges in the timeliness of information delivery (Lippitt et al. 2014). The RSCM borrows from the remote sensing model (RSM) of Strahler, Woodcock, & Smith (1986) and remote sensing of ecosystems framework of Phinn et al. (2003), which are based on integrated scene and sensor models (Lippitt et al. 2014).

In emergency management situations, the availability and utility of remotely sensed information have been barriers to its use. Challenges include the amount of time required to collect, process, and deliver usable information to the decision makers, the accuracy of the information, the overwhelming volume of data, and the suitability of the data for assessing the situation (Cutter 2003; Joyce et al. 2009). The RSCM can help address these challenges by allowing the decision maker to configure a system where the needs of a situation can be balanced between desired information type, accuracy, and timeliness.

Three components of the RSCM define the configurations and effects on timeliness of a TSRSS, 1) acquisition capacity, 2) transmission capacity, and 3) analyst capacity. The information needs of an emergency manager (i.e., desired spatial, spectral and radiometric resolutions of imagery, the area of the scene to be imaged, absolute positional accuracy of data, accuracy of damage detection, identification, or other information products, and

timeliness of information delivery) can be evaluated in terms of time with the RSCM capacity analysis.

The acquisition capacity is an estimation of the timeliness of the 1) sensors, 2) platforms, and 3) scenes to be imaged. This estimation, and the quality of the data acquired, is inherently related to the configuration of these three components. For example, a larger sensor means more area covered in an image frame but also a larger volume of data acquired, a faster platform means more area covered in less time but potentially greater risk of motion blurring in images, and a lens with a wider focal-length means more area covered over time but a lower spatial-resolution. Likewise, the area and/or features in the scene(s) to be imaged will impact the time required to complete the acquisition. The decisions made regarding the platforms, sensors, and scenes in the acquisition capacity impacts the timeliness of information transmission from the sensors to the analyst, the timeliness of the analyst, and the accuracy and quality of information derived from analysis.

Equation IV.1 represents the acquisition timeliness (T_{Acq}) of a system in seconds. B_s is the total number of bits required to image the scene and C_{Acq} is the capacity of the aerial platform/sensor combination in bits imaged per second; together these represent the timeliness of the system's acquisition capacity at the scene to be imaged. T_D is the time required to move the platform from its starting location to and from the scene and T_M is the time required to transition the platform from one flight line to the next. N is the number of flights required to achieve the desired image overlap covering the scene.

$$T_{Acq} = \frac{B_S}{C_{Acq}} + T_D + T_M(N-1) \quad \textbf{(IV.1)}$$

The transmission capacity (T_{Chan}) for a TSRSS is the timeliness, in seconds (s), of the various transmission channels over which the data collected by the sensor is sent (Lippitt et al. 2014; Shannon and Weaver 1963). These channels can be physical (i.e., cables, hard disk drives) or wireless (i.e., cellular, Wi-Fi). The speed at which data can be transmitted over these channels is directly related to the chosen medium and the volume of data collected by the sensor. A user can consider the configuration of transmission channels to improve the timeliness of this component, and thus the overall TSRSS (Lippitt et al. 2014).

Equation IV.2 is the model for calculating transmission timeliness. The numerical value of B_s is determined initially by and identical to the same term in the acquisition capacity analysis. $C_{Chan,I}$ is the speed at which each transmission channel can move data. L_i represents the additional time required for each channel accounting for human interactions in the process. The method for estimating L_i in transmission capacity analysis is undefined.

$$T_{Chan} = \sum_{i}^{N} \quad 1 \frac{B_S}{C_{Chan,i}} + L_i \tag{IV.2}$$

The analyst capacity is the timeliness, (s), for an analyst, either human or machine, to produce information from the collected data for use in decision making (Lippitt et al. 2014; Lippitt, Stow, and Riggan 2016). Here, the end-user can configure a system to account for information accuracy, level of detail, method of information display, computing capability of a machine, and efficiency of a software application. Analyst capacity for automated systems applicable to the data from myriad sensors and scenes can be measured and benchmarked using Big O notation, or the relative capacity for a given sensor and scene can be empirically derived as a rate in bits processed per second (Lippitt et al. 2014; Loerch 2016). Big O notation can be derived for a given computer process by testing how long it takes a process to

complete with varying volumes of data and finding the function of the line of best fit; the highest polynomial order of the function is the Big O notation (Rubinstein-Salzedo 2018). Where the processing hardware, sensors, and scene characteristics are static, the processing rates for different analysis functions can be directly compared.

The RSCM model has been empirically tested and validated for the acquisition capacity (T_{Acq}), (Equation IV.1), and transmission capacity (T_{Chan}), (Equation IV.2) (Loerch 2016). The previous testing and validation were conducted with several piloted aircraft and small-unpiloted aerial systems (sUAS) over several sites of varying dimensions, with flight patterns designed for photogrammetric orthomosaics and digital terrain model generations. This research seeks to expand on the previous RSCM literature by evaluating different flight patterns designed for different goals, and defining/demonstrating a methodology for estimating analyst capacity. This research also seeks to update the RSCM acquisition capacity to address a missing time-cost assessment.

Although the RSCM contains a term representing human latency in transmission capacity (L_i), this is missing in the acquisition capacity estimation shown in Equation IV.1. Furthermore, L_i lacks a clearly defined methodology for estimating the human-time cost of human interactions with a process. This research proposes a useful method to account for L_i that maintains the original RSCM objective of informing the impacts of a TSRSS configuration on timeliness, while acknowledging post-positivist geographic theory related to human-subject modeling (Sharp et al. 2011; Smith 1979; Sui 1994).

Lastly, this research provides the first demonstrable application of the RSCM analyst capacity, to critical infrastructure sites in San Diego County.

B. Objectives

This chapter pertains to the acquisition and analyst capacities of the RSCM to time-sensitive remote sensing of features representing critical infrastructure, based on sUAS imaging. The purpose is to determine and demonstrate how the configurations impact timeliness and accuracy. Additionally, this research adds weight to the requirement of a human-timeliness modeling component in the acquisition capacity of the RSCM.

The following research questions are addressed in the context of post-hazard damage assessment for critical infrastructure:

1. What is the impact of sUAS-based RSI versus non-RSI (i.e., traditional) imaging missions on the acquisition capacity of a TSRSS?
2. What is the analyst capacity and accuracy trade-off between the Mask R-CNN, CNN, and RNN machine-learning models tested (in Chapters 2 and 3)?
3. How do timeliness estimates of acquisition and analyst capacities of sUAS configurations differ for RSI and non-RSI missions?

C. Methods

The acquisition and analyst capacities of the sensors and platforms, and machine-learning models from Chapters 2 and 3 were empirically derived. These capacities are compared with the accuracy results from the previous chapters. The previously flown TSRSS configurations at the San Diego State University sites are used as the basis for estimating (not empirically deriving) the timeliness of acquisition and analysis for three critical infrastructure sites in San Diego County, CA.

1. Data and Study Areas

Imagery, sensor and platform specifications, sUAS flight times, image co-registration processing times and accuracy, and CNN/RNN processing times and accuracies, actually

flown at San Diego State University and collected in Chapters 2 and 3 were used in here to determine their acquisition and analyst capacities.

sUAS imagery of sites around San Diego State University (SDSU) was collected using a DJI Mavic 1 with an FC220 camera, and DJI Matrice 300 with a Zenmuse H20 camera. These configurations were selected to represent commonly used off-the-shelf small, low-cost platforms (DJI Mavic 1) and survey-capable, larger, higher-cost platforms (DJI Matrice 300). Figure IV.1 shows the SDSU study area and Table IV.1 shows the cameras' specifications and total number of images collected for each platform using RSI and non-RSI imaging methods.

