Title	Synergistic use of camera targets and GNSS technology for rain-induced landslide detection and response
Sub Title	
Author	ンドゥワイエズ, ガブリエル(Nduwayezu, Gabriel) 神武, 直彦(Kohtake, Naohiko)
Publisher	慶應義塾大学大学院システムデザイン・マネジメント研究科
Publication year	2023
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2023年度システムエンジニアリング学 第356号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40002001-00002023-0009

Master’s Dissertation **2023**

Synergistic Use of Camera Targets and GNSS Technology for Rain-Induced Landslide Detection and Response

Nduwayezu Gabriel

（Student ID Number：82134581）

Supervisor: Professor Naohiko Kohtake

September 2023

Graduate School of System Design and Management,
Keio University
Major in System Design and Management

ABSTRACT

Student Identification Number	82134581	**Name**	Nduwayezu Gabriel

Title: Synergistic Use of Camera Targets and GNSS Technology for Rain-Induced Landslide Detection and Response

Abstract

Heavy rainfall, earthquakes, volcanic eruptions, or human activities like deforestation, mining, or construction cause landslides. Landslides take human lives and destroy homes, roads, bridges, power lines, and water supplies.

Rain-induced landslides frequently strike Rwanda's western region, causing loss of life and property damage.

Why do landslides still surprise people and the cycle repeats yearly during the heavy rain season, despite past events and messages informing the danger of a landslide as a method currently used in Rwanda? There are examples of research and implementation of various landslide detection technologies, such as video surveillance. However, there is still a gap that exists to utilize these technologies in Rwanda because of the algorithms that do not differentiate the noises in the environment such as passing-by objects and flying birds.

Thus, here we show the approach that aims to detect the most landslide in a timely manner and help the victims access the rescue when they need it, just by alerting the concerned parties like the rescue services on behalf of the population.

The approach consists of a camera that monitors the camera targets incorporating Global Navigation Satellite System GNSS receiver, installed along the slope using a camera and object detection and tracking to detect any movement which will indicate an occurrence of a land movement. In the event of a landslide, people become vulnerable and helpless to be able to take any action while fighting for their life. The system will detect the landslide for the rescue services to know when and where a landslide happens. The results have shown a successful detection of the targets and their movement, which is a good step in addressing the shortcomings of similar systems using conventional computer vision algorithms.

With that system in place, the landslides will accurately be detected using the camera targets working in synergy with the GNSS that triggers the camera when there is a movement to maximize the camera's effective use of resources and detection accuracy. As an effect, the system will alert the rescue services on behalf of the population, increasing the chances of the victims being rescued during the golden hour, meaning the first 60 minutes after major trauma.

Keywords (5 words): GNSS, Camera targets, Computer Vision, Objects detection, Object tracking

ACKNOWLEDGMENTS

I want to extend my gratitude to all individuals who have provided their unwavering support during my research endeavor. I thank my supervisor, Professor Naohiko Kohtake, for his invaluable mentorship, unwavering support, and insightful feedback. He has exhibited exceptional qualities as a guide, listener, and source of inspiration in my personal experience. I would like to express my gratitude to the other faculty members of the SDM department, especially Professor Hiroko Inokuma, who served as my second supervisor, for their valuable recommendations and guidance.

I am grateful to the Rwanda Space Agency for allowing me to join JICA-JAXA Network for Space Technology Utilization capacity development. I thank Mr. Hiroshi Shirakawa, Mr. Yuta Hayakawa, Mr. Tatsuji Nishikawa, Mr. Oki Tomoyuki, Ms. Mari Konishi, Mr. Akira Nishimoto and the team from JAXA and JICA for the support during my studies.

The support of my family residing in my place of origin has consistently been provided through their expressions of affection, concern, and supplications. My educational and career aspirations have been supported through their notable acts of sacrifice. This thesis is dedicated to the individuals mentioned.

I am grateful to my classmates, Sensing and Design Lab colleagues, and friends who have provided support and companionship throughout this endeavor. The individuals mentioned above have provided inspiration, assistance, and fulfillment.

I thank Akira Kodaka for the valuable time and guidance he has dedicated to his research endeavors. I would like to thank Bernadette Detera, Akihiko Nishino, Yuki Tomita, and Yu Ohno for their invaluable guidance and mentorship in various facets of mentorship and academic pursuits.

Finally, I would like to express my gratitude to all the individuals I have encountered and established connections with throughout my stay in Japan. The inclusion of culture, hospitality, and friendship has enriched my life. The individuals provided me with a sense of comfort and familiarity during my time spent away from home.

I express profound gratitude to all individuals who have made significant contributions to the accomplishments of my research endeavors thus far.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PICTURES

DEFINITION OF KEY TERMS

Landslide – a geological event caused by gravitational pull that involves the downward and outward movement of slope-forming elements such as rock, soil, artificial fill, or a mix of these. Natural phenomena such as prolonged heavy rainfall, quick snowmelt, earthquakes, volcanic activity, and changes in groundwater levels can all cause landslides. Landslides can cause major property damage, interrupt transportation, and pose serious dangers to human safety due to their rapid pace.

Landslide detection – a process of detecting and monitoring shifts in topography that could lead to or indicate a landslide event.

GNSS (Global Navigation Satellite System) – a satellite system that provides global coverage for locating and navigation.

Camera – an optical device used to capture visual data in the form of images or video recordings. Cameras are made up of a lens that focuses light onto a sensitive medium (film in the past, but digital sensors in modern cameras) to form a picture. They are essential instruments in a wide range of applications, from everyday photography to specialist applications such as surveillance, scientific research, and, as in this study, monitoring natural events.

Camera Target – a specified object or marker on which a camera is programmed to focus or track. Camera targets in the context of landslide detection can be predetermined landmarks or markers placed on a slope. When these objectives are changed or displaced, it can signal landslide activity. Targets improve detection accuracy by providing a stable reference point for the camera system, reducing false alarms from other unrelated movements in the field of view.

Computer Vision – a branch of artificial intelligence and computer science concerned with teaching machines to comprehend and make judgments based on visual data. It aims to imitate and surpass human visual capabilities by allowing machines to comprehend and act on visual inputs from the outside world, such as photos and videos, automatically and intelligently.

Object Detection – a computer vision technique for locating and identifying items in an image or video frame. It not only categorizes things in a scene but also offers spatial positions for them by drawing bounding boxes around them.

Object Tracking – a computer vision wise, tracking refers to the process of tracing an object's route or movement from one frame to the next in a video or series of images. The goal is to keep the object's identification and location consistent across time, especially in the face of problems like occlusion or changes in scale.

OpenCV (Open-Source Computer Vision Library) – a large library of programming functions geared largely at real-time computer vision. In the context of this study, OpenCV is useful for processing visual data, particularly object detection and tracking.

I. INTRODUCTION

I.1 Background

Landslides, as complex and multifaceted natural disasters, continue to pose significant threats to numerous regions around the world. These geological phenomena, triggered by myriad causes – including rainfall, earthquakes, slope instability, human activities, and the overarching influence of climate change – have left an indelible mark on the annals of global disaster history. From the tragic Oso landslide in the United States in 2014 [1] to the devastating mudslides in Sierra Leone in 2017 [2], the global community has witnessed the harrowing impacts of landslides. Between 2000 and 2019, these events resulted in over 32,000 fatalities, affecting an astounding 4.8 million individuals globally (International Disaster Database, 2020).

While landslides have left their mark worldwide, Africa remains a continent particularly vulnerable, given its diverse topography and climatic challenges. Among the nations of Africa, Rwanda, a picturesque yet topographically challenging landlocked country in East Africa, stands out. Often referred to as "The Land of a Thousand Hills," Rwanda's undulating terrains, steep slopes, and high rainfall patterns make it a hotspot for landslide occurrences [3]. Macaulay [4] highlighted that Rwanda grappled with severe landslides caused by heavy rainfall just in May of that year. These events led to the tragic loss of over 130 lives, widespread displacement of its citizens, and significant damage to infrastructure and crops, underscoring the nation's vulnerability.

Picture 1:” A house destroyed by a LANDSLİDE in North of Rwanda, a family of 6 people killed MIDIMAR, 2011” [3]

In recognizing the increasing frequency and intensity of these events, the Rwandan government attributed them to the emerging threat of climate change [4]. This recognition underscores the dire need for effective adaptation and mitigation strategies. But what makes certain regions more susceptible to landslides than others? Topographically, landslide-prone areas are typified by their unstable slopes, which become ticking time bombs, vulnerable to the abrupt and often unpredictable movement of soil, rocks, or debris. These regions, with their steep gradients, fragmented soil structures, and propensity for substantial rainfall or seismic events, become ground zeros for landslide occurrences.

The repercussions of landslides are not limited to the immediate aftermath. The cascading effects, from damaged infrastructure and property loss to the profound psychological impact on survivors, reverberate for years. In the immediate wake of such events, the "golden hour" becomes pivotal. This critical window, the first-hour post-trauma, often becomes the difference between life and death, underscoring the importance of timely rescue intervention.

Given this backdrop, the urgency to proactively monitor and identify potential landslide hotspots cannot be overstated. While the dream of a world completely insulated from the threat of landslides might be distant, advancements in early detection can tilt the scales in our favor. Swift identification of key landslide characteristics – from onset timing and scale to precise location – can revolutionize disaster response, ensuring timely evacuation and mitigation efforts.

In the realm of early detection, many traditional and technologically advanced methods have been explored [5]. Techniques spanning from remote sensing, aerial photography, and ground-based sensing stations offer hope. For instance, the Global Navigation Satellite System (GNSS) has emerged as a beacon in this space, enabling precise gauging of land surface displacement and deformation [6]. Yet, a one-size-fits-all approach remains elusive. Regionals' diverse geographic, economic, and infrastructural landscapes demand bespoke solutions tailored to their unique challenges.

This research aims to explore the mosaic of techniques available for landslide detection and critically evaluate their applicability, effectiveness, and limitations. In doing so, it seeks to weave a tapestry of solutions, each tailored to address the unique challenges nations like Rwanda face, ensuring that we are not found wanting when nature's fury unleashes.

While various methods for early landslide detection have been explored, their effectiveness, applicability, and adaptability to diverse geographical and infrastructural contexts remain an open question. Moreover, with the increasing frequency and severity of landslide events, especially in regions like Rwanda, there is a pressing need to evaluate, refine, and potentially innovate upon existing methodologies.

Picture 2: The mountainous topography of the Karongi District in Rwanda's Western Province (Photo by the author)

Picture 3: A closer look at the slope (Photo by the author)

I.2 Problem statement

Despite the warnings issued and past experiences, landslides surprise people, especially ones who are doing agriculture and living near their farmland.

Currently, no system is in place to detect a landslide unless visually precepted and orally reported. Despite the advancements in landslide detection techniques, there is a gap in understanding their comprehensive effectiveness, especially in regions with unique topographical challenges and limited resources like Rwanda.

I.3 Research Objectives

The objective of this research is to detect the maximum possible number of landslides by detecting land surface changes (like landslide behavior) more accurately and efficiently to be feasible in real world and reporting these occurrences to the rescue services, with efforts directed towards detecting and mitigating their potentially devastating impacts using technological solutions.

I.4 Research Question:

How can existing and emerging landslide detection techniques be effectively tailored and applied to regions like Rwanda, ensuring timely, accurate, and resource-efficient early warning and response?

I.5 Originality

To accomplish the research aims, a unique methodology is suggested that integrates the functionalities of GNSS receivers, integrated within camera targets, and video surveillance for landslide detection. In contrast to conventional approaches that exclusively depend on GNSS receivers or video surveillance, this research leverages the combined advantages of both methodologies, thereby successfully mitigating the limitations associated with their individual use.

The table below presents the main points of originality found in the research.

Table 1: Highlights of the research originality

Problems	**Objectives**	**Approach**
The accuracy of movement detection is constrained, resulting in a source of false warnings.	Utilize alternate methodologies for landslide detection that diverge from the prevailing approaches reliant on video analysis.	The proposed methodology involves employing an objects detection and tracking technique, wherein the predetermined attributes of the camera targets are established.
The existing systems exhibit a deficiency in efficiency, resulting in prolonged operation and suboptimal utilization of resources.	The objective is to identify an effective approach for optimizing resource use in systems operating in sloping terrains with infrequent maintenance requirements.	The activation of the camera will be contingent upon the detection of potential landslide movement, as indicated by the GNSS data. Subsequently, the camera will proceed with the detection process by continuously monitoring the movements of the camera targets. The integration of Global Navigation Satellite Systems (GNSS) and cameras will effectively collaborate to optimize resource utilization, namely in terms of power and bandwidth conservation.
The capacity to adjust and embrace the natural environment.	The objective of this study is to ascertain the optimal locations for deploying the camera target.	The determination of the camera's position, whether it is situated uphill or downhill, as well as the positioning of the Global

		Navigation Satellite System (GNSS), can be inferred by analyzing the shapes of the slope.

The research is undertaken with the intention of addressing the main highlights mentioned in Table 1. The following details are provided to elucidate these highlights and contribute to the improvement of the overall system.

The concept of an integrated system refers to the combination of several components or subsystems into a unified whole, functioning together to achieve by incorporating Global Navigation Satellite System (GNSS) receivers onto camera targets, we are able to acquire accurate location data and utilize this information to facilitate the activation of the camera in response to detecting even minimal movements. The integration of this fusion technique significantly improves the operational effectiveness, precision, and timeliness of the detection system.

This study utilizes sophisticated computer vision methodologies, particularly object detection and tracking, to analyze predetermined targets on slopes, in comparison to the already available algorithms [7] [6] [8] [9]. The utilization of this technique effectively mitigates the possibility of interference caused by extraneous moving entities, hence decreasing superfluous calculations and mitigating the likelihood of erroneous notifications.

The study presents a monitoring system that demonstrates adaptability in two scenarios: when the target travels towards the camera, suggesting a possible receding landslide, and when it moves away from the camera, indicating a potential advancing landslide. The ability to adapt and perform various functions is of utmost importance in achieving a full understanding of the mechanisms and behavior of landslides.

The comprehensive analysis of data involves the thorough investigation of distances, speeds, and their temporal changes, providing a detailed comprehension of landslide dynamics. This level of information is essential for implementing timely interventions.

The adaptability of the system to developing regions is evident due to its design, which allows it to be implemented successfully even in nations with little infrastructure. This characteristic renders it a universally applicable solution.

This project endeavors to expand the limits of landslide detection approaches by creatively integrating GNSS technology with camera surveillance and modern data analytics. The resulting system aims to enhance efficiency, accuracy, and comprehensiveness. Moreover, the applicability of this technology in real-world circumstances, particularly in its ability to enhance a disaster planning and response within rescue organizations, underscores its importance in the field of geotechnical research and expedited search and rescue operations.

I.6 The structure of the thesis

This thesis is divided into the following chapters:

Chapter 1 introduces the topic of this study, its objective, and its significance.

Chapter 2 reviews existing research on landslide detection methods, their strengths and weaknesses, and how they contribute to landslide detection and warning. These past investigations provide a consistent foundation for reasons that support the approach taken in this study.

Chapter 3 introduces a proposed system, its architecture, and methods are all introduced in. The system emphasizes the necessity to overcome the shortcomings of the optical approaches used to identify landslides using the camera in conjunction with the GNSS to maximize resource utilization and ensure the camera's accuracy and feasibility.

Chapter 4 assesses the system's suitability for landslide detection through system verification and system validation. The chapter is based on the prototype and experiment that were carried out to validate the outcomes against the requirements.

Chapter 5 addresses the significance of the experiment outcomes. The author makes recommendations based on what the results show about the camera targets' features and which features are optimal, the deployment of the system based on the nature of the slope, and the contribution of the synergy between the GNSS data and the camera to make the system more feasible in the deployment site.

Chapter 6 finishes the research by comparing the study's outcomes to the research objectives and verifies the hypotheses that led to the research. The author offers a future step for this research in the future.

II. LITERATURE REVIEW

II.1 Landslide detection overview

Landslide detection and monitoring are critical for assessing and managing landslide risk.
“The current availability of advanced remote sensing technologies in the field of landslide analysis allows for rapid and easily updatable data acquisitions, improving the traditional capabilities of detection, mapping and monitoring, as well as optimizing fieldwork and investigating hazardous or inaccessible areas, while granting at the same time the safety of the operators.” [10]

Geotechnical sensors, geodetic surveys, and remote sensing techniques (RSTs) have all been used to detect and monitor landslide displacement and deformation. RSTs are particularly useful, as they could provide continuous geographical and temporal information on the earth’s surface without requiring physical contact with a given location [11]. RSTs are divided into two categories, namely optical and radar. Optical RSTs employ visible or near-infrared light to take photographs of the ground, while radar RSTs use microwave signals to determine the target’s distance and velocity.

Following the introduction of interferometric synthetic aperture radar (InSAR) technology, radar RSTs have become widely used for landslide monitoring and measuring [12]. The InSAR technique uses the phase difference between two or more radar images obtained at various times from the exact location to calculate surface displacement in the line-of-sight direction. It can provide high-precision and large-coverage information on landslide movement with sub-centimeter accuracy and spatial resolution of a few meters. However, radar RSTs have various limitations, including sensitivity to temporal decorrelation (changes in surface properties between acquisitions), spatial decorrelation (changes in baseline length and geometry between acquisitions), atmospheric effects (water vapor and tropospheric delays), and topographic effects (slope and terrain aspect).