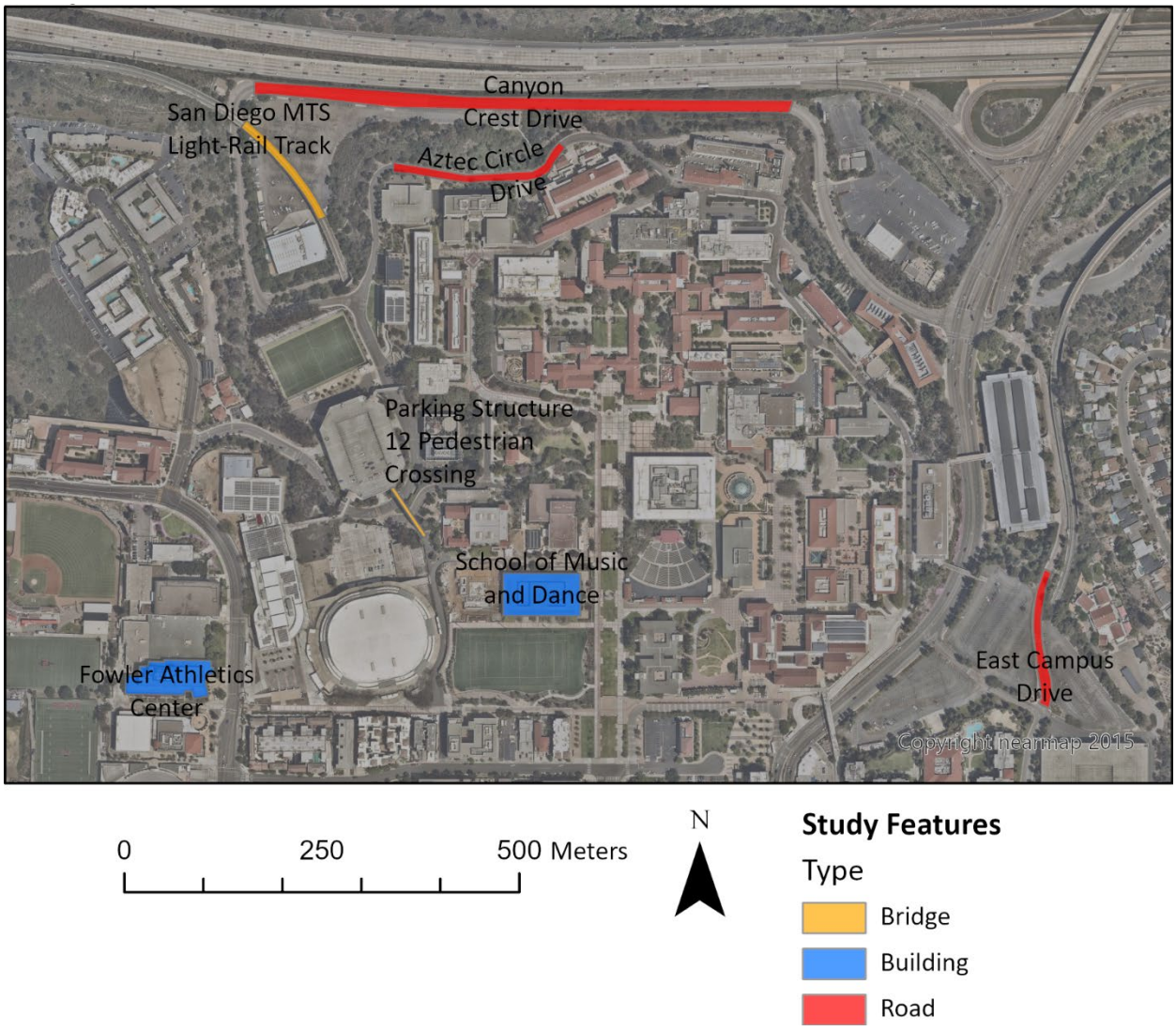

Figure IV.1. Map showing the chosen study sites within the SDSU campus

Table IV.1. Sensor specifications and image counts with RSI and non-RSI methods

	Non-RSI, DJI Mavic 1	Non-RSI, DJI Matrice 300	RSI, DJI Matrice 300
Camera Model	DJI FC220	Zenmuse H20	Zenmuse H20
Image Dimensions	4000 x 3000	5184 x 3888	5184 x 3888
Sensor Size	6.16 mm x 4.55 mm	9.50 mm x 5.70 mm	9.50 mm x 5.70 mm
Focal Length	4.7 mm	25.4 mm	25.4 mm
# Images Acquired	999	384	384

Three sites representing different types of critical infrastructure were selected for the second study area from a previous survey of critical infrastructure locations in San Diego County, CA (Loerch 2016). These sites were not flown and results not directly tested; rather, they are used as examples of how the RSCM acquisition and analyst capacities can be applied for pre-hazard planning, by determining an optimal TSRSS configuration in terms of both time and accuracy. These three sites are the San Vicente Dam, the Torrey Pines Bridge, and the San Elijo water reclamation facility. The locations of these critical infrastructure and potential sUAS camera stations (chosen to provide 1 cm GSD imagery covering a single face of each site) are shown in Figure IV.2. The sensor and platform configurations are the same as in Table 1, with the hypothetical number of images per site and RSI imaging / traditional imaging (non-RSI) reflected in Table IV.2. Flight distances in Table IV.2 were measured in ArcGIS Pro for each linear critical infrastructure site with an oblique view perspective (San Vicente Dam and the Torrey Pines Bridge) as the distance from one end of the site to the other. For the area-based nadir view perspective of the San Elijo water reclamation facility, the flight distance was determined by ensuring a complete coverage of the buildings and water-holding tanks based on a 1 cm ground-sample-distance (GSD) and the sensors' ground coverage extents. The numbers of images required for each platform/sensor and CI site is based on a 1 cm GSD, the necessary distance from the platform to the feature to achieve that GSD, and complete, non-overlapping coverage the flight distance with the image extents for the various sensors listed.

Figure IV.2. Locations of three critical infrastructure sites in San Diego County: A) San Vicente Dam, B) Torrey Pines Bridge, C) San Elijo Water Reclamation Facility. Example camera stations for the Mavic 1 and Matrice 300 are shown.

Table IV.2. Number of images required to fully inspect a single face of three critical infrastructure sites in San Diego County at 1 cm GSD using the RSI and non-RSI methods with the DJI Mavic 1 and DJI Matrice 300.

	San Vicente Dam (oblique view)	Torrey Pines Bridge (oblique view)	San Elijo Water Reclamation (nadir view)
DJI Mavic 1, non-RSI	14 images flight distance = 519 m	5 images flight distance = 160 m	10 images flight distance = 128 m
DJI Matrice 300 RSI and non-RSI	11 images flight distance = 505 m	4 images flight distance = 144 m	6 images flight distance = 145 m

2. RSCM Capacity Measurements and Estimation

Leveraging the data availability from the DJI Mavic 1, DJI Matrice 300, associated cameras, and time-monitoring functions in python applications, evaluation of the automated portions of the acquisition and analyst capacities for the configured platforms, sensors and applications was straightforward. Modeling the human interaction and timeliness components of these capacities, however, requires additional considerations and may not be tractable.

The acquisition capacities of the given configurations in Table IV.1 for the SDSU study area were determined for the automated portions (takeoff, image acquisition, landing) using Equation 1. The resultant capacities are presented in data bits acquired per m s^{-1}. Estimates, based on these capacities, would be presented in seconds, as was done with the CI sites' timeliness estimates.

The Lippitt et al. (2014) representation of Equation 1 of the RSCM lacks a term integrating human latency into acquisition capacity configurations. Equation IV.3 shows the RSCM acquisition capacity, modified to account for this term. The time required to prepare a sUAS for flight varies by platform, scene, and the human interactions between them. Additionally, in the case of RSI flights using RTK GNSS, the interactions between a human

and the RTK system add additional complexity to modeling acquisition capacity and estimating time. Therefore, the empirically-derived measurements of acquisition capacities rates for the tested TSRSS configurations in this dissertation include the term L_i, where i is the number of human-related processes that increase complexity and decrease timeliness.

$$T_{Acq} = \frac{B_S}{C_{Acq}} + T_D + T_M(N-1) + L_i \quad \textbf{(IV.3)}$$

A means for actually modeling L_i as a rate, and estimating L_i in seconds is beyond the scope of this study. When modelling the timeliness of automated processes such as a software function, information about the computer processing hardware, data volumes, and processing times must exist (Rubinstein-Salzedo 2018). The research and methods presented here acknowledge that this human latency component is important, and that a meaningful representation of a human configuration does not exist as it does with computing hardware, thus any empirically-derived or modelled capacity of humans is not attempted and likely not appropriate. The proposed and applied method for using L_i, maintains the utility of the RSCM for aiding emergency managers in configuring and estimating their TSRSS.

The human-related processes assigned to L_i for acquisition capacity are additive. For non-RSI, non-RTK flights, L_i is 1, representing the platform and sensor setup and pre-flight preparations. For RSI flights with RTK, L_i is 2, representing the additional process of setting up and utilizing an RTK GNSS system. The utility of this ordinal approach in configuring a TSRSS is evident; a more complex set of human processes will take longer than a less complex approach, for a single person. If the processes can be done simultaneously, as in the case of pre-flight preparations and RTK system setup, an emergency manager could engage two people for the tasks, or find a single person with experience performing both efficiently.

The analyst capacities of the automated software algorithms were determined individually for the image co-registration process, the Mask R-CNN from Chapter 2, and the CNN and RNN from Chapter 3. Within the custom python software applications, the 'time' module was used to track the time to completion for each image or image pair. The number of bits of data processed in this way were also tracked, to provide the analyst capacities in bits processed per second.

Estimation of the acquisition capacities for the three critical infrastructure sites in San Diego County were based on Equation IV.1, Table IV.1 and Table IV.2, with the human-interaction term added as previously described. The analyst capacities were estimated from the empirical testing performed with the SDSU sites. Image co-registration analyst capacities are estimated for each TSRSS configuration, while the three machine-learning analyst capacities are estimated only for the DJI Matrice 300 with the RSI collection configuration. Data volumes are based on the estimated number of images to acquire, as shown in Table IV.2.

D. Analytical procedures

To address the research questions in this component, the timeliness and accuracy metrics of the sUAS/sensor combinations and three neural network models were compared. The acquisition capacities of the TSRSS for each site at SDSU are presented in bits m^{-1} s^{-1} (bits of image data, flight-length meters, flight time seconds) the accuracy metric for the image co-registration process is presented in MAE of pixel locations, the Mask R-CNN accuracy metric is presented in mean IoU, and the CNN and RNN are presented in overall accuracy percent. This method allows for a qualitative evaluation of timeliness versus accuracy for RSI versus non-RSI acquisition methods and several methods of analyst capacity. The

machine-learning analyst capacities for the non-RSI TSRSS configurations were not evaluated.

A quantitative approach for analyzing the tradeoff between accuracy and timeliness is not conducted as the information needs of an emergency manager may commonly select configurations that weights timeliness over accuracy. For example, if an emergency response decision is so slow that cannot be made in sufficient time to be effective (e.g., save lives and properties), the accuracy may be of secondary concern. Also, it is challenging to intractable to value or quantify accuracy in meaningful units that compare with time. While it is left to the emergency managers to decide which of these is most important for any given task, the demonstrated analytical method provides the information needed for those decisions to be made.

The acquisition and analyst timeliness capacities for each TSRSS are presented and evaluated similarly to the SDSU sites, for the three critical infrastructure sites in San Diego County. The purpose is to demonstrate the ability for a user to configure a TSRSS prioritizing timeliness, accuracy, or a balanced approach using the RSCM. The analysis shows the most appropriate configurations for timeliness and accuracy.

E. Results

The results are split into three sections for ease of comparison, 1) acquisition capacity and image co-registration analyst capacity, 2) machine-learning analyst capacity, and 3) acquisition and analyst timeliness estimations for San Diego County critical infrastructure sites.

1. Acquisition Capacity and Image Co-registration Analyst Capacity

Results of the acquisition capacity for the configured TSRSS at the SDSU sites show the DJI Matrice 300 with the non-RSI acquisition method to have the greatest timeliness, at 1.02 x 10^7 bits m^{-1} s^{-1}. The DJI Matrice 300 with the RSI acquisition method exhibited greater timeliness than the DJI Mavic 1 with non-RSI. However, the human-interaction complexity term associated with the RSI acquisitions indicates that the DJI Matrice 300 with the RSI method has the lowest overall acquisition capacity of tested configurations, as elaborated upon later.

The analyst capacity of image co-registration varied greatly between the Mavic 1 image pairs (2.38 x 10^6 bits s^{-1}), and those from the Matrice 300 non-RSI (1.30 x 10^7 bits s^{-1}) and RSI (1.16 x 10^7 bits s^{-1}). Of these, the Matrice 300 non-RSI configuration had the greatest timeliness capacity, while the Matrice 300 RSI configuration had the greatest co-registration accuracy (2.2 pixels MAE). For comparison of the acquisition capacities, image co-registration analyst capacities, and image co-registration accuracies, as shown in Table IV.3.

Table IV.3. Average acquisition and image co-registration analyst capacities of the evaluated TSRSS configurations for the SDSU study sites. Acquisition capacities are in bits m^{-1} s^{-1} and include human-interaction complexity (L_i). The image co-registration analyst capacity is in bits s^{-1}, along with the co-registration accuracies in pixels MAE.

	Mavic 1 (non-RSI)	Matrice 300 (non-RSI)	Matrice 300 (RSI)
Acquisition Capacity (bits m^{-1} $s^{-1)}$	3.90 x 10^6	1.02 x 10^7	7.40 x 10^6
Human-interaction Complexity (L_i)	1	1	2
Co-registration Analyst Capacity (bits s^{-1})	2.38 x 10^6	1.30 x 10^7	1.16 x 10^7
Co-registration Accuracy (pixels MAE)	75.1	4.4	2.2

2. Machine-learning Analyst Capacity

Of the three machine-learning models' analyst capacities evaluated with the imagery from the Matrice 300 using the RSI collection method, the CNN from Chapter 3 had the greatest timeliness (9.00 x 10^7 bits s^{-1}) and the Mask R-CNN had the lowest (5.23 x 10^7 bits s^{-1}). The RNN from Chapter 3 had the highest average overall accuracy for bridges, buildings and roads (98.4%). Table IV.4 shows the analyst capacities of each machine-learning model and their accuracies for the RSI TSRSS configuration.

Table IV.4. Analyst capacities in bits/s and associated accuracy results for the Mask R-CNN, CNN, and RNN models used to detect damage (cracks) in buildings, bridges and roads.

	Mask R-CNN	CNN	RNN
Analyst Capacity (bits s^{-1})	5.23 x 10^7	9.00 x 10^7	6.68 x 10^7
Accuracy	83.7% mIoU	96.9% OA	98.4% OA

3. Acquisition and Analyst Timeliness Estimates for San Diego County CI Sites

The Matrice 300 with the non-RSI acquisition method had the greatest acquisition timeliness for the San Vicente Dam (458.6s + L_i^1s), Torrey Pines Bridge (140.1s + L_i^1s) and San Elijo Water Reclamation Facility (141.0s + L_i^1s. Results of the acquisition capacity timeliness estimates for all three CI by TSRSS configuration are presented in Table IV.5.

Table IV.5. Timeliness estimates for the acquisition capacities of configured TSRSS at three CI sites in San Diego County, CA.

	San Vicente Dam	Torrey Pines Bridge	San Elijo Water Reclamation Facility
Mavic 1 (non-RSI)	747.0s + L_i^1	240.2s + L_i^1	195.0s + L_i^1
Matrice 300 (non-RSI)	458.6s + L_i^1	140.1s + L_i^1	141.0s + L_i^1
Matrice 300 (RSI)	638.0s + L_i^2	193.2s + L_i^2	194.4s + L_i^2

Timeliness estimates from the analyst capacities of image co-registration shows again that the Matrice 300 non-RSI TSRSS is most timely for the San Vicente Dam (140.2 s), Torrey Pines Bridge (51 s) and San Elijo Water Reclamation Facility (76.4 s). The timeliness

estimates for the three machine-learning models with data from the Matrice 300 RSI configuration shows the Chapter 3 CNN is most timely for the San Vicente Dam (20.3 s), Torrey Pines Bridge (7.4 s) and San Elijo Water Reclamation Facility (11.1 s). Table IV.6 shows the analyst capacity timeliness estimates for the image co-registration and machine-learning processes.

Table IV.6. Timeliness estimates from the analyst capacities of configured TSRSS for image co-registration and machine-learning damage detection models.

	San Vicente Dam (s)	Torrey Pines Bridge (s)	San Elijo Water Reclamation Facility (s)
Analyst Capacity – Image Co-registration			
Mavic 1 (non-RSI)	366.8	131.0	262.0
Matrice 300 (non-RSI)	140.2	51.0	76.4
Matrice 300 (RSI)	156.9	57.0	85.6
Analyst Capacity – Damage Detection			
Mask R-CNN	34.9	12.7	19.0
CNN	20.3	7.4	11.1
RNN	27.3	9.9	14.9

F. Discussion

A synthesis of findings and related discussion are presented here in the context of the research questions. Study limitations and future research potential are also discussed.

While this study and the results are applicable specifically to the listed platforms, sensors, scenes characterized, and analyst algorithms, the primary intent of the RSCM is to provide a framework for estimating accuracy and timelines of different configurations of these components. Previous research on the RSCM was either theoretical or did not apply to actual implementations of analyst capacity and did not account for any useable method of modelling human latency in a process (Lippitt et al. 2014; Loerch 2016). This study demonstrates an applied approach to the RSCM, filling the gaps in the extant literature.