“The use of multidisciplinary approaches where InSAR has been deployed for a complete analysis of landslide kinematics and driving mechanisms is still limited and, consequently, the translation of these information in LRMs is even rarer so resulting in the absence of measures sufficient for landslide risk management/reduction.” [13]

To overcome these limitations, some researchers have explored combining optical and radar RSTs with other approaches, such as GNSS, laser scanning (lidar), photogrammetry, or geotechnical sensors. GNSS uses satellite signals to determine the position and velocity of a terrestrial receiver. It can offer accurate, continuous information on the three-dimensional displacement and deformation of a point or an array of points on the terrestrial surface. GNSS has been widely employed for the geodetic monitoring of landslides [14], and the combination of data from GNSS and remote sensing technology has been shown to provide more information for landslide dynamics. However, GNSS also has significant disadvantages (e.g., high costs, limited spatial coverage, signal blocking by vegetation or buildings, and multipath errors).

Laser scanning measures the distance and reflectivity of a target using laser beams. It can be carried out from space, the air, or the ground. According to Xiong et al. [15], laser scanning can provide accurate high-resolution three-dimensional point clouds of the ground surface, which can then be used to create digital elevation models, slope maps, aspect maps, roughness maps, and change detection maps. Laser scanning has been widely employed to classify and investigate landslides. However, it also has numerous disadvantages (e.g., high cost, low temporal frequency, occlusion by plants or structures, and difficulties distinguishing between landslide movement and other surface changes).

Photogrammetry uses photographs captured from various angles to rebuild a scene's three-dimensional geometry. Photogrammetry can be carried out from space, the air, or the ground [16], and it may provide accurate high-resolution three-dimensional point clouds or orthophotos of the ground surface, which can be used similarly to laser scanning. Photogrammetry has been widely used to detect and map landslides. However, it has several limitations (e.g., reliance on image quality, resolution, overlap, orientation, calibration, matching errors, and occlusion by vegetation or buildings).

Geotechnical sensors monitor the physical parameters of soil or rock mass (e.g., pore pressure, water content, strain, stress, tilt, acceleration, and vibration). They can offer real-time data on landslides' interior state and behavior [17], and they are commonly used for landslide research and early detection. However, geotechnical sensors have some drawbacks (e.g., high costs, limited geographical coverage, installation difficulty, maintenance requirements, and data transmission challenges).

As a result of the above stated methods, only some techniques can provide full and reliable information about landslides.

Each technique has advantages and disadvantages that vary depending on the application, the magnitude and type of landslide, the deformation pattern, environmental conditions, and available resources.

The integration and fusion of several methodologies can enhance landslide detection and monitoring accuracy, completeness, and reliability [18]. However, integrating and fusing various data sources can lead to issues related to data compatibility, calibration, registration, resolution, ambiguity, and interpretation. In this regard, the ground-based approach is a better choice for this research because it is accessible anytime and easy to have control over even basic infrastructures found in the developing countries. Among them, there is research on using cameras to detect landslides as discussed below.

II.2 **Use of GNSS to detect landslides.**

According to this paper [6], Shen et al. (2021) proposed a method for processing GNSS landslide observations using a kinematic model that is relatively real-time. The authors conducted a number of tests, including simulation and field studies, to validate the viability of the suggested strategy [6]. The data used in this research comprises noise-affected simulated data and a trend item from the simulation experiment, as well as multiple displacement changes induced manually in the field experiment.

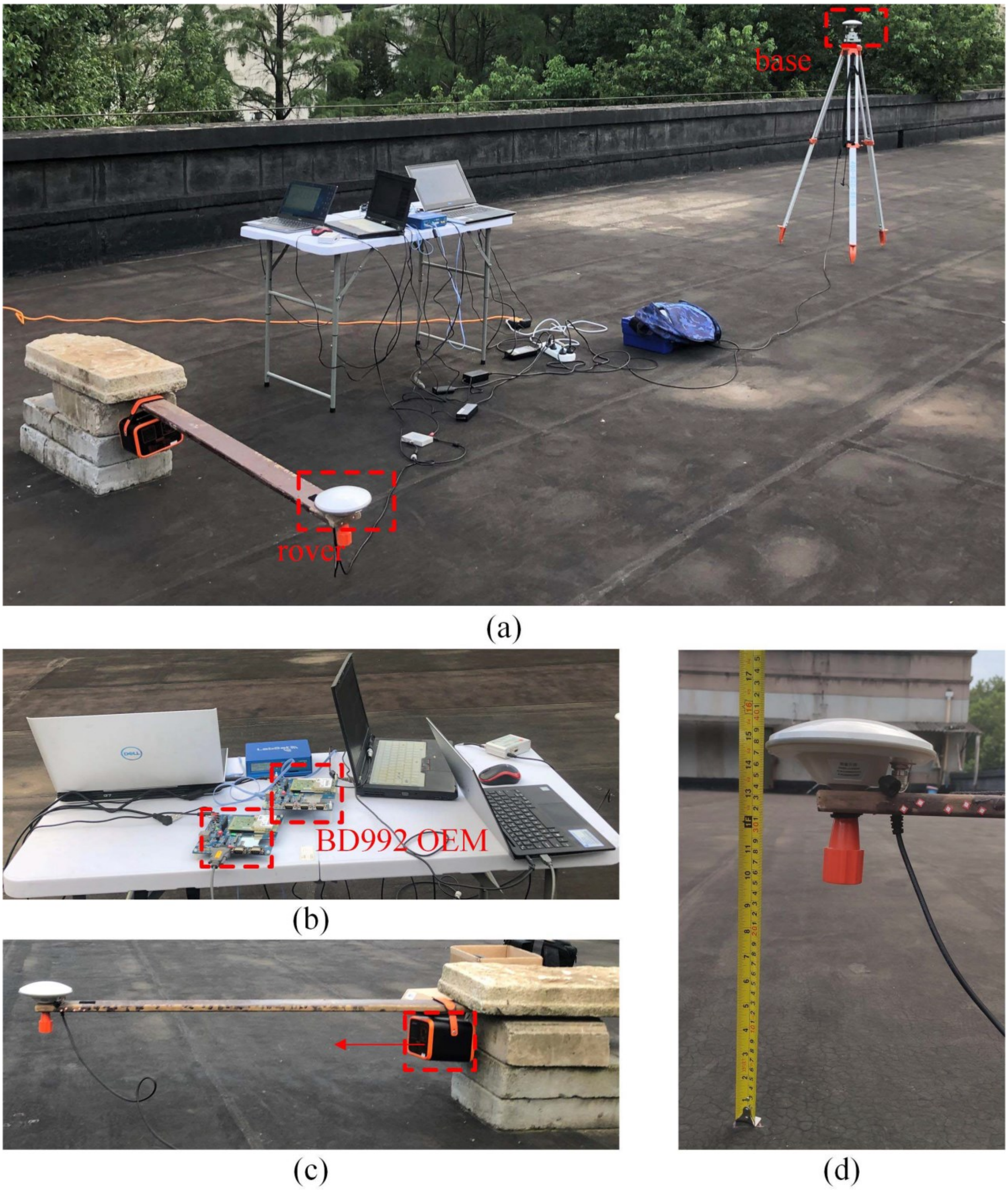


Figure 1:"Field experiment configuration. (a) Overview of the experiment. (b) BD992 OEM boards and data collection computers. (c) Displacement control device. (d) Displacement measurement." [6]

The proposed method is used to process the simulation data. In the field experiment, the relative real-time kinematic model is employed to process the GNSS measurements. Data impacted by noise and trend items are simulated in the simulation experiment. Several displacement adjustments were manually initiated in the field experiment. The results of the experiments suggest that the proposed method for processing GNSS landslide observations is possible.

The approaches employed in this paper are as follows: The concept of short-term landslide monitoring is described, displacement identification based on time series segmentation is proposed, GNSS kinematic positioning is introduced, and a workflow for displacement detection by time series segmentation of GPS real-time kinematic positioning is provided.

The Landslide has different stages, and this research gives an idea of the stage at which our system is suitable. “In the accelerated deformation stage, the deformation rate increases continuously until the landslide occurs. Studies show that the accelerated deformation stage of the slope is the basis and prerequisite for the occurrence of landslides” [6] as shown in the figure below.

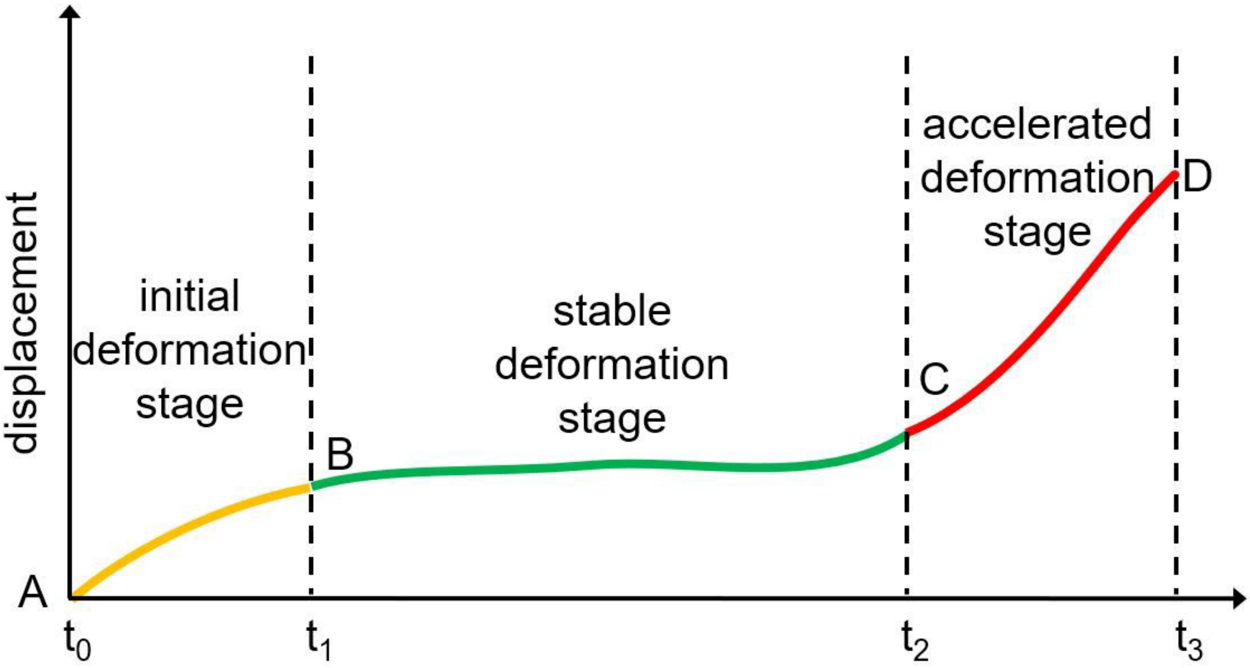


Figure 2:" Ideal diagram of slope body displacement [19]" [6]

This paper makes three contributions: it proposes a method for processing GNSS observations using a relative real-time kinematic model, it conducts simulation and field experiments to verify the feasibility of the proposed method, and it demonstrates that the proposed method is feasible for processing GNSS landslide observations.

The limitation is that the current method of displacement detection and extraction is separate and complex, which may introduce additional complexity and potential errors in the process, reducing the use in detecting small displacements.

II.3 **Use of camera to detect landslides.**

II.3.1 ***Landslide occurrence prediction using optical flow.***

This paper proposes a method of predicting the occurrence of landslides by taking a video of the slope and using optical flow measurement to obtain the slope's sliding velocity. The authors used the spatiotemporal differentiation method (an optical flow measurement method suitable for measuring small displacements). The paper also describes the method of smoothing multiple images. The authors carried out two types of experiment: a simulation and measuring the displacement of the slope with images of actual forests [7].

The contributions of this paper are proposing a system for predicting the occurrence of landslides utilizing a fixed-point camera and monitoring a diverse range of forest and soil slopes and terrains by processing videos and photos, optical flow measurement is used to examine the motion of things. It also helps to reduce the impact of weather conditions and investigates the possibilities of anticipating landslides.

Another contribution is detecting displacement when a landslide begins and reducing slope displacement to zero when a landslide does not occur and detecting a slope movement of 1 mm each day by monitoring an area of around 100x100 square meters.

The paper author acknowledges that weather conditions, such as wind and rain, can affect the accuracy of the velocity vectors and increase the difficulty of detecting landslides. [7]

II.3.2 ***Landslide Monitoring System Implementing IOT Using Video Camera***

This paper proposes a landslide monitoring system based on Raspberry Pi and implementing IOT using a video camera [9]. The proposed system primarily functions by receiving input through a digital video camera in the form of a continuous video stream. The system employs the OpenCV library, a renowned tool in computer vision applications to process these videos, especially when operating on the Raspberry Pi.

One of the critical processes involved in landslide detection is background subtraction. The system utilizes a Gaussian-mixture-based background/foreground segmentation algorithm to achieve this. This algorithm assists in discerning potential landslides in the ongoing video stream. Once this segmentation is completed, the system calculates the centroid of the resultant frame, leveraging image moments for this purpose.

Furthermore, the system incorporates median filtering to ensure the video stream's clarity and mitigate any potential noise. This method is specially chosen for its non-linear noise removal capability, which allows it to preserve the critical edges in the video, ensuring a precise analysis. Figure 3 below shows the system review of the system.

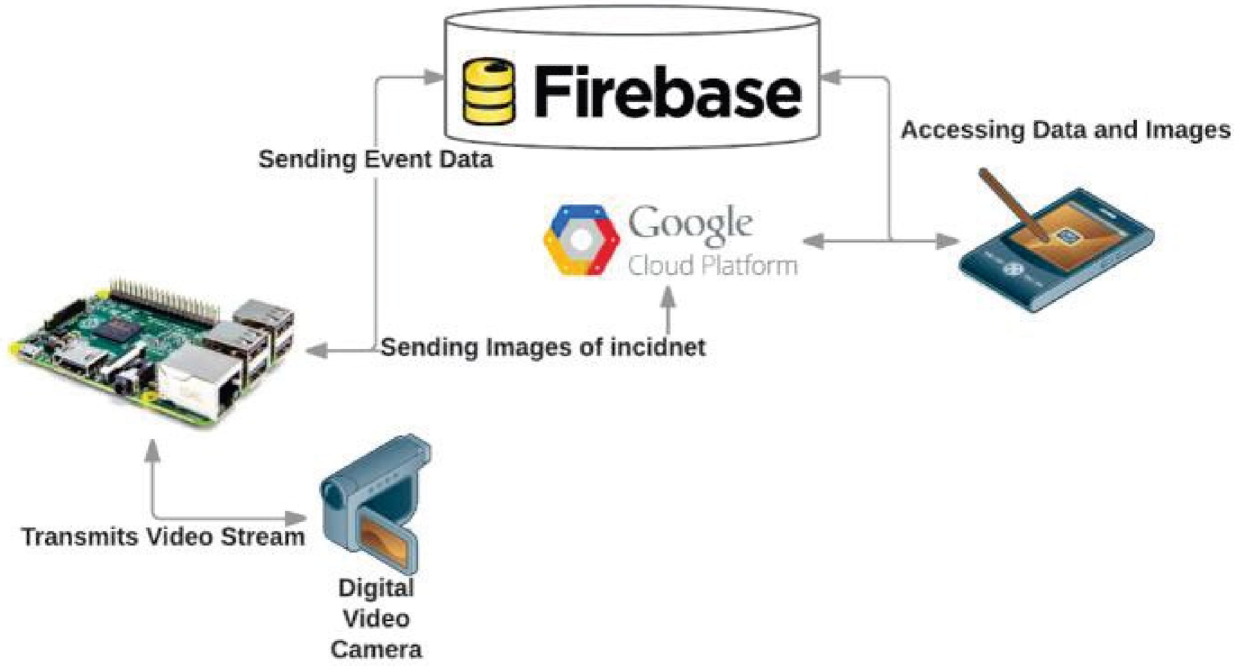


Figure 3: "System overview" of a Landslide Monitoring System Implementing IOT Using Video Camera [9]

The system was tested on a waterfall video approximated to a layered landslide. The following results were obtained:

- The system was able to detect the motion of the waterfall and classify it as a landslide.
- The system was able to calculate the centroid of the frame obtained from the background subtraction and noise removal process.
- The system was able to notify stakeholders of landslide occurrences via an Android app.
- The system achieved a detection accuracy of 95% and a false positive rate of 5%.
- The system could process the video stream at a rate of 10 frames per second on the Raspberry Pi.
- The system operated on a low-cost device with low power demands, making it suitable for installation in remote regions.

This paper made the following contributions:

The paper introduces a novel, low-cost landslide monitoring system that leverages the capabilities of Raspberry Pi. This system uses a video camera to integrate the Internet of Things

(IoT). The system conducts a real-time analysis of the designated area by tapping into the video stream. To detect potential landslides, it employs computer vision algorithms. Once a landslide is identified, the system promptly sends alerts to stakeholders through a dedicated Android app. The paper also offers an in-depth look into the system architecture, outlining the hardware and software components integral to its operation. A significant portion of the paper is dedicated to explaining the computer vision algorithms employed, notably motion detection and centroid calculation. These techniques play a crucial role in detecting landslides as they occur.