The assessments of analyst capacity and accuracy are based on four assumptions: 1) pre-hazard imagery was already collected, 2) a machine-learning model was already trained on pre- and post-hazard image pairs containing representative manifestations of damage, 3) the acquisition capacities are based on a specific, limited set of TSRSS platforms and sensors meant to show the relative timeliness impacts of RSI and non-RSI missions and 4) the analyst capacities are based on measurements from a specific set of computing hardware (Appendix A). These assessments are meant to understand the relative impacts of different analyst algorithms on timeliness.

What is the impact of sUAS-based RSI versus non-RSI imaging missions on the acquisition capacity of a TSRSS?

From the results of the Matrice 300 flown with non-RSI methods and the Matrice 300 flown with RSI, it is clear that when critical infrastructure and platform/sensor combinations are the same, sUAS-based RSI missions have a lower acquisition capacity and lesser timeliness than their non-RSI counterparts. This finding can be attributed in-part to time cost associated with establishing the *in-situ* observation that the Matrice 300 takes longer to reach camera stations and acquire images when RTK GNSS is enabled. Attribution of RTK GNSS navigation as the primary factor effecting the difference in acquisition capacities of the Matrice 300 with RSI versus non-RSI methods is based on the enabling of RTK GNSS as the only modified variable between those methods for the Matrice 300.

The acquisition capacity of the Matrice 300 with the non-RSI acquisition method is 10.8% greater than with the same platform and the RSI method, based solely on the measurable parameters from Equation 1. However, the human interaction with the sUAS flight-preparation is more complex for an RSI acquisition, as it requires setting up and/or

connecting to an RTK GNSS base unit. This study of acquisition capacities makes no attempt to directly model and measure the timeliness of human interactions with the TSRSS; rather, it presents an estimate of complexity in the term L_i. Given the difference in complexity (an additional process requirement) between the RSI and non-RSI methods, it is reasonable to conclude that the actual acquisition capacity of a similarly configured TSRSS flying non-RSI missions is much higher than one flying RSI missions.

The Mavic 1, an integrated platform and sensor system incapable of RSI missions, had the lowest measurable acquisition capacity. This is explainable in terms of the sensor size and focal length. With an image width of 4,000 pixels, collecting images at 1 cm GSD, each image from the Mavic 1 can cover a lateral extent of 40 m. Compared with the Zenmuse H20 on the Matrice 300 (5,184 pixels width, 51.8 m lateral coverage per 1 cm GSD image), the Mavic 1 collects fewer bits per image and requires more images to cover the same area. Despite the lower acquisition capacity of the Mavic 1 compared to the Matrice 300, it is still possible that given the human interaction complexity of RSI missions, the acquisition capacity of the Mavic 1 may be higher than the Matrice 300 using RTK for RSI.

What is the analyst capacity and accuracy trade-off between the machine-learning Mask R-CNN, CNN and RNN models?

The CNN had the highest analyst capacity (9.00×10^7 bits/s) and second highest accuracy (96.9% OA) of crack detections. This model used convolution filters of sizes 3x3 and 5x5 with a stride of 1 across the entire images and binary class labels applied to individual 256 x 256 pixel image tiles, to identify whether tiles contained cracks or not (Chapter 3). The Mask R-CNN had both the lowest analyst capacity (5.23×10^7 bits/s) and accuracy (83.7% mIoU). The Mark R-CNN used region-based convolution filters, class-based bounding boxes as

labels, and image segmentation to identify the cracks and their locations ((He et al. 2017). The RNN model yielded the highest accuracy (98.4% OA) and had the second highest analyst capacity (6.68 x 10^7 bits/s). The RNN structure included an identical configuration of convolution filters, class labeling and damage identification as the CNN, and added a layer of 64 LSTM recurrent network cells (Chapter 3, Pinheiro & Com, 2014). The CNN is 25.8% more time-efficient than the RNN and 1.5% less accurate. Given this trade-off between timeliness and accuracy, a user prioritizing timeliness may choose the CNN model over the RNN, and conversely where accuracy is the priority, may choose the RNN.

The lower performance of the Mask R-CNN for both analyst capacity and accuracy is likely attributable to the model's algorithmic complexity (method of change-object detection and presentation) and training data requirements. Where the CNN and RNN evaluate image tiles of dimensions 256 x 256 pixels and outputs whether the tiles contain post-hazard damage or not, the Mask R-CNN model processes larger tiles of dimensions 1024 x 1024 pixels, and outputs the specific locations of post-event damage in those tiles. The result is that the Mask R-CNN has greater complexity and therefore a lower analyst capacity compared with the other two models. Another result of the increased complexity of the Mask R-CNN model is that it likely requires more training data, not available for this study, to achieve a higher accuracy.

How do timeliness estimates of acquisition and analyst capacities of sUAS configurations differ for RSI and non-RSI missions?

For the three critical infrastructure sites evaluated, the maximum acquisition time was estimated with the Mavic 1 flying a non-RSI mission at the San Clemente Dam (12 minutes, 27 seconds) and the shortest acquisition time was estimated with the Matrice 300 flying a

non-RSI mission at the Torrey Pines Bridge (2 minutes, 20 seconds). The non-RSI missions are estimated to have greater timeliness than the RSI missions for each of the critical infrastructure types. The choice of specific TSRSS platform and sensor combinations has a greater impact on estimated timeliness, excluding human interaction complexity, than the choice of RSI versus non-RSI missions. This is the result of differences in sensor sizes and focal lengths.

The timeliness of analyst capacities for the image co-registration process varies by platform/sensor combination and RSI versus non-RSI missions. The timeliness for image co-registration with the Mavic 1 is the lowest across each critical infrastructure type. Given the low co-registration accuracy of the Mavic 1 image pairs (75.1 pixels MAE), it is likely that the analyst capacity for the Mavic 1 is impacted by complexities in locating good matching points between images and computing the transformations for the smaller, wider-angle lens for image co-registration. The Matrice 300 non-RSI configuration consistently has the timeliest estimates for the co-registration process across critical infrastructure types. This configuration yielded much higher accuracy (4.4 pixels MAE) than the Mavic 1, and the higher timeliness may be attributable to greater overlap between image pairs and less complex transformations related to the optical distortions of its lens.

Timeliness estimates from the analyst capacities of the machine-learning models were created for a single TSRSS platform/sensor configuration, the Matrice 300 flying with RSI. The rationale for not evaluating the timeliness of these models for the other configurations is that analyst capacity modeling is a function of the input data volume and dimensions. These do not vary between the non-RSI and RSI missions once the co-registration process is completed. Of the estimates, the CNN for Torrey Pines Bridge was most timely (7.4s) and

the Mask R-CNN for San Vicente Dam was least timely (34.9s). For all three critical infrastructure sites, the CNN model had the greatest timeliness, followed by the RNN model, and lastly, the Mask R-CNN.

G. Conclusion

This chapter presents a time-sensitive approach to RSI and non-RSI image acquisitions and post-hazard damage detection with machine-learning algorithms. It builds on the RSCM by adding an approach to evaluating human complexity in the acquisition capacity, and establishes an approach for estimating analyst capacity. This enhanced RSCM approach was applied to examples of critical infrastructure in San Diego County, demonstrating its utility in optimally configuring a TSRSS.

The acquisition capacities of three TSRSS platform, sensor, and collection method combinations were empirically derived from three SDSU sites representing three critical infrastructure types. Of these combinations, the non-RSI missions utilizing the larger Zenmuse H20 sensor had the highest acquisition capacity. Missions flown with RSI and the Zenmuse H20 sensor likely have the lowest acquisition capacity of the evaluated configurations due to the inclusion of RTK GNSS navigation and human interaction complexity.

The analyst capacity measurements for the SDSU sites and timeliness estimates of San Diego County critical infrastructure sites shows that the non-RSI missions flown with the Matrice 300 had the highest image co-registration analyst capacity, while those flown using RSI-missions had the greatest co-registration accuracy. Analyst capacity measurements for the three machine-learning models, using the RSI-based datasets, shows that the CNN has the greatest capacity and timeliness, while the RNN yields the greatest accuracy.