To underscore the effectiveness of this proposed system, the paper presents experimental results. These results showcase the system's capabilities in detecting landslides within a simulated environment. However, recognizing the potential for improvement, the paper also suggests avenues for future research. This includes the possible integration of deep learning techniques, incorporating additional sensors, and developing predictive models to further refine the accuracy of landslide detection.

II.3.3 ***Development of landslide detection surveillance***

This paper proposes a landslide detection surveillance system that uses a network of cameras to continuously monitor an area for precursor events to a landslide [8].

The methodology involves the following steps:

- A network of cameras is installed in the field.
- The live input video is stitched and then processed frame by frame.
- Preprocessing operations are performed to denoise image frames.
- The algorithm justifies the detected changes as a precursor event by monitoring its vertical displacement using the Lucas-Kanade method to determine optimal flow.
- A sample video of falling rock is recorded to determine the threshold pixel length, equal to 1 m of actual fall.

The proposed system requires inexpensive cameras with low frame rates and less memory and bandwidth, making it an effective landslide warning system in terms of use of the power resources.

The paper does not provide detailed results regarding the proposed system’s performance. However, the authors state they implemented the proposed algorithm in Python using the OpenCV library and tested it on five sample videos. They used a smaller window for generating motion vectors using the Lucas-Kanade optical flow algorithm, which was able to track the smallest precursor event (i.e., the smallest falling rock) smoothly. The authors also mention that adopting FOSS (Free and Open-Source Software) decreases the cost of the system; lower costs, in turn, increase user acceptability.

The contribution of this paper is to propose a cost-effective real-time surveillance system to detect landslide precursor events. The system uses a network of low-cost cameras and an image processing algorithm to continuously monitor a given area for any such events. It issues an alert as soon as any such event is detected. The proposed system requires low memory and bandwidth, making it an effective landslide warning system.

Having examined the extensive body of literature surrounding landslide detection systems, it becomes evident that while numerous methods have been proposed, certain gaps and challenges remain unaddressed. The existing systems offer a foundational understanding of the processes and technologies available. Building upon this foundation and aiming to address some of the identified limitations, we introduce our proposed system that uses GNSS and camera targets in synergy to detect landslides. This novel approach integrates advancements in algorithm design, promising a more efficient and accurate means of monitoring potential landslides by improving the optical methods to detect landslides.

III. PROPOSED SYSTEM

This study introduces a complete landslide detection system in response to the rising threat of landslides aggravated by climate change, particularly in locations such as western Rwanda. The suggested system uses computer vision techniques via OpenCV by combining the precision of GNSS receivers which are installed into the camera target with the real-time monitoring capabilities of camera surveillance. This integration ensures that predetermined camera targets placed on slopes, which serve as indicators of probable landslide activity, are accurately detected. The technology not only improves the accuracy and reliability of landslide detection, but it also supports timely responses, coinciding with global efforts such as the United Nations Sustainable Development Goal 13 to counteract the effects of climate change and protect populations.

In addition to GNSS, video surveillance will be used to monitor and detect a landslide on a slope where a camera target that embeds GNSS receiver is installed physically. Combining these technologies is essential to reduce the possibility of not detecting a landslide, which could occur when GNSS receivers alone are used, and increase the system efficiency as the GNSS data will be used to trigger the camera only then there is a slight movement detected in the camera target.

The system was built for experiments and verification against the system requirements. The results will be discussed with rescue professionals in Rwanda and/or Japan to validate their usefulness in improving rescue operations and response time.

In earlier research, many advancements have been made using different algorithms and methods, including the visual background extractor, optical flow algorithms, and the Harris corner detector. However, these methods suffer from the same drawback— "none of them assure if the visual change/optical motion caused is corresponding to a precursor event of landslide. The changes could be due to noise (e.g., from moving trees or flying birds). Also, unnecessary computations are performed in monitoring points that are not of concern. Our proposed algorithm does not have these drawbacks and outperforms the proposed earlier algorithms." [8]

The proposed system applies the predefined features of the targets to object detection algorithms to eliminate the unnecessary computations that characterize existing systems. This addition is necessary to ensure the system is precise and to minimize false alerts, as targets are set, and other types of interference will be ignored.

The system was called Heron System with the reason being that Herons are birds with an acute sense of observation. They remain motionless for lengthy periods of time before swiftly capturing their victim in water. Similarly, the Heron system continuously monitors and examines slopes for indicators of landslides, ready to send an alert when certain criteria are met.

III.1 Concept of operation

The present study proposes a novel system for landslide detection, which involves using color-coded targets embedding the GNSS receiver positioned on a slope.

- The targets have been intentionally designed with varying shapes and colors to enhance their detectability and recognizability by a camera system.
- The camera intermittently records images of the incline and transmits them to a processing unit.
- The processing unit uses a shape-based approach to identify targets and determine their positions within the images. Furthermore, it assigns three-dimensional coordinates to these targets based on both their color and shape characteristics.
- The processing unit conducts a comparative analysis of the targets' coordinates across a given period, thereby determining their displacement and velocity.
- If the displacement or velocity of a given target surpasses a predetermined threshold, the system will activate an alarm and inform rescue services that a landslide has occurred.

The scope of this research is the reduction of the probability of detection of a landslide due to a limited number of GNSS in case the GNSS method is used alone.

The items listed in the figure below are derived from the Functional Flow Block Diagram components shown in Figure 4.

- ✓ GNSS Receiver: This component represents the GNSS monitoring component.
- ✓ Camera: Represents the camera that is activated when a trigger is received.
- ✓ Computer Vision & Data Processor: This component represents the system's data processing.
- ✓ Alert System: This is the communication module in charge of sending out alerts.

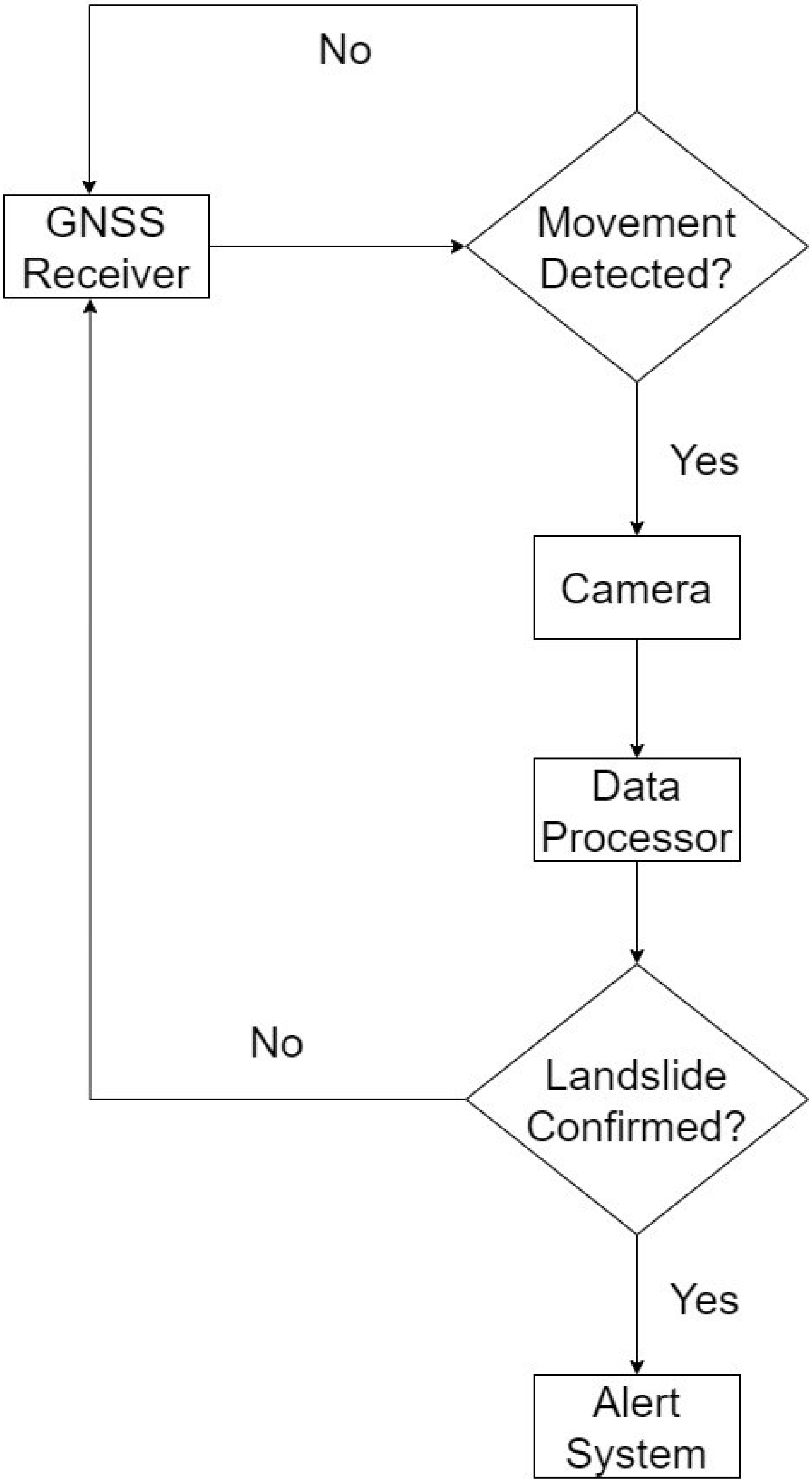


Figure 4:Simplified Data Flow Diagram for Concept of Operations

The CONOPS explains in detail how the Landslide Detection System works during a landslide scenario. The primary goal is to detect potential landslides as soon as possible and to distribute alerts as quickly as possible in order to mitigate risks and ensure community safety.

III.2 System Stakeholders

Stakeholder analysis is an important aspect of this system since it ensures that all parties affected by or influencing the system's outcome are identified, understood, and catered to. Heron system stakeholder analysis is carried out by:

III.2.1 *Identifying Stakeholders:*

Individuals, groups, or institutions who may affect or be affected system outcomes either directly or indirectly are the stakeholders.

- ✓ Local Communities: People who live in landslide-prone locations.
- ✓ Government: Bodies of local, regional, and national government in charge of disaster management and infrastructure.
- ✓ Rescue Services: First responders such as paramedics, firefighters, and police.
- ✓ Research Institutions: Academic or research organizations that investigate landslides or other natural disasters.
- ✓ Environmental Groups: Organizations interested in landslide environmental implications and preventive techniques.
- ✓ Funding Bodies: Organizations or individuals who contribute to the project's funding.
- ✓ Technology Suppliers: Vendors or suppliers of system equipment and software.
- ✓ Project Team: Individuals who are directly involved in the system's design, development, and deployment.

III.2.2 *Stakeholder Interest and Influence:*

Table 2: System stakeholders list

Stakeholder	**Interest**	**Influence**	**Concerns**
Local Communities	High	Medium	Safety, timely rescue, access to the golden hour
Government	High	High	Public safety, infrastructure protection, cost-efficiency.

Rescue Services	High	High	Accurate and timely alerts, accessibility of affected areas.
Research Institutions	Medium	Medium	Data accuracy, research opportunities.
Environmental Groups	Medium	Medium	Ecological impact, prevention methods.
Funding Bodies	Medium	High	ROI, project success, public image.
Technology Suppliers	Medium	Medium	Profit, technical requirements.
Project Team	High	High	Project success, technical feasibility.

III.3 Requirement analysis

A use case diagram depicts how system users (or "actors") interact with a system, and it aids in identifying and defining the system's main functionalities from the user's point of view. According to the use case diagram below in figure 5, the rescue services are the primary users of the system.

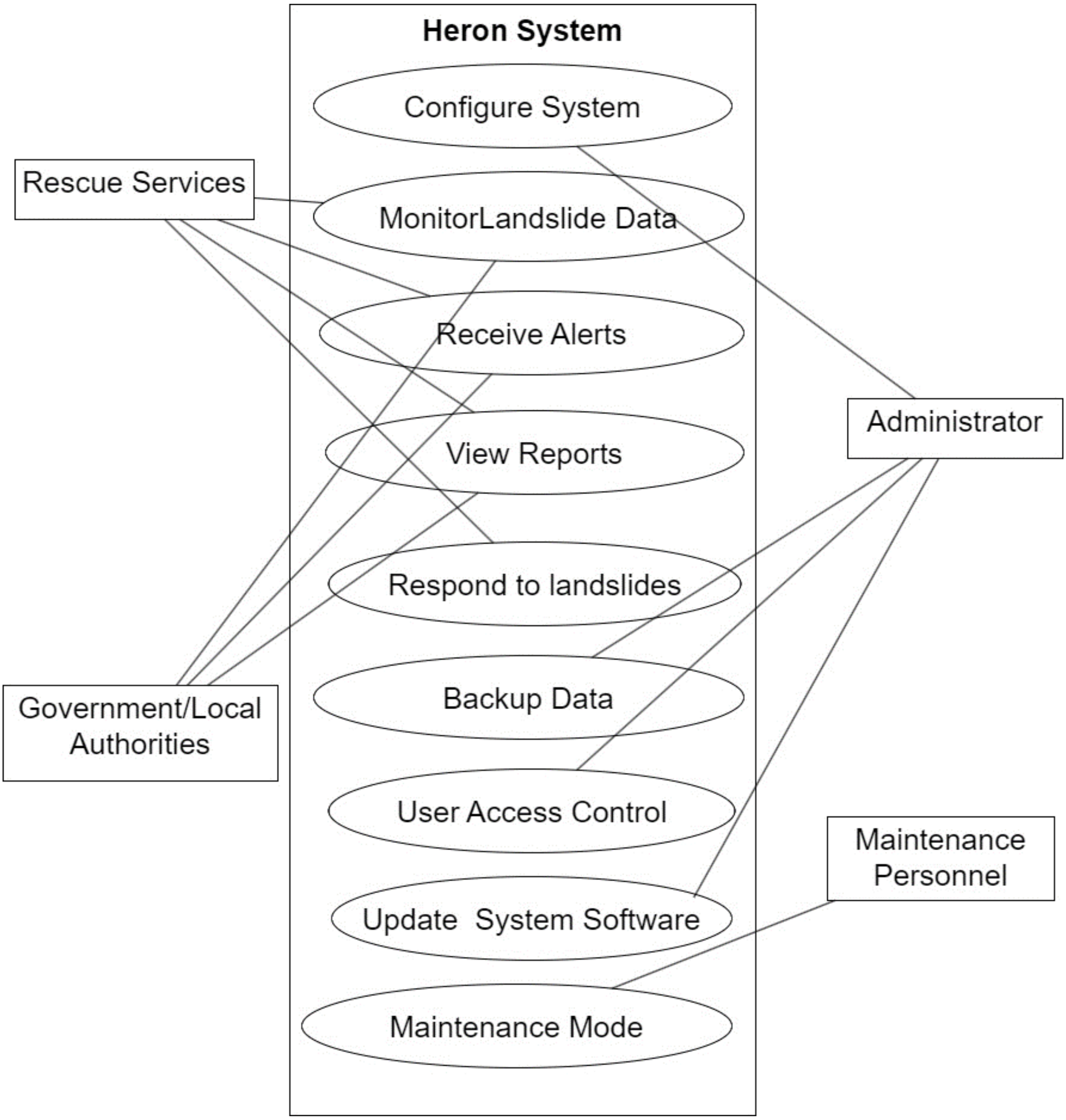


Figure 5: System Use Case Diagram

III.4 System Requirements

The following requirements outlined below serve as the fundamental basis for the design, development, and deployment of the system. Ensuring the appropriate fulfillment of these requirements is crucial for optimizing the operational efficiency and effectiveness of the system within the landslide detection context.

III.4.1 *Functional Requirements:*

Data Acquisition:

- ✓ The system shall continuously acquire data from the GNSS receiver to monitor terrain movement.
- ✓ The system shall activate the camera to capture visual data when slight movement is detected by the GNSS.

Data Processing:

- ✓ The system shall process GNSS data in real-time to detect any movement or changes in the terrain.
- ✓ The system shall process visual data from the camera to identify possible landslides or other significant events.

Data Transmission:

- ✓ The system shall transmit processed data to a central monitoring station at regular intervals or when significant events are detected.

Alert Mechanism:

- ✓ Upon detecting potential landslide activity, the system shall activate an alert mechanism, sending notifications to designated stakeholders or authorities.

Data Storage:

- ✓ The central monitoring station shall store received data for future analysis and reference.

System Control:

- ✓ Users shall be able to control system parameters, establish thresholds, and manage communication preferences via a user interface.

III.4.2 *Non-Functional Requirements:*

Reliability:

- ✓ The system shall have a minimum uptime of 99.5% and should be able to recover quickly from any failures.

Accuracy:

- ✓ The GNSS receiver shall detect terrain movements with an accuracy of at least 95%.
- ✓ The camera shall measure the distance and the velocity of the targets with the accuracy of at least 95%

Durability:

- ✓ The system components shall be weatherproof and durable, capable of withstanding the environmental conditions of the deployment area.

Scalability:

- ✓ The system shall be scalable to accommodate additional sensors or cameras if required.

Security:

- ✓ Data transmission shall be encrypted to ensure confidentiality and integrity.
- ✓ The system shall have mechanisms to prevent unauthorized access or tampering.

Power Efficiency:

- ✓ The system shall operate efficiently to maximize the duration of operation on battery or solar power.

Maintainability:

- ✓ The system shall be designed in a modular fashion, allowing easy maintenance and component replacements.

User Interface:

- ✓ The user interface at the central monitoring station shall be intuitive and user-friendly, allowing easy system configuration and data visualization.

Response Time:

- ✓ The system shall have a maximum response time of 5 seconds from the moment a potential landslide activity is detected to the activation of the alert mechanism.

Compatibility:

- ✓ The system's software components shall be compatible with standard operating systems and should allow integration with other monitoring systems if needed.

The aforementioned requirements list is delineated in the subsequent table.

Table 3: System Requirements

Requirement Type	**Requirement ID**	**Requirement Description**
Functional	FR1	Continuously acquire data from the GNSS receiver.
	FR2	Activate the camera to capture visual data when slight movement is detected by the GNSS.
	FR3	Process GNSS data in real-time.
	FR4	Process visual data to identify possible landslides.
	FR5	Transmit processed data to a central monitoring station regularly or upon significant event detection.
	FR6	Activate an alert mechanism upon potential landslide detection.
	FR7	Store received data at the central monitoring station.
	FR8	Control system parameters, establish thresholds, and manage communication via user interface.
Non-Functional	NFR1	System uptime of 99.5% with quick recovery from failures.
	NFR2	GNSS detection accuracy of at least 95%.
	NFR3	Ensure camera measurement of the distance and the velocity with the accuracy of at least 95%
	NFR4	Weatherproof and durable system components.
	NFR5	Scalability to accommodate additional sensors or cameras.
	NFR6	Encrypted data transmission for security.

	NFR7	Efficient power consumption for maximized battery or solar power operation.
	NFR8	Modular system design for easy maintenance.
	NFR9	Intuitive and user-friendly user interface.
	NFR10	Maximum system response time of 5 seconds for alert mechanism activation.
	NFR11	Software compatibility with standard operating systems and potential for integration with other systems.

III.5 System Architecture

Our system's architectural blueprint comprises distinct layers, each contributing to the holistic functioning in figure 6 below:

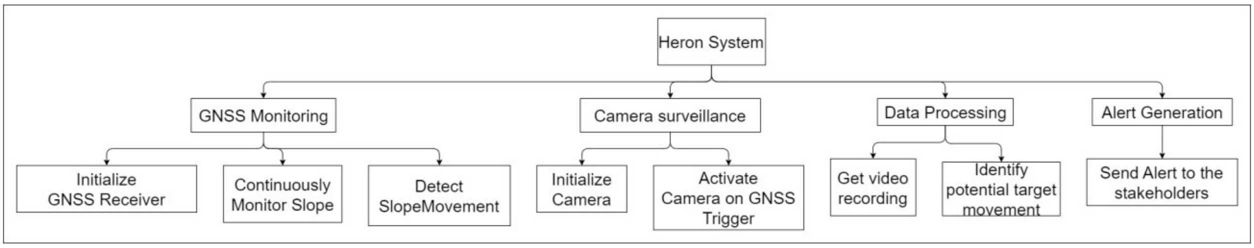


Figure 6:System Functional Flow Block Diagram

Heron system conceptual model that defines a system’s structure, behavior, and interactions. The figure 7 depicts the physical design and the subsystems here called system layers.

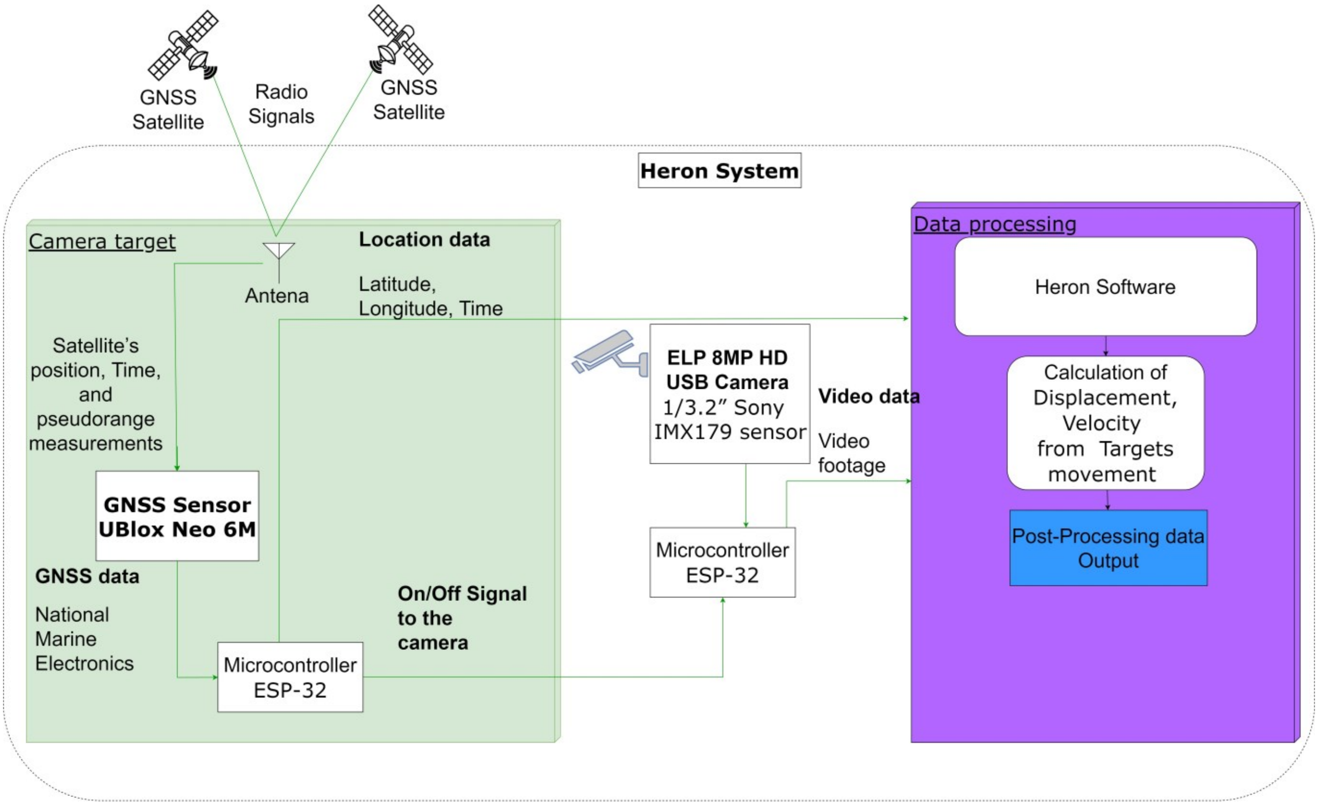


Figure 7: Physical Design of Heron System

a. The sensor layer is the system's first layer.
    - A UVC-compatible ELP USB camera is used to acquire visual data on the slope.
    - A GNSS sensor, specifically the Ublox Neo-6M model, is seamlessly integrated with an external GPS antenna to acquire precise positioning data effectively.

b. The microcontroller layer comprises the ESP32-S3 microcontroller. It is equipped with wireless communication capabilities and serves as the system's pivotal control unit. The system establishes an interface with the camera and the GNSS sensor to retrieve data from both.

The microcontroller establishes connections with the camera and GNSS sensor using suitable interfaces (e.g., USB for the camera and serial communication for the GNSS sensor). These interfaces facilitate the reception of data from the sensors.

c. The data acquisition and preprocessing layer is responsible for the collection and initial processing of data in the system. It is designed to gather data from sensors efficiently and prepare the data for further analysis.

The ESP32-S3 microcontroller acquires data from both the camera and GNSS sensor simultaneously.

Data preprocessing is a crucial step in the data acquisition process, where the microcontroller undertakes various tasks to prepare the data for further analysis. These tasks include buffering, formatting, and synchronization, which are essential to ensure data integrity and compatibility.

d. Wireless Communication Layer:

Wi-Fi Connectivity: The ESP32-S3 microcontroller uses its wireless communication capabilities, specifically Wi-Fi, to establish a reliable internet connection.

Data transmission involves the microcontroller securely sending the gathered data (including images and GNSS position data) to a web-based processing application. The transmission is conducted using robust data transmission protocols (e.g., HTTP/HTTPS), ensuring the integrity and confidentiality of the transmitted information.

e. Processing application.

The processing application effectively acquires the transmitted data from the microcontroller. Subsequently, the application undertakes data processing activities, including image analysis, target detection, target localization, displacement and velocity calculation, and landslide detection. These activities are undertaken according to the system's established algorithms and logical framework.

Data storage involves the retention of processed data (the identified landslides, their attributes, and other relevant details) in a designated database or persistent storage medium. This enables subsequent analysis and reporting.

f. User Interface Layer:

Web-Based User Interface: The web-based processing application facilitates a user interface that can be accessed through a web browser. The system allows the retrieval of both real-time and historical data. Additionally, users can visualize identified landslides, analyze displacement patterns, and modify system configuration settings.

The system's configuration and control aspect entails providing a user interface that enables modifying system parameters, establishing thresholds, and governing communication preferences based on user specifications.

The primary users of this system would be rescuing services, with the potential involvement of other stakeholders such as paramedical services and local authorities.

III.6 **Methodology**

The research methodology is meticulously structured to enable an accurate and timely landslide detection mechanism. It is segmented into four primary components:

- Site selection and field survey
- GNSS data collection and processing
- Camera activation and video surveillance recordings
- Data integration and analysis

Each component has been tailored to serve the overarching aim of employing object detection and tracking for effective landslide detection. The methodology on figure 8 minimizes potential interference from unrelated moving entities by centering the analysis on predefined targets on slopes, thereby reducing computational demands and false alert risks.

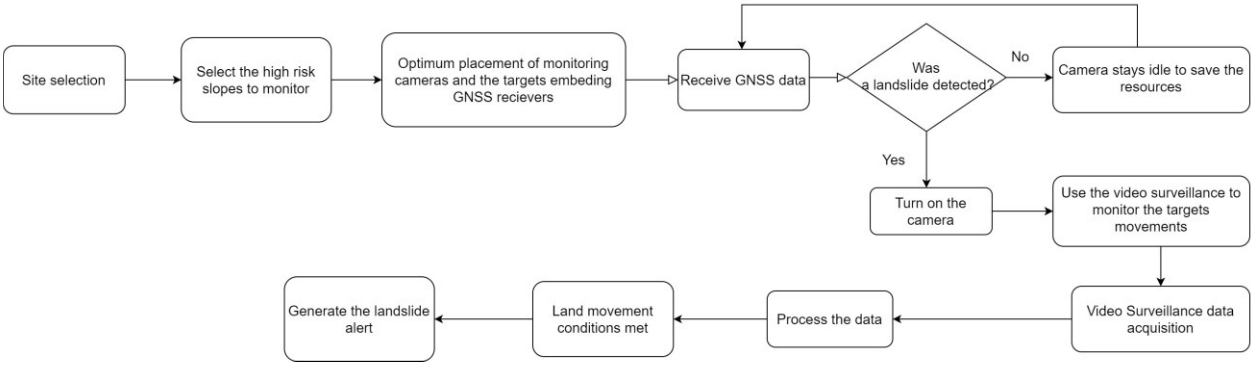


Figure 8: Methodology

III.6.1 ***Site Selection and Field Survey***

The initial step revolves around pinpointing an ideal site. Given the western region of Rwanda's predisposition to landslides, it emerges as a primary consideration. Several criteria govern the site selection:

- Unobstructed Sky View: Vital for the GNSS receivers to optimally capture satellite signals.
- Clear Line of Sight to the Slope: Ensuring that the camera has an unhampered view for surveillance.
- Accessibility: Simplifying installation, maintenance, and regular monitoring.
- Security: The equipment and setup need protection from potential theft and vandalism.

After selection, a comprehensive field survey follows, focusing on geomorphology, geology, hydrology, vegetation, land use, and previous landslide occurrences. The survey will employ tools like GPS devices, compasses, clinometers, cameras, and field notebooks to meticulously document the area.

III.6.2 *GNSS Data Collection and Processing*

Once the site is finalized, the GNSS data collection begins. The Ublox Neo-6M GPS module receivers/or RTK-Enabled GNSS receivers, capable of tracking GPS, will be strategically installed at the site, and we precise that they will be embedded into the camera targets. The installation will be guided by the optimum placement compared to the position of the camera. These will be connected to essential peripherals like antennas, batteries, solar panels, and the Espressif ESP32-S3-DevkitC-1-N8R2 development board.

The crux of this phase is the processing of GNSS data. Differential positioning techniques will be applied to derive precise displacement measurements. The unique algorithm developed for this study processes this data and acts as a trigger for the camera system based on slight movements detected by the GNSS receiver embedded in the camera target.

III.6.3 *Camera Activation and Video Surveillance Recordings*

Upon detection of significant movement by the GNSS receiver embedded in the camera target, the camera system is activated. This ensures that the camera only records pertinent events, optimizing storage and power usage. The surveillance primarily focuses on the predefined targets on the slopes, ensuring that the recorded footage is relevant and accurate.

III.6.4 *Data Integration and Analysis*

After capturing both GNSS data and video recordings, the integration phase begins. By combining these datasets, the system can corroborate landslide occurrences, significantly enhancing detection accuracy. Advanced algorithms will analyze this integrated data, spotlighting patterns, movements, and potential landslide risks. The insights derived from this analysis can be instantly relayed to relevant stakeholders, facilitating swift responses and possibly preventing disasters.

By synergizing the strengths of GNSS receivers and camera surveillance activated by the GNSS-triggered mechanism, the research methodology ensures a multi-faceted and robust approach to landslide detection.

IV. SYSTEM EVALUATION

It is critical to evaluate the performance and effectiveness of the Heron system in order to assure its dependability and accuracy. Here are evaluation methods that we were able to conduct:

A. Indoor tests:

Create controlled target movements in an indoor situation and see if the system identifies them accurately.

Simulate diverse environmental conditions for foreign objects in the camera's field of vision and lighting change by modifying the camera aperture.

B. Testing in the Field:

Interaction with environment: Test the visibility in different slope shapes and distance to evaluate the features of the camera targets embedding GNSS receiver and the position of both them and the camera**.**

Controlled Landslides: simulate a landslide by moving a camera target in a controlled area to evaluate the system's real-world response.

Accuracy Evaluation:

Calculate the number of times the system incorrectly reported a landslide when none occurred.

Environmental and Power usage Analysis: Examine the system's power usage, especially for a system that is designed to be used in remote locations. For this reason, we test the usefulness of the synergy between GNSS data and camera data in controlling when the camera operates or not to maximize the effective use of power mainly and other resources like bandwidth.

We can ensure a full review of the system's capabilities, strengths, and areas that may need modification or enhancement by combining the above various evaluation methodologies.

We will cover the results of the above methods in the experiment section. But before, we need a prototype to use for the experiments and produce the results.

IV.1 Prototyping

The goal of developing a prototype based on the hardware and software described below is to create a functional sensing hardware consisting of a camera for slope monitoring that detects predefined, stationary, and clearly visible camera targets embedding DNSS receiver in the camera's field of view. The target's movement with the slope indicates potential landslide events. The prototype is aimed at demonstrating the hardware components' ability to capture data, monitor target slope conditions, and provide real-time or near-real-time information for warning and analysis.

IV.1.1 *Hardware*

The hardware shown in Figure 6 is part of the Heron system as the hardware sensing station. The components of the Heron device are described below:

a. Camera:
   - 8 megapixel high-definition camera modules with variable aperture and focal length range of 5–50 mm.
   - Functions: Image capture and video recording are supported for the visual monitoring of the targets deployed on the slope.
   - Connection: The ESP32 microcontroller interfaced with the camera via native USB support (General-Purpose Input/Output GPIO Pin 20 for D- data line and GPIO Pin 19 for D+ data line).
   - As it is Universal Serial Bus (USB) capable, the camera can also be interfaced directly to a Personal Computer (PC)
   - It is powered by a 5V power supply.

b. GNSS Receiver1:
   - Type: Standard-precision GNSS (GPS) receiver module (Ublox Neo-6M).
   - It is powered by a 3.3V power supply.
   - Functions: Provides positioning, velocity, and timing data to monitor slope movement.
   - Connection: Via UART interface with the ESP32 microcontroller (GPIO Pin 35 as RX and GPIO Pin 36 as TX).

c. GNSS Receiver2:
   - RTK-Enabled ZED-F9P.
   - 5V powered via USB.
   - Interfaced to a Personal Computer (PC) via USB.

- We log the GNSS data from Ublox (manufacturer of the receiver) U-Center software for analysis.

d. GNSS Receiver3:

- RTK-Enabled C94-M8P-4.
- 5V powered via USB.
- Interfaced to a Personal Computer (PC) via USB.
- We log the GNSS data from Ublox (manufacturer of the receiver) U-Center software for analysis.

e. Power Supply:

- Power supply type: Direct current (DC) with dual rechargeable 18650 backup batteries installed in parallel.
- Voltage requirements: 3.3V and 5V voltage levels are required to power all components.
- Connection: Used to supply power to the ESP32 microcontroller and other components via power lines.

f. Global Positioning System Antenna/GNSS:

- GPS receiver module’s active GPS antenna.
- Functions: Receives satellite signals to pinpoint the location of the sensing station and conveys them to the GPS receiver through the cable.
- Connection: A coaxial cable with an IPS-SMA Conversion Cable connects the antenna to the GNSS receiver module.
- Powered by a 3–5V power supply.
- Frequency: 1575 MHz

g. Microcontroller ESP32:

- Development board type: ESP32 (Espressif ESP32-S3-DevkitC-1-N8R2 development board).
- Specifications: Strong microcontroller with Wi-Fi and Bluetooth capabilities.
- Connections:
    - Camera: Connected to the camera module for image and video capture.
    - GNSS receiver: This module communicates with the GNSS receiver module to access precise positioning data.
    - Power supply: Connected to the power supply to receive power.