Future research should focus on refining the RSCM, TSRSS configurations, and the Mask R-CNN, CNN, and RNN. A successful method for modeling of human timeliness that provides estimates in seconds is not currently available. The ability to model the analyst capacity of a software process in a generalizable way (for varied hardware configurations) likely exists through a combination of big O notation and the availability of additional sUAS hardware and image datasets ((Rubinstein-Salzedo 2018). Technological progress of TSRSS platforms and sensors, particularly improvements in precision navigation, have and may continue to lead to improvements in the repeatability of image collections. Finally, as with the majority of machine learning models, room exists for improvements in efficiency through programming structures and accuracy through additional training data samples.

V. Conclusions

For Chapter 2, I evaluated the impacts of state-of-the-art small unpiloted aircraft systems' (sUAS) navigation with and without real-time-kinematic (RTK) corrections for repeat station imaging on the repeatability and co-registration of very high-resolution (1 cm ground sample distance (GSD) image pairs. The impacts of image pair mis-registration on a machine-learning-based change detection model (Mask R-CNN) were then assessed.

I compared the change detection accuracies, training and processing times, between a convolutional neural network (CNN) and a similarly structured recurrent neural network (RNN) that contained long-short-term-memory nodes in Chapter 3. Samples of post-hazard damage were automatically generated in a form simulating cracks, and added to time-2 images of buildings, bridges and roads. For the CNN, the bitemporal images were combined into single-image layer stacks where change objects could be detected as the difference in brightness values, object shapes, size and texture from image band 1 to image band 2. For the RNN, the image pairs were provided as time-sequential images, where the model effectively learns the parameters of a scene without damage compared to the parameters of a scene with damage.

In the context of the Disaster Management Cycle (DMC), Chapter 4 entailed an evaluation of sUAS, their imaging sensors, and navigation capabilities for acquisition capacity timeliness using the Remote Sensing Communication Model (RSCM). The RSCM is updated in Chapter 4 to account for human interaction complexity in acquisition capacity modelling. This research also adds and demonstrates an analytical methodology for determining analyst capacity, a previously underdeveloped component of the RSCM.

Estimates of timeliness using the updated RSCM acquisition and analyst capacities for three sites representing critical infrastructure in San Diego County are generated to demonstrate the utility of the RSCM, repeat station and traditional imaging, and crack detection analysis for real-world applications.

This final chapter highlights the key findings of the research, provides recommendations for future research, and details the novel elements and key innovations stemming from the dissertation.

A. Key Findings

The key findings of the research pertain to 1) impacts of sUAS navigation capabilities and image collection methods on image co-registration accuracy, 2) impacts of image co-registration accuracy on the ability of a machine learning model (Mask R-CNN) to accurately detect simulations of post-hazard damage in bitemporal images, 3) accuracy and utility differences between single bi-temporal layer-stacked images used for damage detection with a CNN versus time-sequential image pairs used for damage detection with an RNN, and in the context of time-sensitivity, 4) an analysis of the RSCM acquisition capacities of sUAS flown using repeat station and traditional imaging methods, 5) an analysis of the RSCM analyst capacities for the co-registration of images acquired using repeat station and traditional imaging, 6) an analysis of the analyst capacities of the Mask R-CNN, CNN, and RNN, and 7) an analysis of the information user's (as defined by the RSCM) choice of sUAS, image collection method, and machine-learning damage detection model on both information timeliness and accuracy.

1. Impacts of navigation capabilities' and image collection methods' of sUAS on image co-registration accuracy

The results of the navigational and co-registration accuracy assessments clearly demonstrate that 1) repeat station imaging using RTK global navigation satellite systems (GNSS) resulted in greater repeatability of camera station waypoints (0.16 m mean absolute error) compared to traditional imaging (2.4 m mean absolute error), 2) repeat station imaging resulted in greater overlap of image pairs (9.4 mean percent error) compared with traditional imaging (16.5 mean percent error), and 3) repeat station imaging resulted in higher image co-registration accuracy (2.2 pixels mean absolute error) compared with traditional imaging (4.4 pixels mean absolute error for the Matrice 300 and 75.1 pixels mean absolute error for the Mavic 1).

2. Impacts of image co-registration accuracy on the ability of a machine-learning model (Mask R-CNN) to accurately detect simulations of post-hazard damage (cracks) in bitemporal images

The Mask R-CNN neural network classifier yielded the highest accuracy (83.7% overall accuracy) when applied to the repeat station imaging-based bi-temporal images co-registered with an accuracy of 2.2 pixels mean absolute error. This is compared to the traditional imaging acquisition approach, where the Mask R-CNN achieved 72.5% overall accuracy for images co-registered with 75.1 pixels mean absolute error and 54.0% overall accuracy for the images co-registered with 4.4 pixels mean absolute error.

3. Accuracy and processing-time differences between single bi-temporal layer-stacked images used for damage detection with a CNN versus time-sequential image pairs used for damage detection with an RNN

The mean overall accuracy of the RNN (98.3%), evaluated on time-sequential image pairs and the CNN (96.9%), evaluated on single bi-temporal layer stacked images, indicates very successful detection of simulated cracks with comparable performance at detecting post-hazard damage. The accuracy of crack detection with the RNN model accuracy was 1.4% higher than the CNN. The user's accuracy (compliment to errors of commission) and producer's accuracy (compliment to errors of omission) for the 'damage' class were lower than for the 'no damage' class across view perspectives with both the CNN and RNN, with accuracies for the damage class ranging from 93.0% to 97.6%.

The training time required for the CNN was lower (4.52 s · megabyte^{-1}) compared to the RNN (7.96 s · megabyte^{-1}). This information is relevant in the disaster preparedness phase of the DMC, when the training of models to detect damage is most likely to occur (Joyce, Wright, et al. 2009). The rate of damage detection was fastest for the CNN (0.09 s · megabyte^{-1}) compared to the RNN (0.12 s · megabyte^{-1}). Although the rates appear similar, as the data volume for the models to evaluate becomes larger, so do the overall evaluation times. This information is relevant in the relief and recovery phases of the DMC (Joyce, Wright, et al. 2009).

4. Analysis of the RSCM acquisition capacities of sUAS flown using repeat station and traditional imaging methods

Results of the acquisition capacity for the configured time-sensitive remote sensing systems (DJI Mavic 1, DJI Matrice 300) at the SDSU sites show the DJI Matrice 300 flown using the traditional acquisition method had the greatest timeliness at 1.02 x 10^7 bits m^{-1} s^{-1}. The DJI Matrice 300 flown using the repeat station imaging acquisition method exhibited

greater timeliness than the DJI Mavic 1 flown with the traditional acquisition method. However, the human-interaction complexity term, added to the RSCM acquisition capacity and applied in this research, indicates that the repeat station imaging acquisitions likely have the lowest overall acquisition capacities of the tested configurations.

5. Analysis of the RSCM analyst capacities for the co-registration of images acquired using repeat station and traditional imaging

The analyst capacity of image co-registration varied greatly between the Mavic 1 image pairs (2.38×10^6 bits s^{-1}), and those from the Matrice 300 flown with the traditional imaging method (1.30×10^7 bits s^{-1}) and the repeat station imaging method (1.16×10^7 bits s^{-1}). Of these, the Matrice 300 traditional imaging configuration had the greatest timeliness capacity.

6. Analysis of the analyst capacities of the Mask R-CNN, convolutional neural network, and recurrent neural network

Of the three machine-learning models' analyst capacities evaluated with the imagery from the Matrice 300 flown with repeat station imaging, the CNN had the greatest timeliness (9.00×10^7 bits s^{-1}) and the Mask R-CNN had the lowest (5.23×10^7 bits s^{-1}). The RNN had a higher analyst capacity (6.68×10^7 bits s^{-1}) than the Mask R-CNN but was lower than the CNN.

7. Analysis of the information user's (as defined by the RSCM) choice of sUAS, image collection method, and machine-learning damage detection model on both information timeliness and accuracy

The acquisition capacities of three remote sensing platform, sensor, and collection method combinations were empirically derived from three SDSU sites representing three

critical infrastructure types (buildings, bridges and roads). Of these system combinations, the traditional imaging missions utilizing the larger Zenmuse H20 sensor had the highest acquisition capacity (1.02×10^7 bits m^{-1} s^{-1}) and missions flown with repeat station imaging and the Zenmuse H20 sensor likely have the lowest acquisition (7.4×10^6 bits m^{-1} s^{-1}) due to the inclusion of RTK navigation corrections and human interaction complexity. The finding is that with an identical platform and sensor combination, the choice of the traditional imaging method will lead to a faster acquisition of imagery compared to use of repeat station imaging.