- Communication interfaces: The ESP32 microcontroller includes Wi-Fi, Bluetooth, and UART interfaces for data transmission to external devices or cloud services. This allows for real-time monitoring and remote access to the collected data.

h. Personal Computer (PC): The personal computer has been used to provide a processing power.
    - 8th generation Intel core i7 Processor
    - Windows 11 Professional Operating System

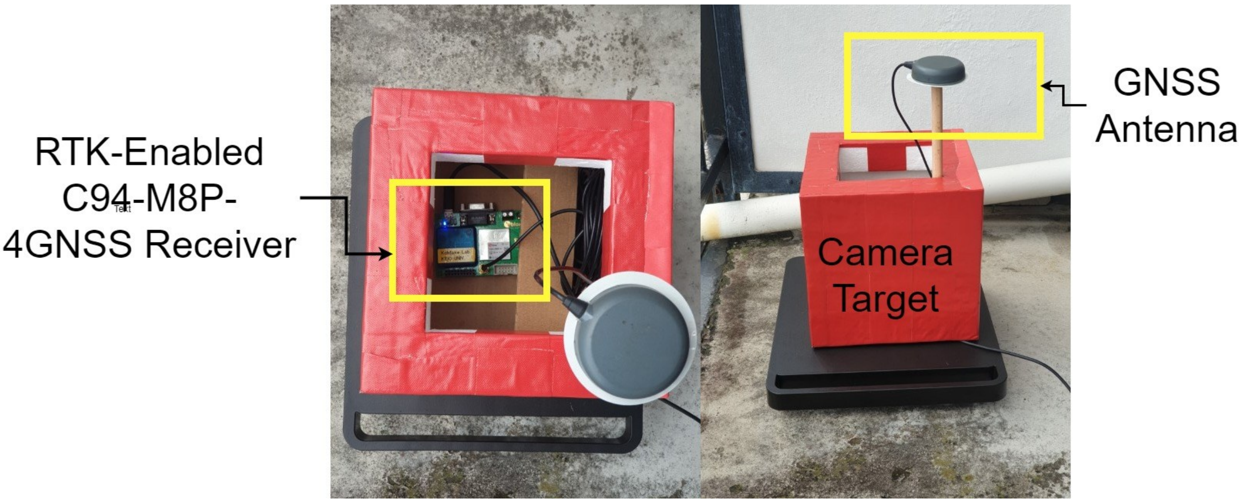


Figure 9: Camera target Embedding the GNSS Receiver

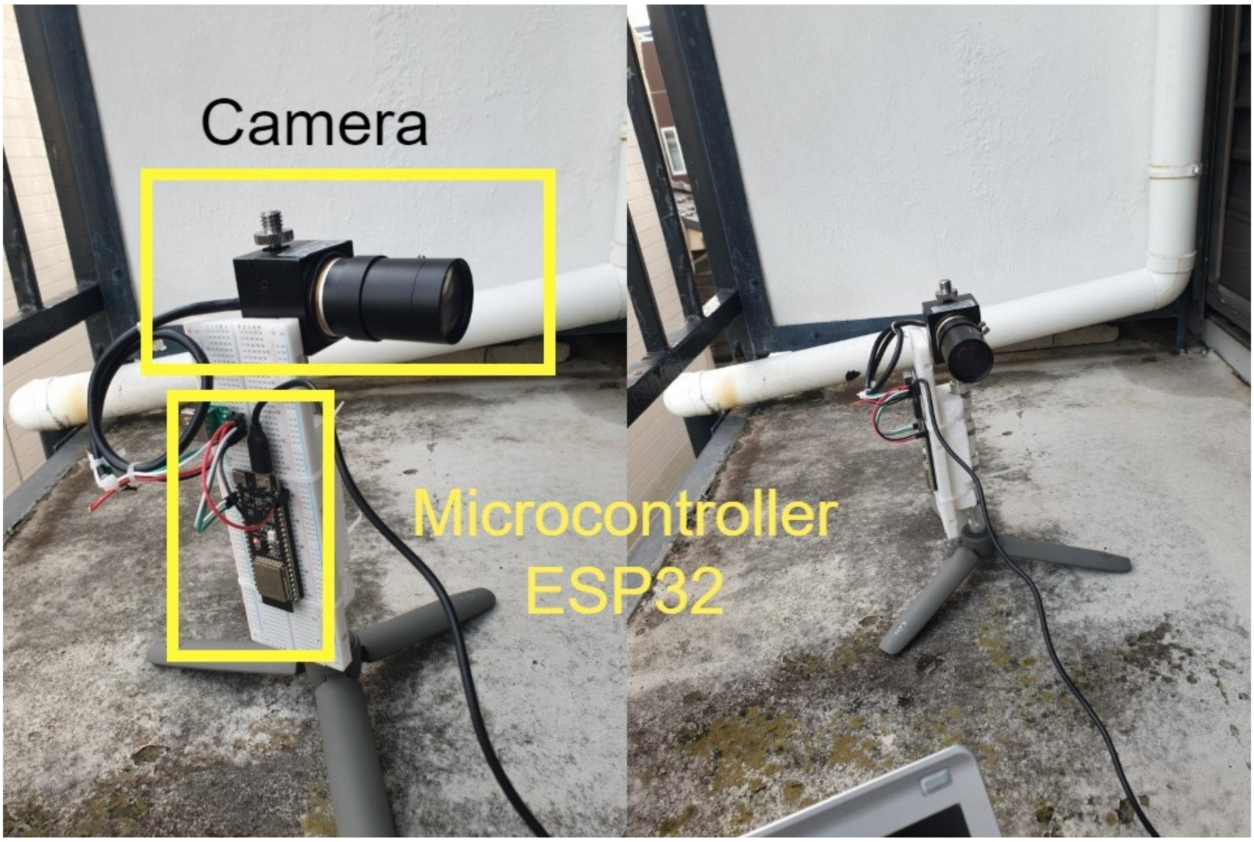


Figure 10: The Camera

IV.1.2 ***Software***

How it works:
Receive the video footage from the camera once the camera has been triggered by the GNSS movement detection. The details of the Data Flow Diagram in figure 11 are given below.

a. Conversion of Color Space The frames that are captured from the video feed are first converted from the RGB (Red, Green, Blue) color space to the HSV (Hue, Saturation, Value) color space by the software. The HSV color space is superior because it better separates color information (Hue) and lighting information (Value), which makes it simpler to recognize colors.
b. Color Masking: Once the conversion has been completed, the software will create a binary mask that corresponds to the color of the target. It then determines which pixels in the image fall within a predetermined range of color (known as the 'lower' and 'upper' bounds). The result is a binary image, where pixels that fall within the color range are represented by the value 1, and pixels that fall outside of the color range are represented by the value 0.

c. Morphological Operations The mask is then subjected to a morphological operation known as opening, which consists of erosion followed by dilation when performed by the software. This operation helps to clean the image of any small noises that may have been present.

d. Contour Detection: Once the software has obtained the final mask, it finds the contours of the. Each contour denotes the edge of an object that can be seen in the image.

e. Calculation of the Speed and Minimum Area Rectangle The software will calculate the speed for each detected contour as well as the minimum area rectangle that will enclose the contour. It does this by tracking the change in position over time, which allows it to calculate the speed of the objects.

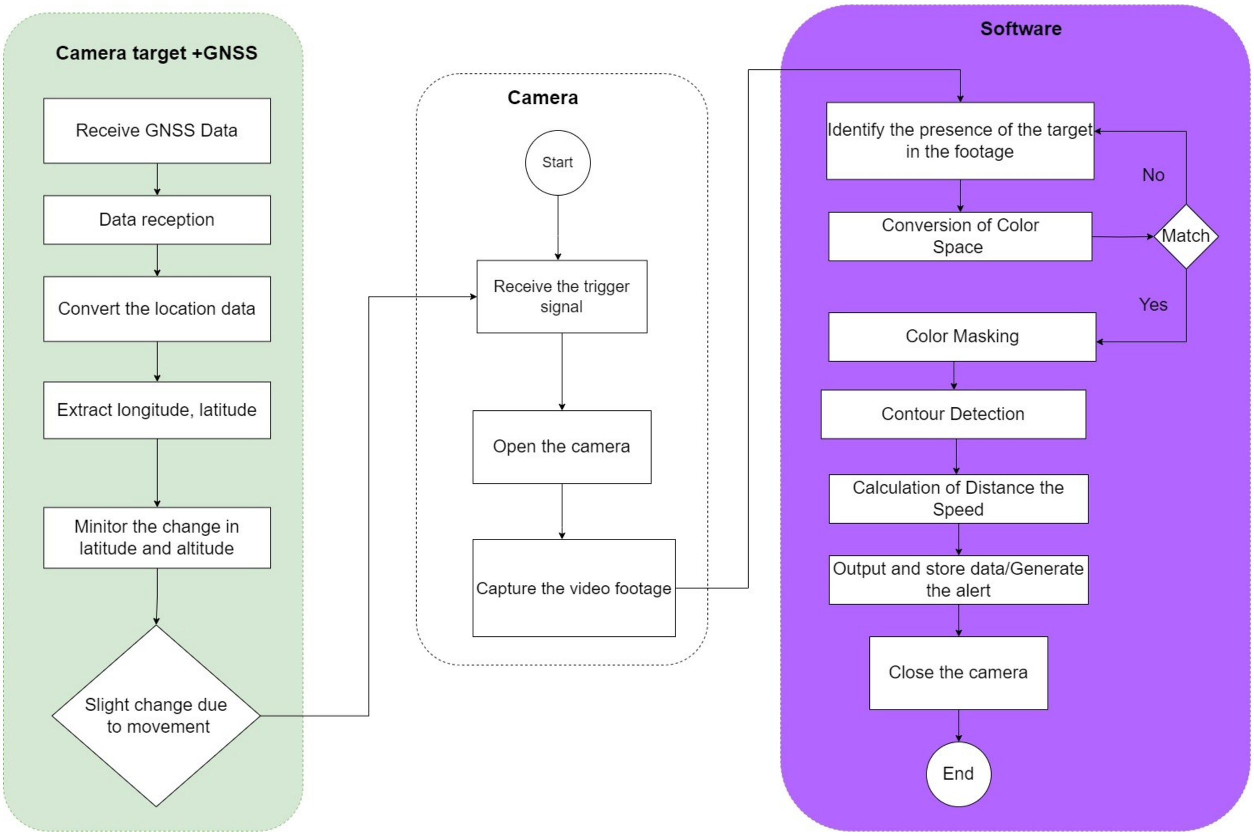


Figure 11: Holistic Data Flow Diagram

The entire procedure is performed for each frame in the video feed, enabling the software to track objects in real time.

IV.2 System experiments and Testing

The experiment consists of deployment of **camera targets** on a hilly terrain that mimics a landslide prone terrain. The hardware consists of a camera that was tested with the visibility on the camera target. We have placed the targets with color markings on the slope in the camera's field of view.

The Heron is a prototype of a landslide detection system, consisting of hardware and software. The device detects a landslide in following way:

A landslide occurs where the camera targets and GNSS receivers are deployed: in that case, the land mass moves with the target, which will change the GNSS coordinates. That position shift gives the triggers the camera which turns on to detect the target movement. targets installed on these sides will move with the land mass, and the camera will detect the movement of the target. An alert will then be generated sent to rescue services.

IV.2.1 *First round experiments*

The first round happened in Mamushidani, Keio University Hiyoshi campus on July 15, 2023. The objective of this first round evaluation is:

- The interaction of the targets and the device on the slope
- Capture and analyze the data based on the desired output, that is, the displacement (the distance that the target covers) and the speed.

Site selection

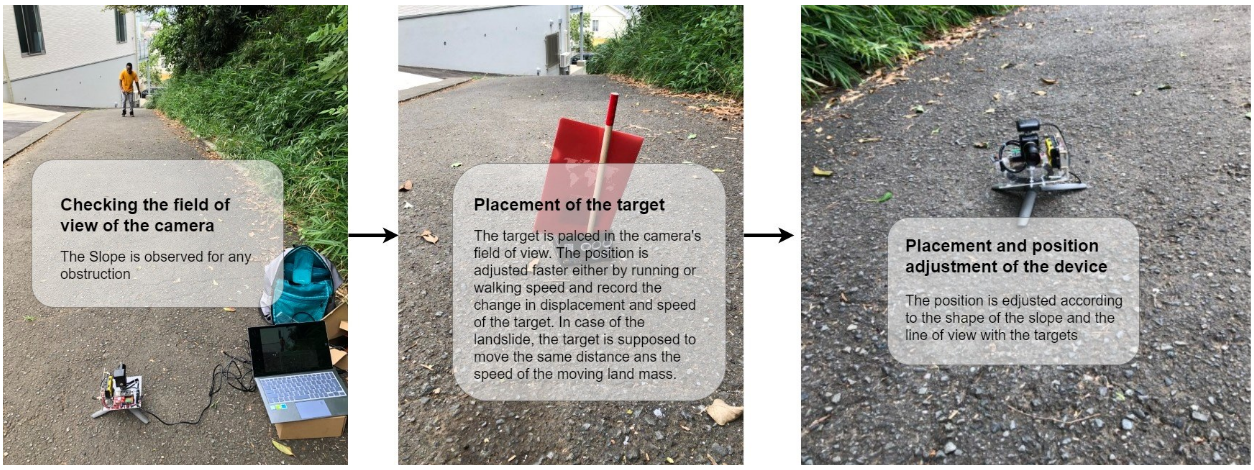
The experiment was carried at Keio University, Hiyoshi Campus in Mamushidani, Saturday July 15 from 10AM to 12PM.

The site was selected because of the slope with convex culvature and the clear unobstructed view

System setup

Connect device and check the connection

Figure 12: The experiment site



Checking the field of view of the camera

The Slope is observed for any obstruction

Placement of the target

The target is palced in the camera's field of view. The position is adjusted faster either by running or walking speed and record the change in displacement and speed of the target. In case of the landslide, the target is supposed to move the same distance ans the speed of the moving land mass.

Placement and position adjustment of the device

The position is edjusted according to the shape of the slope and the line of view with the targets

Figure 13: Experiment flow

The table 4 below shows the visibility according to the shape of the slope curvature and action taken to overcome the challenges.

Table 4:Deployment of the system in the field. Hardware vs the slope

Slope shape (between the camera and the target)	**Challenge**	**Solution**
Curved outwards	Obstructed view	Elevate both the camera and the target. We tested 166 cm height.
Curves inwards	No challenge. The view is clear	No adjustment needed
Strait line	No challenge. The view is clear	No adjustment needed

IV.2.2 *Second round experiments*

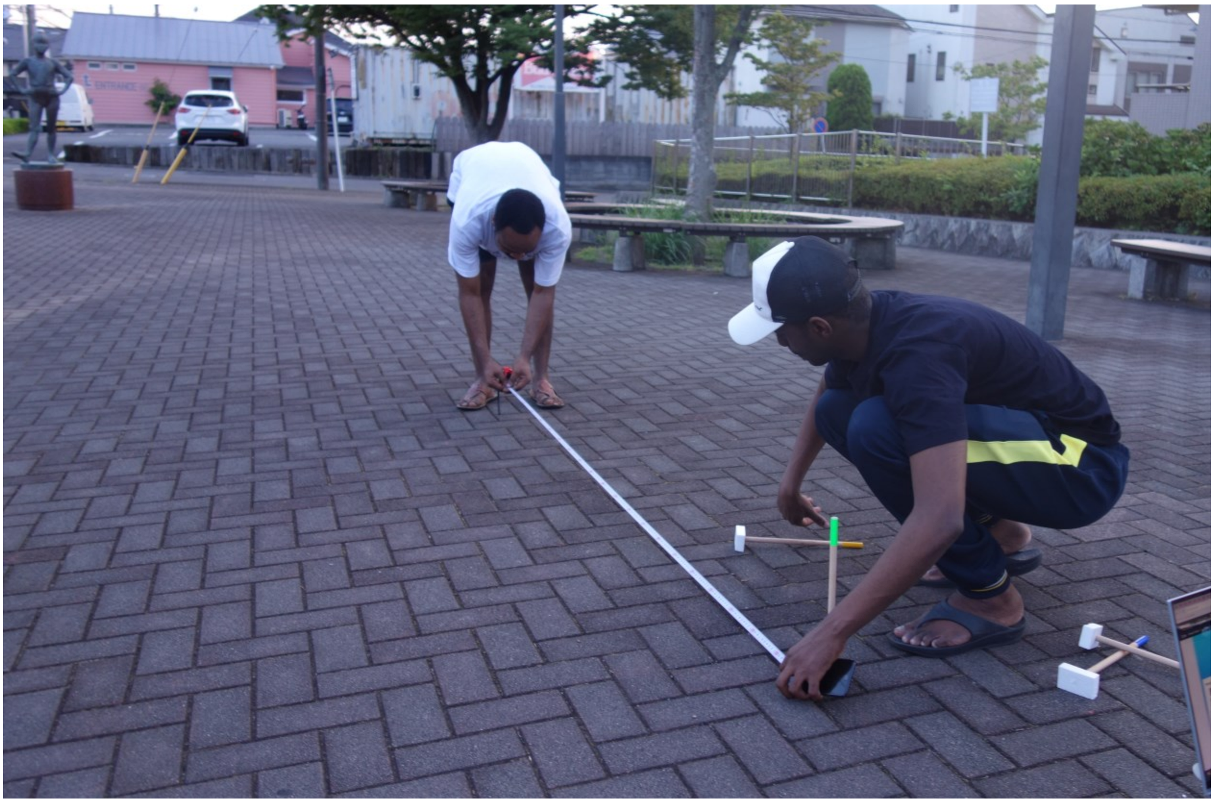
The second round happened in Ichigao (彫刻広場（オクトス市ケ尾公開空地）on July 27, 2023. The second round of the evaluation had the objective of measuring the accuracy of the system in measuring the displacement and speed and avoidance of foreign objects before the camera targets.

Picture 4: Site setup

The first experiment consists of measuring the accuracy of measuring the distance between the camera target embedding the GNSS receiver. There are two ways we conducted the experiment:

a. **Static experiment**:

We place the camera target at a known distance from the camera as seen in picture 4 above. Notice the vertical distance markers. The reference distance from the camera was five meters that we measured manually as shown on the picture 5 below.



Picture 5: Manual reference measurement to evaluate the system measurement accuracy.

The below datasets are recorded by Heron software and saved into a Comma Separated Values (CSV) file for later analysis. The distance and speed values will later confirm the occurrence or not of a landslide.

According to the calculated and recorded value of the speed, the system can deduct the nature of the landslide as shown to the figure below:

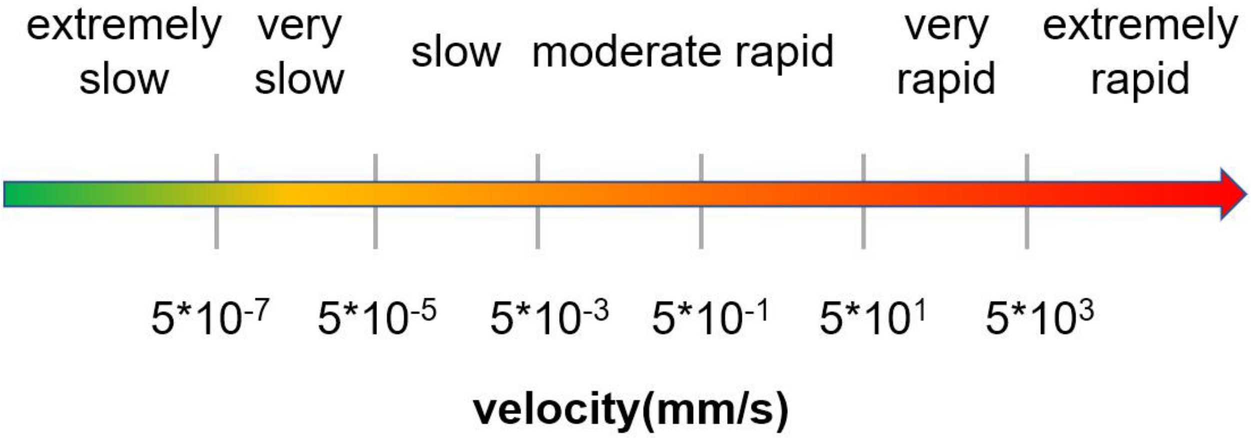


Figure 14:"Landslide grade according to velocity." [6]

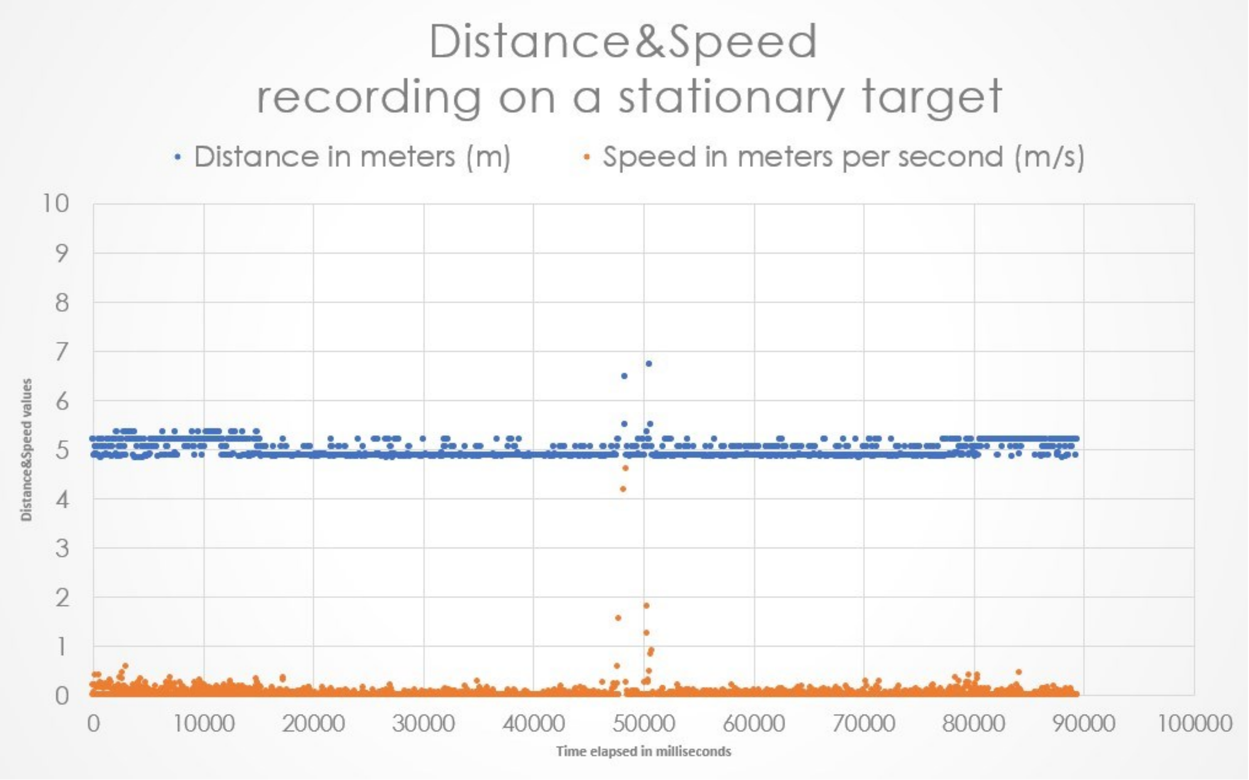


Figure 15: The scatter graph showing the recorded Distance and the speed of a stationary camera target.

From the statistical analysis of the data for a static camera target positioned at exactly five meters from the camera, these two tables were produced, with table 5 containing the raw values, meaning as they were recorded by Heron software. Table 5, on the other hand, has anomalies such as a sudden jump to the big value diverging from the normal distribution as it is clear on the graph in figure 15 above.

Table 5:Raw data accuracy assessment table

Measurement	**Dataset**	**Count**	**Mean**	**Standard Deviation**	**Minimum**	**Median**	**Maximum**
Distance	Raw	2641	4.980m	0.421m	4.813m	4.875m	18.140m
Speed	Raw	2641	0.099m/s	1.338m/s	0.000m/s	0.000m/s	30.348m

The data showed certain abnormal values of both speed and distance for a target that was still and at a known and measured distance. We filtered out these values in table 6 below to see how they can affect the whole dataset.

Table 6: Accuracy assessment data with anomalies filtered out.

Measurement	Dataset	Count	Mean	Standard Deviation	Minimum	Median	Maximum
Distance	Filtered	2628	4.965 m	0.138m	4.813m	4.875m	5.497m
Speed	Filtered	2628	0.027m/s	0.073m/s	0.000m/s	0.000m/s	1.795m/s

The following analysis compares the distance measurements in the original (raw) dataset to a reference value of 5 meters:
The standard deviation is -0.0200 meters. The original dataset's average measurement is roughly 0.0200 meters smaller than the reference value of 5 meters. This indicates very little systematic bias, with measurements typically falling just below the reference.

The standard deviation of the deviations is 0.4209 meters.
The measurements differ from the reference value in the original dataset by an average of 0.4209 meters. This value is greater than the filtered dataset, showing that the raw data around the 5-meter point is more variable.
Percentage Within 0.5 meters of the reference point 99.77%
Similarly, to the filtered dataset, almost all of the measurements in the original dataset (99.77%) fall within 0.5 meters of the 5-meter reference.
Interpretation and comparison:
- When compared to the filtered dataset, the raw dataset exhibits a slightly lower systematic bias (deviation of -0.0200 meters vs. -0.0355 meters).
- As evidenced by the bigger standard deviation (0.4209 meters vs. 0.1378 meters), the raw dataset displays greater variability in measurements around the 5-meter mark than the filtered sample.
- Both datasets show a high percentage of measurements that fall within 0.5 meters of the reference value, demonstrating good precision in comparison to the reference value.

Finally, while both the raw and filtered datasets have acceptable precision with respect to the 5-meter reference, the filtered dataset is more consistent with a smaller spread around the reference. The raw dataset, on the other hand, has a somewhat lower systematic bias, which means that the measurements are closer to the real or reference value, making them more dependable and accurate for analysis and interpretation.

b. Moving camera target Experiment

On the graph below (figure 16), observe the reading of the distance and the speed of the target. The readings were recorded when the camera target was moved both towards and away from the camera.

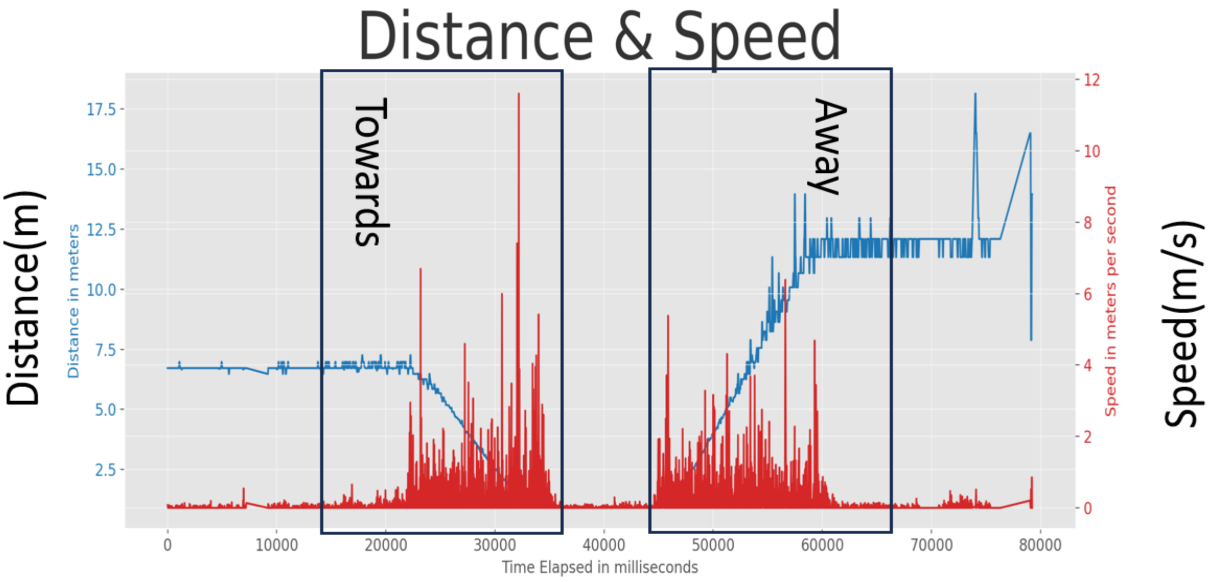


Figure 16: Distance in meters and the speed in meters per second of the targets

These values change depending on the specifications of the camera. The system can capture the distance and the speed of the targets in the field of view. The above line graphs were obtained from the records of every single frame over time.

A comparison between the performance of the proposed system and the existing methods for landslide detection or alternative algorithms. This comparative analysis aims to offer a comprehensive examination of the system's merits, drawbacks, and efficacy in relation to alternative methodologies.

The results and conclusion of this experiment mark the end of the system verification. The results to be shared with the stakeholders (rescue service) for the system validation.

IV.2.3 ***Third round experiments***

The third round of experiment has been conducted in Shimoda from August 12, 2023

The objective is verifying the synergy between the GNSS and the camera to make sure that the camera is not always on and only activated when the GNSS speed reaches 0.1m/s.

The speed has been chosen from the scale of the landslide gravity according to the speed on figure 14 [6].

The objective is to ensure the algorithm works every time.

Software: Heron Software, the source code of the python script available in the appendix.

Hardware: Windows 11 Professional PC

GNSS Receiver: Ublox Neo 6M interfaced via USB to Serial Bridge.

Camera: Logitech C980GR Interfaced via USB C

The activation by GNSS ensures the reduced use of resources leading to increased system efficiency.

IV.3 System Verification

System verification is a crucial procedure that aims to ascertain whether a system fulfills its designated requirements and effectively carries out its intended functions. System verification for this study is conducted at various stages of system design, implementation, and testing. The process of system verification methodologies to use in this case are inspection, analysis, demonstration, and testing. The errors discovered in the process of system verification help in the identification and rectification of errors, enhancement of quality, mitigation of risks, and augmentation of stakeholders (Rescue services and Victims) satisfaction.

Table 7: System Verification Methods

Requirement ID	Requirement Description	Verification Method
FR1	Continuously acquire data from the GNSS receiver.	Test: Monitor data acquisition over a set period.
FR2	Activate the camera to capture visual data when slight movement is detected by the GNSS.	Test: Simulate slight movements and observe camera activation.
FR3	Process GNSS data in real-time.	Test: Check processing latency using timestamped data.
FR4	Process visual data to identify possible landslides.	Test: Capture sample data and verify processing results.

FR5	Transmit processed data to a central monitoring station regularly or upon significant event detection.	Test: Monitor data reception at the central station.
FR6	Activate an alert mechanism upon potential landslide detection.	Test: Simulate landslide activity and check for alerts.
FR7	Store received data at the central monitoring station.	Inspection: Check stored data against transmitted data.
FR8	Control system parameters, establish thresholds, and manage communication via user interface.	Test: Change parameters and verify system behavior.
NFR1	System uptime of 99.5% with quick recovery from failures.	Test: Monitor system uptime over a set period.
NFR2	GNSS detection accuracy of at least 95%.	Test: Compare GNSS results with a reference system.
NFR3	Ensure camera measurement of the distance and the velocity with the accuracy of at least 95%	Test: Capture images/videos under different conditions and assess clarity.
NFR4	Weatherproof and durable system components.	Test: Expose components to various environmental conditions and check for damage.
NFR5	Scalability to accommodate additional sensors or cameras.	Inspection: Check system architecture and design for scalability provisions.
NFR6	Encrypted data transmission for security.	Inspection: Monitor transmitted data for encryption.
NFR7	Efficient power consumption for maximized battery or solar power operation.	Test: Attempt unauthorized access and monitor system response.
NFR8	Modular system design for easy maintenance.	Test: Monitor power consumption under normal operation.
NFR9	Intuitive and user-friendly user interface.	Inspection: Check system design and modularity.
NFR10	Maximum system response time	Test: User testing and feedback.

	of 5 seconds for alert mechanism activation.	
NFR11	Software compatibility with standard operating systems and potential for integration with other systems.	Test: Install and run software on different operating systems.

IV.3.1 *System verification metrics*

Table 8: Sample System Verification table

Requirement	**Support Data**	**Passing criterion**	**Comment**	**Conclusion**
Target position	Target distance from the device can be calculated	A positive value	This depends on the camera's focal length, and with the proper input, the system calculates the distance in cm between the camera and the position of the target.	Passed
Speed	The output value of the speed from the system	A positive value	The variation of the distance over time gives the value of the speed, considering the camera is stationary	Passed
Target detection	The system can detect the targets that match the predefined features in the software	Detect targets in shades of red	The system can detect the targets, but further steps in carefully selecting the material is necessary to overcome the natural disturbance such as sunlight and reflections	Passed with adjustments needed

IV.4 System Validation

System validation ensures that the system fulfills its intended purpose and meets the needs of the user. It entails testing and assessing the entire system in its intended operational environment.

IV.4.1 *Validation and Field Testing*

Why is System Validation Necessary? System validation verifies that the designed system functions as expected in real-world settings. It is an important element in the product development process since it ensures that the system operates in controlled environments as well as diverse and unpredictable real-world conditions.
The system's performance will be validated through field tests and real-world deployment in various regions susceptible to landslides. Its performance across various terrains, lighting circumstances, and meteorological patterns will be evaluated. Field testing enables the assessment of the system's resilience, dependability, and pragmatic viability.

Table 9: System Validation results

Validation ID	**Validation Description**	**Validation Method**	**Results**
V01	The camera detects the camera target	Field Test: Static observation of the target	Pass. All the targets falling in predefined features can be detected by the system.
V02	The System accurately measures the distance between the camera and the target.	Field Test: Static observation of the target	Pass. The results show the accuracy of 99.77%.
V03	Camera activates upon detection of slight movement by GNSS.	Observation: check that the camera activates when there is the given value for the GNSS speed	Pass. The camera is triggered by a value set as reference. We observed 0.1m/s

		is reached, deactivated otherwise.	
V04	The system can measure the speed of moving camera target	Field Test: Static observation of the target	Pass. The system measures the speed of a moving target both towards and away from the camera.