While the timeliness is greater for traditional imaging, the co-registration accuracy of traditional imaging is lower (75.1 pixels mean absolute error for the Mavic 1 and 4.4 pixels mean absolute error for the Matrice 300) than repeat station imaging (2.2 pixels mean absolute error for the Matrice 300). This finding is also reflected in the accuracy results of the Mask R-CNN, where the images collected and co-registered from the timelier traditional imaging method resulted in lower damage detection accuracies (72.5% overall accuracy for the Mavic 1 image pairs and 54.0% overall accuracy for the Matrice 300 image pairs) compared with the less timely repeat station imaging method (83.7% overall accuracy).

B. Recommended Research

Research recommendations from this dissertation encompass: 1) additional sUAS configurations focused on precise and accurate navigation, 2) additional image pre-processing steps and algorithms, 3) development and testing of additional machine learning models, and 4) RSCM capacity estimations based on more generalizable parameters (in analyst capacity) and alternative methods of human complexity modelling.

1. Additional sUAS configurations focused on precise and accurate navigation

The use of RTK based navigation of sUAS poses some challenges, including: 1) setting up a local RTK base receiver or remotely accessing an RTK base is required, 2) providing remote access of a RTK base receiver requires internet data availability, 3) environmental conditions like wind and radio interference can lead to less-accurate navigation, and 4) proprietary sUAS autopilot systems like those from DJI may handle horizontal and vertical navigation by RTK differently than other manufacturers' systems. Technical advancements in the delivery of RTK corrections to GNSS receivers and the implementation of RTK on sUAS are expected to increase overall navigation accuracy and convenience. Additional research and engineering efforts to achieve repeatable image collections from aerial systems is warranted considering the effects the repeatability has on image co-registration and change detection accuracy.

2. Additional image pre-processing steps and algorithms

While the navigation capabilities of small unpiloted aerial platforms and available GNSS solutions such as RTK may continue to improve, other factors can affect automated image co-registration and could be explored further. Detection and elimination of image match points along the boundaries of transient shadows (i.e., shadows which shift position in bi-temporal image pairs) could result in lower co-registration errors (Storey et al. 2017). Additionally, accurate image alignment and generation of photogrammetric points clouds relies on the identification and correction of lens distortions (Turner et al. 2012). It is reasonable to conclude that uncorrected distortions in the imagery may reduce the overall co-registration accuracy. Finally, exploration of other algorithms for image co-registration is warranted; this research used the Oriented Fast and Rotated Brief and Hamming Distance for

feature and match point detection in order to co-register images (Karami et al. 2017; Norouzi et al. 2012).

3. Development and testing of additional training data and machine-learning models

Results from this dissertation are representative of the training data and developed machine learning models utilized in testing. Training data were comprised of automatically generated cracks in simulated post-hazard time-*n* imagery. The practical application of the training data to real-world post-hazard damage assessments would necessitate fully-realistic (or real) training samples of damage of interest. This is an area of potential future research. Both the convolutional neural network and recurrent neural network are capable, with a modification to the 'output classifier', of learning and assessing multi-class change objects. Given that cracks are not the only representations of damage of concern, future research testing these models on multiple damage classes could be valuable (Blong 2003). Furthermore, minor modifications to the structure of a CNN can impact the amount of training data required, the number of classes a model can learn to detect, and the accuracy of those detections. This indicates that myriad extent neural network models, and yet to be developed ones could be assessed using the image collection and data preparation methodologies outlined in this research and potentially achieve higher accuracies at faster rates of evaluation.

4. RSCM capacity estimations based on more generalizable parameters (in analyst capacity) or alternative methods of human complexity modelling

The RSCM, developed initially in 2014 and updated in 2016, is a relatively new model for time-sensitive remote sensing (Lippitt et al. 2014; Loerch 2016). This research adds to the

RSCM a term for assessing the human complexity component in the acquisition capacity of a remote sensing system and provides a method for determining the analyst capacity of a system.

However, further research on improving the RSCM is warranted. A successful method for modelling of human timeliness that provides estimates with greater temporal precision (e.g., seconds) is not currently available. The ability to model the analyst capacity of a software process in a generalizable way (for varied computer processing hardware configurations) is tractable through a combination of big O notation and the availability of additional hardware and datasets (Rubinstein-Salzedo 2018).

C. Novel Elements and Key Innovations

This section provides a comprehensive summary of the novel elements and key innovations associated with this dissertation. It addresses the gaps in the extant literature and use cases of repeat station imaging, change detection with machine-learning models, and the timeliness estimation of remote sensing systems with the RSCM.

1. Repeat Station Imaging, sUAS Navigation, and Image Co-registration

Precisely and accurately capturing bi-temporal aerial images from/with the same geolocations, view perspectives, and camera results in the ability to directly and more accurately co-register the images (Coulter et al. 2003). Achieving this capability of repeating image captures from a moving aerial platform is challenging, and quantifications and assessments in the extant literature which emphasize the challenges of this approach exclusively address either full-frame camera systems mounted on fixed-wing, piloted aircraft, or satellite systems (Coulter et al. 2003; Stow et al. 2016). Chapter 2 of this dissertation provides the first application and quantification of image co-registration and challenges

associated with capturing images using repeat station imaging from sUAS with very high resolution (1 cm GSD) imagery, small-frame sensors, and short focal-length lenses (increased geometric distortions in images).

The availability of RTK GNSS corrections for use in sUAS navigation available to users is relatively new (since 2016). Published use cases of this capability and navigation accuracy focus either on photogrammetric mosaicking and single-flight absolute positional accuracy measurements for navigation, and not for repeated image collections from the same positions over time (Deng, Wang, and Low 2021; Gabrlik et al. 2018). Chapter 2 of this dissertation provides a novel evaluation of the relative repeatability of image captures from sUAS with RTK capabilities for the purpose of reducing errors in image co-registration. It also makes a clear case for use of RTK GNSS, not simply in the current majority of applications for geotagging (i.e., location based on a single latitude, longitude, altitude value without respect to orientation and scale) images with absolute positional accuracy constraints, but within the context of repeating camera stations for accurate image co-registration when change detection is the goal.

2. Machine-learning Approaches to Change Detection

Several general considerations guide the selection or development of machine-learning models for change detection: 1) structure of the multi-time imagery (i.e., number of spectral bands, image and object scales, 2) presence and / or precision of multi-time image alignment), method and 3) structure for providing imagery to the model (i.e., as single images for post-classification change detection, as bitemporal layer stacks for change object detection, as time-sequential images for scene anomaly detection), manner in which images

will be labelled for the training process (if at all), the model structure (i.e., number and type of network layers, error assessment and learning methods, type of image classifier), and the format of the information output by the models (Dumoulin and Visin 2016; Long et al. 2015; Lyndon et al. 2015; Lyu et al. 2016; Pinheiro and Com 2014).

The very nature of machine-learning models is such that 'novelty' for machine-learning application-based research tends to be defined differently than for computer-science research into machine-learning itself. For example, is a convolutional neural network 'novel' due the number and layout of the layers, the application it is applied to, or is it the case that no convolution neural network is truly novel save the first published use?

Chapter 2 demonstrated the training and application of an extant machine-learning model, Mask R-CNN, on single-image bitemporal layer stacks for the purpose of change object detection. Despite the reputation of Mask R-CNN as a state-of-the-art machine-learning algorithm for rapid object detection, no literature was identified for use of a similar application of the Mask R-CNN to change-object detection.

Chapter 3 adapts fairly standard structures for a CNN and an RNN, containing layers of convolution nodes, fully-connected nodes for decision making, and binary classifiers, for the specific purpose of detecting change objects in single, bitemporal layer-stacked images and in time-sequential image pairs. The CNN and RNN models developed for this research are 'novel' because of the combination of the number of convolution nodes and kernel dimensions; they are not 'off-the-shelf.' The RNN structure for this dissertation, for example, differs from Mou et al. (2019), who developed an RNN used the outputs of long-short-term-memory nodes trained on convolution layers for the time-1 images as inputs to the long-short-term-memory nodes trained on time-*n* images. In comparison, the RNN developed and

tested in Chapter 3 used long-short-term-memory nodes trained on the convolution layers of the time-1 and time-*n* images together. As far as I can determine, Mou et al. 2019 and this dissertation study are the only two studies to have employed RNNs in this manner for the detection of changes in remotely sensed imagery, and this dissertation is the only to have applied such a method in this context (very high spatial resolution imagery for the detection of post-hazard damage).