IV.4.2 *Collection of Feedback and Iterative Enhancements*

Gather input from users, stakeholders, and specialists involved in the evaluation process. Use the recommendations and insights supplied by users to iteratively improve the system's architecture, algorithms, and overall performance.

Feedback Important for User-Centered Design, meaning systems that incorporate end-user feedback are more likely to meet actual demands, resulting in higher adoption rates.

System validation ensures that the solution not only works in theory but also works in practice. For the future of this research, optimization for a solution that is both technically solid and user-centric by combining real-world testing with ongoing feedback will be essential to ensure the satisfaction of the rescue teams.

V. DISCUSSION

The utilization of the camera targets offers numerous benefits; however, light conditions present a significant challenge to the system. To see and accurately map the location of the camera targets, the prototype experiment camera must be used during daylight hours. A significant improvement could be made by using a different type of camera and developing an effective strategy for localizing targets in environments with little or no available light.

The use of reflective or glowing surfaces cannot be effective in complete darkness, which is one of the most crucial times for landslide detection. The results of the field test indicate that reflective target surfaces not only render targets useless but also fragment them into a number of individual targets.

Is there a possibility that the system could detect landslides? The preliminary system test showed it can detect predefined targets and measure the distance, displacement, and speed at which the target moves. This ability was confirmed by the test in the observed change in distance between the camera and the moving target. That way, the results observed in Figure 9 show the ability to detect the target, movement and speed according to the set features of the camera target. Figure 17 shows what will occur in the event of a landslide.

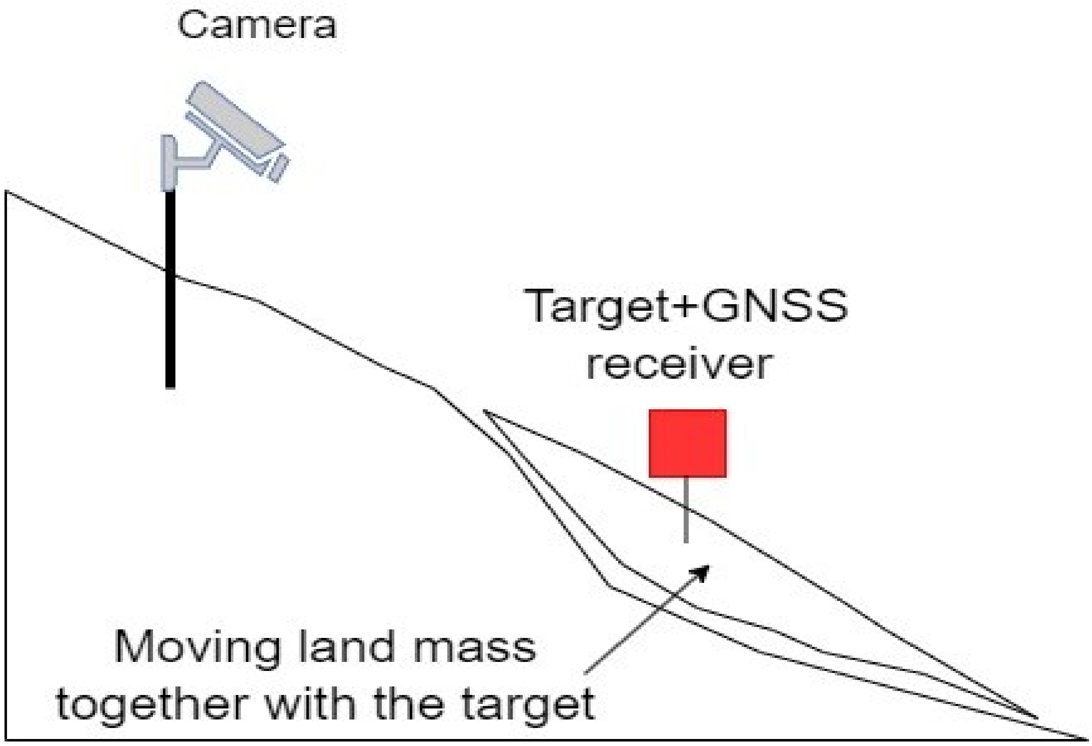


Figure 17:System deployment illustration

The characteristics of the target:

To obtain the desired results, it is necessary to consider not only the output of the software in the form of figures but also the hardware and the geotechnical characteristics of the area. These aspects of the system's environment, which were discovered through observations made during the field test, must be considered to ensure the proper operation of the system.

Color:

Red was selected for the purpose of the experiment. Color-coding the targets helps to create a uniform reference for the software to recognize them easily and ensure they are not captured along with every other passing object.

The color of interest, which will be referred to as "red," is described in terms of a range about the HSV color space. The program creates a binary mask, which identifies pixels in the image with colors within the parameters of the provided range. This mask is applied to the image before further processing. This enables parts of the picture containing the subject of interest to be extracted and separated from the rest of the image.

The size and shape:

The size of the target is extremely important when applying this method to landslide detection. The smaller targets (1.20 x 4.9 cm), which performed exceptionally well indoors and at shorter distances, did not perform at all during the field experiment, even at distances under approximately 5 m. However, an object measuring 15.6 x 22 cm (A5 in size, for comparison, or 14.85 x 21 cm) could be seen from a much further distance (approximately 20 m downhill). If this is the case, then the size greater than the size stated above should be good regardless of the orientation, up to A4 size (no larger, for cosmetic reasons, and a larger size may not be necessary).The shape is a horizontal plane perpendicular to the direction in which the camera is pointing.

Surface:

The surface is highly significant. During the experiment, the surface had a shiny appearance and was flexible. The effects of the surface on the software performance are illustrated these two characteristics.

1. The target lost its color because of the reflection, which caused the software to disregard it as an invalid selection.

2. The irregularities of the flexible surface produced reflections on different parts of the surface exposed directly to sunlight, as seen above. This means non-reflective parts of the surface were seen as different targets than would have been the case without the irregularities. As a result, a

single target was subdivided into many targets.

Therefore, the author recommends using a consistent, regular, flat, non-reflective surface for the best possible outcomes.

Position:

The shape of the slope on which the sensing station is located may obscure the camera's line of sight. The shape of the slope can present a variety of challenges, which could be compounded by the uphill placement.

For this purpose, elevating both the target and the camera (within the sensing station) to average adult height (1.66 cm at the time of the experiment) provides a direct view that is unimpeded by any obstructions.

The contours of the hillside: The outcome showed that the sensing station's performance is not affected by whether it is located uphill (see Figure 18, where a and b show the deployment of the system and the factors of the slope.)

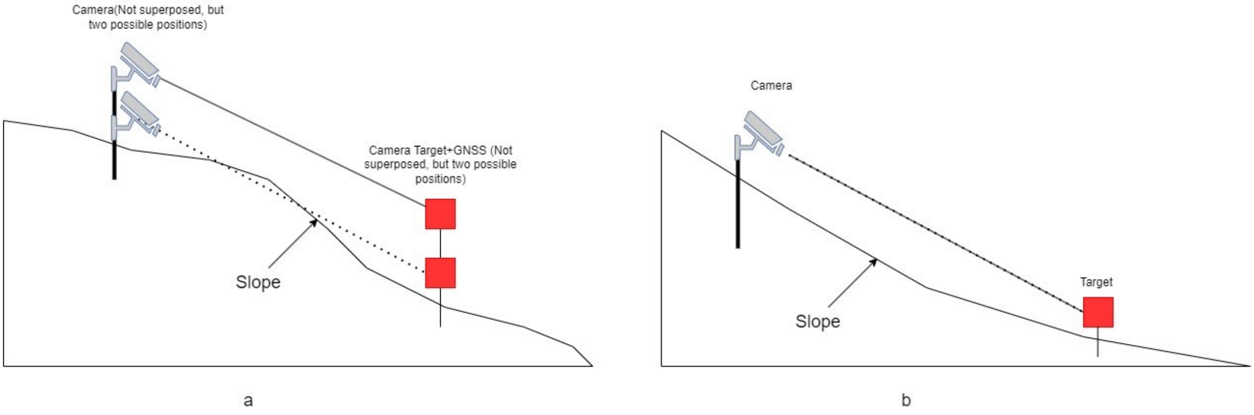


Figure 18: The optimal placement of the station and target according to the shape of the slope

V.1 Reflection on experiments

The findings of this research highlight the possible benefits that could be gained by utilizing camera targets in the process of landslide detection. The capacity to function in a wide range of lighting environments is one of the most significant advantages offered. Nevertheless, difficulties exist, most notably with regard to the constraints of carrying out operations in the dark. Previous research, which was highlighted in our literature analysis, underlined how important it was to maintain constant monitoring, and the findings of this research support those previous findings.

Our system has been put through some preliminary tests, and those tests have shown that it is able to reliably detect predefined targets, measure distances and displacements, and determine speeds. These findings are consistent with the conclusions that [8] reached, which emphasized the significance of accurate detection in order to prevent the occurrence of false alarms. This alignment is also consistent with the more general issues that have been highlighted in the research, which state that many of the present systems struggle with interferences that can result in erroneous results.

The priority that our system places on color-coded targets as a method of detection is very similar to the priority that previous research has placed on the detection of target movement. However, by centering our attention on the predefined aspects of our system, we have been able to avoid potential traps such as unnecessary computations, resulting in a system that is both accurate and effective. This strategy is consistent with the issues expressed by earlier researchers on the difficulties of distinguishing between genuine landslide precursors and visual noise.

The observations from the field tests contribute further to our growing depth of comprehension. This observation is in line with the debates that took place throughout our examination of the relevant literature, in which an emphasis was placed on the significance of distinguishing markers in detection systems.

In addition, the capability of the system to issue warnings in real time can be extremely useful during the crucial moments that immediately follow a landslide incident. This capability not only helps the nearby community, but it is also aligned with larger global aims, such as the United Nations Sustainable Development Goal 13, which emphasizes the interrelated nature of our research with worldwide efforts to promote sustainability and resilience.

V.2 Future work

It is recommended that the architecture of the system be modified to ensure compatibility with the actual conditions in the real-world setting. Aspects that should be considered include the system’s resistance to the elements and the available communication capabilities at the respective deployment sites. Light is the primary medium the camera uses to create visual representations. Landslides could potentially take place at any point in time. Therefore, the next stage of this research will focus on incorporating night vision capabilities, which will ensure system functionality throughout the entire day.

VI. CONCLUSION

This study presents a system that combines the strengths of global navigation satellite system (GNSS) monitoring and video surveillance. The foundation for this study is an extensive body of research on landslide detection. The results of our study provide further evidence that this method has the potential to provide accurate and timely landslide detection.

The utilization of techniques such as object detection and tracking, which have been utilized in the past in a variety of detection contexts, demonstrates its value in the field of landslide detection as well. The capability of our system to continually monitor slopes and use GNSS to assess movements, in combination with timely notifications, stands as a testament to the potential influence it may have in limiting the negative consequences of landslides on human settlements and infrastructure.

Our research is in line with the overarching goal of the scientific community, which is to reduce the devastation caused by landslides. The performance of the system in tests, in particular the synergy between the camera and the GNSS, demonstrates an effective utilization of resources and a potential step forward in this attempt.

In the future, the design of the system may require some modifications in order to accommodate the conditions of the actual world. These modifications may include solutions to problems such as resistance to the elements and improved communication capabilities at deployment sites. In addition, the incorporation of night vision capabilities is a potential direction for research in the future, which would ensure performance throughout the clock.

VII. REFERENCES

[1] J. Wartman, D. R. Montgomery, S. Anderson, J. R. Keaton, J. Benoît, J. D. Chapelle and R. B. Gilbert, "The 22 March 2014 Oso landslide, Washington, USA," *Geomorphology,* vol. 253, p. 275–288, January 2016.

[2] R. Ratcliffe and C. Inveen, "Sierra Leone mudslide: president calls for urgent help as search continues," October 2022.

[3] H. Bizimana and O. Sönmez, "Landslide Occurrences in The Hilly Areas of Rwanda, Their Causes and Protection Measures," *Disaster Science and Engineering (DSE),* vol. 1, p. 1–7, April 2015.

[4] B. C. S. &. C. Macaulay, "Rwanda floods and landslides kill more than 130 people," May 2023.

[5] J. Lazar, "Existing landslide monitoring systems and techniques," *www.academia.edu,* February 2015.

[6] N. Shen, L. Chen, L. Wang, H. Hu, X. Lu, C. Qian, J. Liu, S. Jin and R. Chen, "Short-Term landslide displacement detection based on GNSS Real-Time Kinematic positioning," *IEEE Transactions on Instrumentation and Measurement,* vol. 70, p. 1–14, January 2021.

[7] K. Amaki, T. Yamaguchi and H. Harada, "Landslide occurrence prediction using optical flow," 2019.

[8] A. Agrawal and S. Kar, "Development of a landslide detection surveillance system," 2022.

[9] S. Aggarwal, P. K. Mishra, K. V. S. Sumakar and P. Chaturvedi, "Landslide Monitoring system implementing IOT using video camera," 2018.

[10] N. Casagli, W. Frodella, S. Morelli, V. Tofani, A. Ciampalini, A. Tordesillas, F. Raspini, G. Rossi, L. Tanteri and P. Lu, "Spaceborne, UAV and ground-based remote sensing techniques for landslide mapping, monitoring and early warning," *Geoenvironmental Disasters,* vol. 4, March 2017.

[11] S. Romeo, A. Cosentino, F. Giani, G. Mastrantoni and P. Mazzanti, "Combining ground based remote sensing tools for rockfalls assessment and monitoring: the Poggio Baldi Landslide Natural Laboratory," *Sensors,* vol. 21, p. 2632, April 2021.

[12] C. Colesanti and J. Wasowski, "Investigating landslides with space-borne Synthetic Aperture Radar (SAR) interferometry," *Engineering Geology,* vol. 88, p. 173–199, December 2006.

[13] A. Novellino, M. Cesarano, P. Cappelletti, D. Di Martire, M. Di Napoli, M. Ramondini, A. Sowter and D. Calcaterra, "Slow-moving landslide risk assessment combining Machine Learning and InSAR techniques," *Catena,* vol. 203, p. 105317, August 2021.

[14] L. Benoit, P. Briole, O. J. F. Martin, C. Thom, J.-P. Malet and P. C. Ulrich, "Monitoring landslide displacements with the Geocube wireless network of low-cost GPS," *Engineering Geology,* vol. 195, p. 111–121, September 2015.

[15] L. Xiong, G. Wang, Y. Bao, X. Zhou, X. Sun and R. Zhao, "Detectability of repeated airborne laser scanning for mountain landslide monitoring," *Geosciences,* vol. 8, p. 469, December 2018.

[16] R. Koschitzki, E. Schwalbe, G. S. Ibbott and H.-G. Maas, "Photogrammetric monitoring concept for remote landslide endangered areas using multi-temporal aerial imagery," 2017.

[17] N. I. M. Ali, M. Drieberg and P. Sebastian, "Development of Wireless Sensor Network for slope monitoring," 2012.

[18] S. Pajalić, J. Peranić, S. Maksimović, N. Čeh, V. Jagodnik and Ž. Arbanas, "Monitoring and data analysis in Small-Scale Landslide Physical model," *Applied sciences,* vol. 11, p. 5040, May 2021.

APPENDICES

DATA COLLECTION

Location data in NMEA format will be collected by the device as well as the camera footage. The process will analyze the position of both Heron device and the marked targets and decide on the landslide occurrence and generate the alert.

The alert: visible or audible message sent to rescue services terminals (subsystem not available for this experiment).