The application of very high spatial resolution co-registered image pairs, containing some apparent mis-registration, to train a CNN for change-object detection is novel. The same applies for the novelty of the application and training of the RNN; previously cited studies relied on georeferenced satellite or fixed-camera image pairs to achieve the change detection results.

3. Timeliness Estimation of Remote Sensing Systems with the RSCM

The RSCM is a model for estimating the timeliness of different configurations of remote sensing systems (Lippitt et al. 2014). The acquisition capacity and transmission capacity components were validated in previous studies, and the analyst capacity component did not have a well-defined methodology for timeliness estimation (Lippitt et al. 2014; Loerch 2016). Chapter 4 is the first study to develop and demonstrate a method for analyst capacity estimation using the RSCM. Additionally, Chapter 4 provides an update to the RSCM acquisition capacity component, to address the time-costs of the complexity of human interactions with automated flights.

4. Software Development for Performing Dissertation Research

The majority of the methods and analytical procedures in this dissertation could not be conducted with existing, off-the-shelf software applications that support automatic pairing of multi-temporal sUAS imagery by location, automated batch co-registration of multi-temporal image pairs, automated evaluation of image co-registration accuracy, methods for providing training data to machine-learning models, the CNN and RNN models and the RSCM capacity estimation tool. I developed substantial amounts and novel types of software for this dissertation to address the lack of readily available options. The software was developed using Python in an object-oriented paradigm, following the principle of model-view-controller application design (Lutz 2013; Veit, on, and 2003 2003). The achieved goals were to develop a suite of software applications that are modular, graphic, and capable of addressing the research questions and repeating the results.

References

Atkinson, P. M., and A. R. L. Tatnall. 1997. “Introduction Neural Networks in Remote Sensing.” *International Journal of Remote Sensing* 18(4):699–709. doi: 10.1080/014311697218700.

Baltsavias, E., K. Cho, F. Remondino, U. Soergel, and H. Wakabayashi. 2013. “Rapidmap – Rapid Mapping and Information Dissemination for Disasters Using Remote Sensing and Geoinformation.” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-7/W2:31–35. doi: 10.5194/isprsarchives-XL-7-W2-31-2013.

Blong, Russell. 2003. “A New Damage Index.” *Natural Hazards* 30(1):1–23. doi: 10.1023/A:1025018822429.

Chen, Liang Chieh, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. Vol. 11211 LNCS.

Cheng, Gong, and Junwei Han. 2016. “A Survey on Object Detection in Optical Remote Sensing Images.” *ISPRS Journal of Photogrammetry and Remote Sensing* 117:11–28. doi: 10.1016/j.isprsjprs.2016.03.014.

Colomina, I., and P. Molina. 2014. “Unmanned Aerial Systems for Photogrammetry and Remote Sensing: A Review.” *ISPRS Journal of Photogrammetry and Remote Sensing* 92:79–97. doi: 10.1016/j.isprsjprs.2014.02.013.

Coulter, L. L., D. a. Stow, and S. Baer. 2003. “A Frame Center Matching Technique for Precise Registration of Multitemporal Airborne Frame Imagery.” *IEEE Transactions on Geoscience and Remote Sensing* 41(11):2436–44. doi: 10.1109/TGRS.2003.819191.

Cutter, Susan L. 2003. “GI Science , Disasters , and Emergency Management.” *Transactions in GIS* 7(4):439–45.

Cvetkovi, Vladimir Miroljub, and Kriminalisticko Policijska Akade. 2013. “Vulnerability of Critical Infrastructure By Natural Disasters.” in *International Scientific Conference, National Critical Infrastructure Protection, Regional Perspective, Book of Abstracts.*

Deng, Chao, Chung Hung John Wang, and Kin Huat Low. 2021. “Preliminary UAS Navigation Performance Analysis in Urban-like Environments.” *AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021*. doi: 10.2514/6.2021-2385.

Dieleman, Sander, Jeffrey de Fauw, and Koray Kavukcuoglu. 2016. “Exploiting Cyclic Symmetry in Convolutional Neural Networks.” *Arxiv* 10.

Dronova, Iryna, Peng Gong, and Lin Wang. 2011. “Object-Based Analysis and Change Detection of Major Wetland Cover Types and Their Classification Uncertainty during the Low Water Period at Poyang Lake, China.” *Remote Sensing of Environment* 115(12):3220–36. doi: 10.1016/j.rse.2011.07.006.

Dumoulin, Vincent, and Francesco Visin. 2016. “A Guide to Convolution Arithmetic for Deep Learning.” *Arxiv* 1–28.

Ehrlich, D., H. D. Guo, K. Molch, J. W. Ma, and M. Pesaresi. 2009. “Identifying Damage Caused by the 2008 Wenchuan Earthquake from VHR Remote Sensing Data.” *International Journal of Digital Earth* 2(4):309–26. doi: 10.1080/17538940902767401.

Gabrlik, P. ;., La Cour-Harbo, A. ;. Kalvodova, P. ;. Zalud, and L. ;. Janata. 2018. "Calibration and Accuracy Assessment in a Direct Georeferencing System for UAS Photogrammetry." *Taylor & Francis* 39(15–16):4931–59. doi: 10.1080/01431161.2018.1434331.

Ganesh, Vidhya R., and H. Ramesh. 2017. "Effectiveness of Contrast Limited Adaptive Histogram Equalization Technique on Multispectral Satellite Imagery." *ACM International Conference Proceeding Series* 234–39. doi: 10.1145/3177404.3177409.

Gerke, Markus, and Norman Kerle. 2011. "Automatic Structural Seismic Damage Assessment with Airborne Oblique Pictometry Imagery." *Photogrammetric Engineering & Remote Sensing* 77(9):885–98. doi: 10.14358/PERS.77.9.885.

Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. "Array Programming with NumPy." *Nature* 585(7825):357–62. doi: 10.1038/S41586-020-2649-2.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. "Mask R-CNN." doi: 10.1109/ICCV.2017.322.

Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9(8):1735–80.

Jensen, John R., and Jungho Im. 2007. “Remote Sensing Change Detection in Urban Environments.” Pp. 7–31 in *Geo-Spatial Technologies in Urban Environments*. Springer.

Joyce, K. E., S. E. Belliss, S. v. Samsonov, S. J. McNeill, and P. J. Glassey. 2009. “A Review of the Status of Satellite Remote Sensing and Image Processing Techniques for Mapping Natural Hazards and Disasters.” *Progress in Physical Geography* 33(2):183–207. doi: 10.1177/0309133309339563.

Joyce, K. E., K. C. Wright, S. v. Samsonov, and V. G. Ambrosia. 2009. “Remote Sensing and the Disaster Management Cycle.” in *Advances in geoscience and remote sensing*, edited by G. Jedlovec. InTech.

Karami, Ebrahim, Siva Prasad, and Mohamed Shehata. 2017. “Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images.” *ArXiv Preprint ArXiv:1710.02726*.

Laben, Craig. 2002. “Integration of Remote Sensing Data and Geographic Information System Technology for Emergency Managers and Their Applications at the Pacific Disaster Center.” *Optical Engineering* 41(9):2129. doi: 10.1117/1.1501137.

Lawrence, Steve, C. Lee Giles, and Sandiway Fong. 2000. “Natural Language Grammatical Inference with Recurrent Neural Networks.” *IEEE Transactions on Knowledge and Data Engineering* 12(1):126–40. doi: 10.1109/69.842255.

Lee, C. Y., S. D. Jones, C. J. Bellman, and L. Buxton. 2008. “Dem Creation of a Snow Covered Surface Using Digital Aerial Photography.” 831–36.

Lippitt, Christopher D., and Douglas A. Stow. 2015. “Remote Sensing Theory and Time-Sensitive Information.” Pp. 1–10 in *Time-Sensitive Remote Sensing*. Springer.

Lippitt, Christopher D., Douglas A. Stow, and Keith C. Clarke. 2014. “On the Nature of Models for Time-Sensitive Remote Sensing.” *International Journal of Remote Sensing* 35(18):6815–41. doi: 10.1080/01431161.2014.965287.

Lippitt, Christopher D., Douglas A. Stow, and Philip J. Riggan. 2016. “Application of the Remote-Sensing Communication Model to a Time-Sensitive Wildfire Remote-Sensing System.” *International Journal of Remote Sensing* 37(14):3272–92.

Lippitt, Christopher D., Douglas Stow, and Lloyd Coulter. 2015. *Time-Sensitive Remote Sensing*. New York: Springer.

Loerch, Andrew. 2016. “Modeling the Timeliness of Airborne Remote Sensing Data.”

Loerch, Andrew C., Gernot Paulus, and Christopher D. Lippitt. 2018. “Volumetric Change Detection with Using Structure from Motion – The Impact of Repeat Station Imaging.” *GI_Forum* (1):135–51. doi: 10.1553/giscience2018_01_s135.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. “Fully Convolutional Networks for Semantic Segmentation.” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June:3431–40. doi: 10.1109/CVPR.2015.7298965.

Lutz, M. 2013. *Learning Python: Powerful Object-Oriented Programming*.

Lyndon, David, Ashnil Kumar, Jinman Kim, Philp H. W. Leong, and Dagang Feng. 2015. “Convolutional Neural Networks for Subfigure Classification.” *Working Notes of {CLEF} 2015*.

Lyu, Haobo, Hui Lu, and Lichao Mou. 2016. “Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection.” *Remote Sensing* 8(6):1–22. doi: 10.3390/rs8060506.

Metternicht, Graciela, Lorenz Hurni, and Radu Gogu. 2005. “Remote Sensing of Landslides: An Analysis of the Potential Contribution to Geo-Spatial Systems for Hazard Assessment in Mountainous Environments.” *Remote Sensing of Environment* 98(2–3):284–303. doi: 10.1016/j.rse.2005.08.004.

Mou, Lichao, Lorenzo Bruzzone, and Xiao Xiang Zhu. 2019. “Learning Spectral-Spatialoral Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery.” *IEEE Transactions on Geoscience and Remote Sensing* 57(2):924–35. doi: 10.1109/TGRS.2018.2863224.

Niethammer, U., M. R. James, S. Rothmund, J. Travelletti, and M. Joswig. 2012. “UAV-Based Remote Sensing of the Super-Sauze Landslide: Evaluation and Results.” *Engineering Geology* 128:2–11. doi: 10.1016/j.enggeo.2011.03.012.

Norouzi, M., DJ Fleet, and Russ Salakhutdinov. 2012. “Hamming Distance Metric Learning.” *Advances in Neural Information Processing Systems* 25.

Nwankpa, Chigozie, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. 2018. “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning.”

Pescaroli, Gianluca, and David Alexander. 2016. “Critical Infrastructure, Panarchies and the Vulnerability Paths of Cascading Disasters.” *Natural Hazards* 82(1):175–92. doi: 10.1007/s11069-016-2186-3.

Pham, Thi-Thanh-Hiên, Philippe Apparicio, Christopher Gomez, Christiane Weber, and Dominique Mathon. 2014. “Towards a Rapid Automatic Detection of Building Damage Using Remote Sensing for Disaster Management.” *Disaster Prevention and Management: An International Journal* 23(1):53–66. doi: 10.1108/DPM-12-2012-0148.

Pinheiro, Pedro O., and Ronan Collobert Com. 2014. “Recurrent Convolutional Neural Networks for Scene Labeling.” in *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32.

Rubinstein-Salzedo, Simon. 2018. *Big O Notation and Algorithm Efficiency*. Cham: Springer.

Shannon, Claude E., and Warren Weaver. 1963. “The Mathematical Theory of Communication. 1949.” *Urbana, IL: University of Illinois Press*.

Sharp, Liz, Adrian Mcdonald, Patrick Sim, Cathy Knamiller, Christine Sefton, and Sam Wong. 2011. “Positivism, Post-Positivism and Domestic Water Demand: Interrelating Science across the Paradigmatic Divide.” *Transactions of the Institute of British Geographers* 36(4):501–15. doi: 10.1111/J.1475-5661.2011.00435.X.

Slama, Chester C., Charles Theurer, and Soren W. Henriksen. 1980. *Manual of Photogrammetry*. American Society of photogrammetry.

Smith, MJ, Jim Chandler, and James Rose. 2009. “High Spatial Resolution Data Acquisition for the Geosciences: Kite Aerial Photography.” *Earth Surface Processes and Landforms* 34:155–61. doi: 10.1002/esp.

Smith, Neil. 1979. “Geography, Science and Post-Positivist Modes of Explanation.” *Progress in Human Geography* 3(3):356–83. doi: 10.1177/030913257900300302.

Storey, Emanuel A., Douglas A. Stow, Lloyd L. Coulter, and Christopher Chen. 2017. “Detecting Shadows in Multi-Temporal Aerial Imagery to Support near-Real-Time Change Detection.” *GIScience \& Remote Sensing* 54(4):453–70.

Stow, D. A. 1999. “Reducing the Effect of Misregistration on Pixel Level Change Detection.” *International Journal of Remote Sensing* 12:2477–83.

Stow, Douglas. 2009. “Geographic Object-Based Image Change Analysis Purpose of GEOBICA.”

Stow, Douglas A., Lloyd C. Coulter, Garrick MacDonald, Christopher D. Lippitt, Richard McCreight, and Nicholas Zamora. 2016. “Evaluation of Geometric Elements of Repeat Station Imaging and Registration.” *Photogrammetric Engineering & Remote Sensing* 82(10):775–88.

Sui, Daniel Z. 1994. “Gis and Urban Studies: Positivism, Post-Positivism, and Beyond.” *Urban Geography* 15(3):258–78. doi: 10.2747/0272-3638.15.3.258.

Tan, HH, and KH Lim. 2019. “Vanishing Gradient Mitigation with Deep Learning Neural Network Optimization.” *7th International Conference on Smart Computing & Communications (ICSCC)*.

Taubenböck, H., and J. Post. 2008. “A Conceptual Vulnerability and Risk Framework as Outline to Identify Capabilities of Remote Sensing.” *Natural Hazards and Earth Systems Science* 8:409–20.

Turner, Darren, Arko Lucieer, and Christopher Watson. 2012. “An Automated Technique for Generating Georectified Mosaics from Ultra-High Resolution Unmanned Aerial Vehicle (UAV) Imagery, Based on Structure from Motion (SfM) Point Clouds.” *Remote Sensing* 4(12):1392–1410. doi: 10.3390/rs4051392.

Veit, M., S. Herrmann-Proceedings of the 2nd international conference on, and undefined 2003. 2003. “Model-View-Controller and Object Teams: A Perfect Match of Paradigms.” *Dl.Acm.Org* 140–49. doi: 10.1145/643603.643618.

Voulodimos, Athanasios, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. 2018. “Deep Learning for Computer Vision: A Brief Review.” *Computational Intelligence and Neuroscience* 2018. doi: 10.1155/2018/7068349.

Westoby, M. J., J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 2012. “‘Structure-from-Motion’ Photogrammetry: A Low-Cost, Effective Tool for Geoscience Applications.” *Geomorphology* 179:300–314. doi: 10.1016/j.geomorph.2012.08.021.

Yu, Y., X. Si, C. Hu, J. Zhang-Neural computation, and undefined 2019. 2019. “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures.” *Direct.Mit.Edu*. doi: 10.1162/neco_a_01199.

Zhang, Su, Christopher D. Lippitt, Cong Chen, Susan M. Bogus, Christopher D. Lippitt, Paul R. H. Neville, Guohui Zhang, Cong Chen, and Vanessa Valentin. 2015. “Extracting Pavement Distress Condition Patterns Based on High Spatial Resolution Multispectral Digital Aerial Photography . Extracting Pavement Surface Distress Conditions Based on High Spatial Resolution Multispectral Digital Aerial Photography.” *Photogrammetric Engineering & Remote Sensing* 81(9):700–720. doi: 10.14358/PERS.81.9.709.

Appendix A

Table 1. Computer processing hardware used for Analyst Capacity timeliness estimates

Hardware Type	Specification	Uses
Central Processing Unit (CPU)	AMD Ryzen 9 3950x – 16 cores, 3500 MHz	• Single-core image co-registration • Multi-core Mask R-CNN, CNN and RNN training and evaluation
Motherboard	Asus ROG Crosshair VIII Extreme	• Data transfers from storage to memory to processor and graphics processing unit
Random Access Memory (RAM)	G.Skill Trident Z Neo – 128 GB (4 x 32 GB) – 2666 MHz	• Data and application storage for rapid access by processor and graphics processing unit
Graphics Processing Unit (GPU)	Gigabyte Nvidia GeForce RTX 3090 – 10,496 CUDA cores, 1755 MHz, 24 GB GDDR6X video RAM	• Machine-learning dataset loading • Multi-core Mask R-CNN, CNN, and RNN training and evaluation
Solid State Disk (SSD)	Samsung SSD EVO 1 TB – 540 MB · s^{-1} data read speed	• Storage for images, labels