CODE FOR HERON SOFTWARE

```
import time
import threading
import msvcrt
import serial
import numpy as np
import cv2
import time
import pandas as pd

class Camera:
    def __init__(self):
        self.cap = None

    def access_camera(self):
        """Activate and return the camera feed."""
        self.cap = cv2.VideoCapture(0)
        if not self.cap.isOpened():
            raise Exception("Could not open video device")
        return self.cap

    def deactivate_camera(self):
        """Deactivate the camera and close the OpenCV window."""
        if self.cap:
            self.cap.release()
        cv2.destroyAllWindows()

def GNSSDataAccess():
    global lat, lon, GNSSSpeed
    while True:
        print(f"\rLatitude: {lat:.5f} Longitude: {lon:.5f} Speed: {GNSSSpeed:.3f} m/s", end='')
        time.sleep(1)

videoCamera = None
# Define camera target specific variables
targetDistance = 0
focalLength = 907  # width of the object in pixels
targetSize = 20  # width of the object in cm
targetSizeInMeters = 0.1  # width of the object in meters
previousFrameTime = None
```

```python
previousCenter = None
pixelsToMeters = targetSizeInMeters / targetSize

# Initialize DataFrame to hold data
data = pd.DataFrame(columns=['Time', 'Target', 'Distance', 'Speed'])
# find the distance from the camera
def getDistance(rectParameters, image, org, font, fontScale, color):
    # find no of pixels covered
    pixels = rectParameters[1][0]
    # calculate distance
    targetDistance = (targetSize * focalLength) / pixels

    # Write on the image
    image = cv2.putText(image, 'DISTANCE FROM THE STATION CM :', org, font,
        fontScale, color, 2, cv2.LINE_AA)

    image = cv2.putText(image, str(targetDistance), (110, 50), font,
        fontScale, color, 2, cv2.LINE_AA)

    return image, targetDistance
def speedMonitor():
    global GNSSSpeed
    previousFrameTime = None
    previousCenter = None
    data = pd.DataFrame(columns=['Time', 'Target', 'Distance', 'Speed'])
    videoCamera = None
    while True:

        # basic constants for opencv
        kernel = np.ones((3, 3), 'uint8')
        font = cv2.FONT_HERSHEY_SIMPLEX
        org = (0, 20)
        fontScale = 0.6
        color = (0, 0, 255)
        thickness = 2
        cv2.namedWindow('DISTANCE TO TARGET ', cv2.WINDOW_NORMAL)
        cv2.resizeWindow('DISTANCE TO TARGET ', 700, 600)
        lastRecordTime = time.time()
        # loop to capture video frames
        while True:
            if GNSSSpeed >= 0.1:
                camera = Camera()
                videoCamera = camera.access_camera()
                print("\nCamera Activated")
                ret, img = videoCamera.read()

                if not ret:
                    break
                cv2.imshow('CAMERA FEED', img)
                hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

                currentFrameTime = time.time()
                # predefined mask for red colour detection
                lower_red1 = np.array([0, 70, 50])
                upper_red1 = np.array([10, 255, 255])
                mask1 = cv2.inRange(hsv_img, lower_red1, upper_red1)
                lower_red2 = np.array([170, 70, 50])
                upper_red2 = np.array([180, 255, 255])
                mask2 = cv2.inRange(hsv_img, lower_red2, upper_red2)
```

```
                mask = mask1 + mask2
                # Image cleanup
                cleanImage = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel, iterations=5)
                # find the histogram
                cont, _ = cv2.findContours(cleanImage, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
                # loop to find the distance of the object

                for i, cnt in enumerate(cont):
                # check for contour area
                    if (100 < cv2.contourArea(cnt) < 306000):
                    # Draw a rectangle on the contour
                        rect = cv2.minAreaRect(cnt)
                        box = cv2.boxPoints(rect)
                        box = np.intp(box)
                        cv2.drawContours(img, [box], -1, (255, 0, 0), 3)

                        img, targetDistance = getDistance(rect, img, org, font, fontScale, color)

                    # Calculate speed and add data to DataFrame outside the loop
                    if previousFrameTime is not None and previousCenter is not None:
                        currentFrameTime = time.time()
                        currentCenter = np.mean(box, axis=0)
                        distanceInPixels = np.linalg.norm(currentCenter - previousCenter)
                        distanceInMeters = distanceInPixels * pixelsToMeters
                        timeInSeconds = currentFrameTime - previousFrameTime
                        targetSpeed = distanceInMeters / timeInSeconds
                        new_data = pd.DataFrame({'Time': [currentFrameTime], 'Target': [i], 'Distance': [targetDistance], 'Speed': [targetSpeed]})
                        data = pd.concat([data, new_data], ignore_index=True)
                        previousCenter = currentCenter
                        previousFrameTime = currentFrameTime
                        # Check if 10 seconds have passed since the last record
                    if currentFrameTime - lastRecordTime >= 10:
                    # Update CSV file
                        data.to_csv('object_data1.csv', index=False)
                        # Update last record time
                        lastRecordTime = currentFrameTime

                    else:
                        previousCenter = np.mean(box, axis=0)
                        previousFrameTime = time.time()

                cv2.imshow('DISTANCE TO TARGET ', img)

                if cv2.waitKey(1) & 0xFF == ord('s'):
                    break
                videoCamera.release()
                cv2.destroyAllWindows()

                # Write DataFrame to CSV file
                data.to_csv('object_data.csv', index=False)
                #data.to_csv('./data/object_data.csv', index=False)
```

```
                time.sleep(1)  # Sleep for 2 minutes
            else:
                camera = Camera()
                videoCamera = camera.deactivate_camera()
                print("\nCamera Deactivated")

                time.sleep(1)  # Check every second

PORT = 'COM10'
BAUD_RATE = 9600
# 1 second timeout
TIMEOUT = 1
# Initialize the serial connection
serialConnection = serial.Serial(PORT, BAUD_RATE, timeout=TIMEOUT)
#Initialize variables we need
lat = 0
lon = 0
GNSSSpeed = 0

# Start the display thread
displayThread = threading.Thread(target=GNSSDataAccess)
displayThread.daemon = True
displayThread.start()

speedMonitorThread = threading.Thread(target=speedMonitor)
speedMonitorThread.daemon = True
speedMonitorThread.start()

# Read data from the serial port and print it
try:
    while True:
        line = serialConnection.readline().decode('utf-8')
        #if line:
            #print(line, end='')
        if "$GPRMC" in line:
            data = line.split(',')
            if len(data) > 7:
                # Convert latitude from DDMM.MMMM to DD.DDDDD format
                if len(data[3]) >= 2 and data[3][:2]:
                    latInDeg = float(data[3][:2])
                else:
                    continue
                latInMin = float(data[3][2:])
                latDecimal = latInDeg + (latInMin / 60)
                if data[4].upper() == 'S':
                    latDecimal = -latDecimal
                # Convert longitude from DDDMM.MMMM to DD.DDDDD format
                if len(data[5]) >= 3 and data[5][:3]:
                    lonInDeg = float(data[5][:3])
                else:
                    continue
                lonInMin = float(data[5][3:])
                lonDecimal = lonInDeg + (lonInMin / 60)
                if data[6].upper() == 'W':
                    lonDecimal = -lonDecimal
                # Convert knots to m/s
                GNSSspeedInKnots = float(data[7])
                GNSSspeedInMps = GNSSspeedInKnots * 0.514444
                lat = latDecimal
```

```
                lon = lonDecimal
                GNSSSpeed = GNSSspeedInMps

        if msvcrt.kbhit():
            break

except KeyboardInterrupt:
    # Close the serial connection when the script is stopped
    serialConnection.close()
    print("\nSerial connection closed.")

```

The script provides a means to interface with both a camera and a GNSS (Global Navigation Satellite System) device to obtain and display data. The script is a combination of GNSS data monitoring and camera feed processing to detect objects and calculate their distance from the camera. The script's primary functionality revolves around monitoring GNSS data, capturing video feed, detecting red objects, calculating their distance from the camera, and estimating their speed.

Here is a guide for users with hardware connected to a PC to test the algorithm and functioning of the system design:

- The main functionality or flow of the script.
- How the script is intended to be executed.

Dependencies: The script imports various libraries, including:

- `time`
- `threading`
- `msvcrt`
- `serial`
- `numpy` as `np`
- `cv2` (OpenCV)
- `pandas` as `pd`

Camera Class: The script defines a `Camera` class with methods for activating and deactivating the camera using OpenCV.

Other Variables: There are a few global variables like `lat`, `lon`, `GNSSSpeed`, `videoCamera`, `targetDistance`, and `focalLength`.

GNSS Data Access Function: The script provides a `GNSSDataAccess` function that seems to display GNSS data (Latitude, Longitude, Speed) every second

Function `getDistance`:

- Calculates the distance from the camera to a detected object.
- Uses the formula for distance based on focal length and target size.
- The distance is also overlaid on the image.

Function `speedMonitor`:

- This is a main loop that continuously monitors the speed from the GNSS data.
- If the GNSS speed is above 0.1, it activates the camera and processes frames.
- Uses OpenCV functions to process the frames and detect objects.
- Constantly displays the camera feed in an OpenCV window.

The continuation of the script defines the behavior of the `speedMonitor` function, focusing on detecting objects and computing their speed:

i. Red Color Detection:
    - The script uses HSV (Hue, Saturation, Value) color space to detect red objects in the camera feed. Two masks are used to capture both the lower and upper range of red in HSV, and these masks are combined.
    - Image cleanup is performed using morphological operations, specifically the `MORPH_OPEN` operation.

ii. Contour Detection:
    - Contours in the cleaned image are detected.
    - For each detected contour, a bounding rectangle is drawn if the contour area is within a specified range.
    - The `getDistance` function is called to calculate the distance of the object from the camera and overlay this information on the image.

iii. Speed Calculation:
    - Speed is calculated based on the distance the detected object has moved between frames.
    - The movement is determined by comparing the center of the bounding rectangle of the detected object in the current frame with its position in the previous frame.
    - This speed is then added to the `data` DataFrame for logging.

iv. Data Logging:

- The script accumulates the detected object's information in the `data` DataFrame.
- If the user presses the 's' key, the video capture is stopped, and the accumulated data is saved to another CSV file, `object_data.csv`.

v. Camera Deactivation:

- If the GNSS speed falls below 0.1, the camera is deactivated.

Serial Connection:

- The script sets up a serial connection (to the GNSS device) with specific port settings (`PORT = 'COM10'`, `BAUD_RATE = 9600`, `TIMEOUT = 1`).
- Initial values for latitude, longitude, and GNSS speed are set to 0.

Threading: Two threads are started:

- `displayThread`: Uses the `GNSSDataAccess` function to display GNSS data.
- `speedMonitorThread`: Uses the `speedMonitor` function to process video frames and detect objects.

The final part of the script provides information on how the GNSS data is processed:
Reading from Serial Connection:

- The script reads lines from the serial connection and looks for lines that contain the "$GPRMC" keyword, which is a common format for GNSS data.

Data Parsing:

- Latitude and Longitude:
- The script parses the latitude and longitude from the GNSS data and converts them from the format DDMM.MMMM (degrees and minutes) to DD.DDDDD (decimal degrees).
- Hemisphere information (North/South for latitude and East/West for longitude) is used to determine the sign (positive or negative) of the coordinates.
- Speed:
- The speed is parsed from the GNSS data (given in knots) and converted to meters per second (m/s).

Instructions for Using the Script

Prerequisites:

- Python: Ensure you have Python installed on your machine.
- Dependencies: Install the required libraries/modules:
    - `time`
    - `threading`
    - `msvcrt`
    - `serial`
    - `numpy`
    - `cv2` (OpenCV)
    - `pandas`

We use pip to install these:

```
pip install pyserial numpy opencv-python pandas
```

- Hardware:
    - Ensure you have a camera connected to your machine.
    - Ensure your GNSS device is connected to the COM port specified in the script (default is `COM10`). Update the `PORT` variable in the script if needed.

Running the Script:

- Navigate to the directory containing the script.
- Run the script using:

```
python synergyGNSS.py
```

- synergyGNSS.py or otherName.py are the examples. Just another file name will be fine.

Script Functionality:

- The script will continuously display GNSS data (Latitude, Longitude, Speed) on the console.
- If the GNSS speed is above 0.1, the script will activate the camera. It will then process the camera's video frames, detecting red objects, calculating their distance from the camera, and calculating their speed.

- Every 10 seconds, the accumulated data of detected objects is saved to a CSV file (`object_data1.csv`).
- Press the 's' key to stop the video capture. The accumulated data will then be saved to another CSV file (`object_data.csv`).

Output:

- The script saves the processed data of detected objects to two CSV files: `object_data1.csv` and `object_data.csv`.

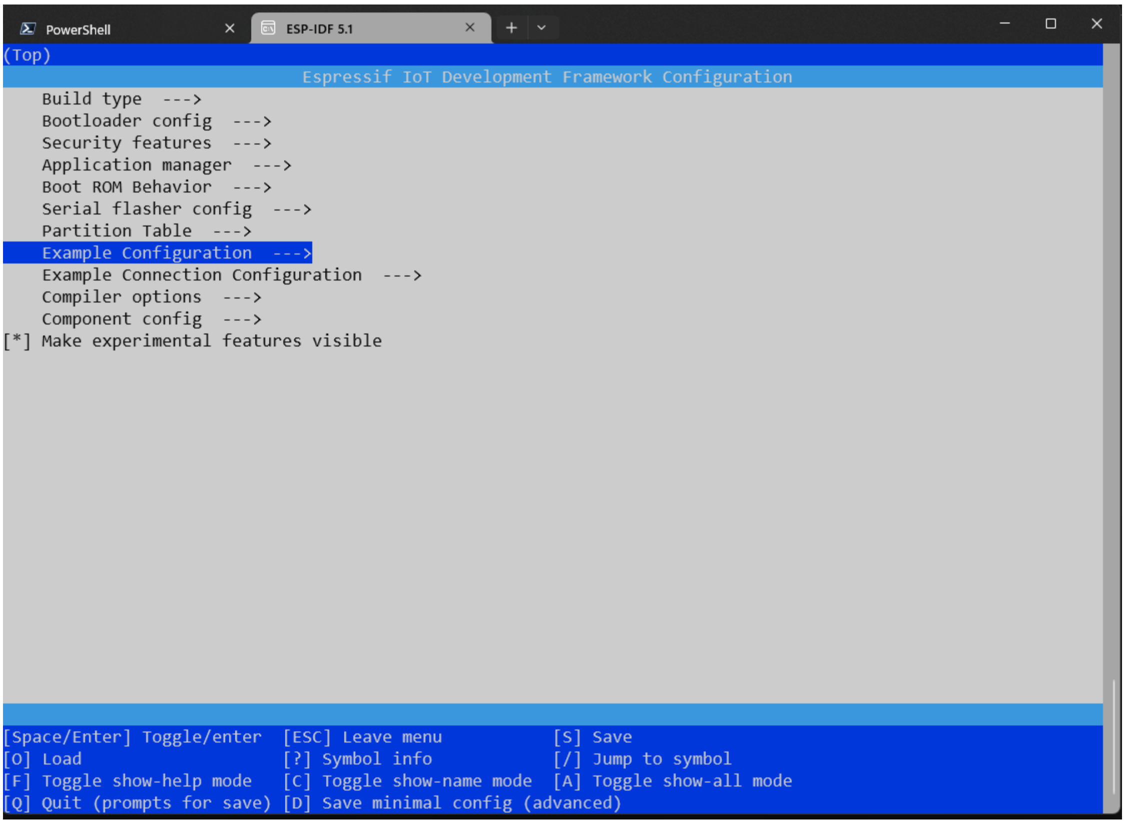
Important Notes:

- Ensure your camera and GNSS device are functioning correctly before running the script.
- The script is designed to detect red objects in the camera feed. Make sure the target objects are of this color for accurate detection.
- Adjust the `PORT`, `BAUD_RATE`, and `TIMEOUT` variables if needed to match your GNSS device's settings.
- Always safely close the script to ensure all resources (like the camera and serial connection) are properly released.

HERON HARDWARE

The Heron device was coded using the ESP-IDF, the framework provided by the microcontroller’s supplier using c programming language in case of using ESP32-S3 mentioned earlier in the thesis as a processing power for the hardware.

According to availability, A USB camera and a GNSS receiver can be interfaced to the PC.



```
(Top)
                        Espressif IoT Development Framework Configuration
    Build type  --->
    Bootloader config  --->
    Security features  --->
    Application manager  --->
    Boot ROM Behavior  --->
    Serial flasher config  --->
    Partition Table  --->
    Example Configuration  --->
    Example Connection Configuration  --->
    Compiler options  --->
    Component config  --->
[*] Make experimental features visible

[Space/Enter] Toggle/enter  [ESC] Leave menu           [S] Save
[O] Load                    [?] Symbol info            [/] Jump to symbol
[F] Toggle show-help mode   [C] Toggle show-name mode  [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Figure 19: Microcontroller configuration menu for Interfaces and Communication

Below is the sample C code running the ESP32-S3 to capture the data.

```c
#include <stdio.h>
#include <unistd.h>
#include "esp_log.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/semphr.h"
#include "freertos/event_groups.h"
#include "driver/gpio.h"
#include "usb/usb_host.h"
#include "esp_err.h"
#include "esp_log.h"
#include "esp_timer.h"
//////////////////////////////
//#include "esp_http_client.h"
#include "esp_log.h"
//#include "driver/usb.h"
#include "hal/usb_hal.h"

#define SERVER_URL ""

void camera_init()
{
    ESP_ERROR_CHECK(usb_host_install());
    uvc_config_t uvc_config = {
        .dev_speed = USB_SPEED_FULL,
```

```
        .configuration = 1,
        .interface = 0,
        .interface_alt = 0,
        .host_handle = NULL,
        .dev_handle = NULL,
        .intf_handle = NULL,
        .is_streaming = false,
    };
    ESP_ERROR_CHECK(uvc_driver_install(&uvc_config));
    ESP_ERROR_CHECK(uvc_start());
}

esp_err_t capture_frame(uint8_t **frame, size_t *len)
{
    *frame = heap_caps_malloc(640 * 480 * 2, MALLOC_CAP_DMA);
    if (*frame == NULL)
    {
        return ESP_ERR_NO_MEM;
    }
    uvc_frame_t *uvc_frame;
    esp_err_t err = uvc_stream_get_frame(&uvc_frame);
    if (err != ESP_OK)
    {
        heap_caps_free(*frame);
        return err;
    }
    memcpy(*frame, uvc_frame->data, uvc_frame->data_len);
    *len = uvc_frame->data_len;
    uvc_stream_return_frame(uvc_frame);
    return ESP_OK;
}

esp_err_t http_post(const char *url, const char *data, int len)
{
    esp_http_client_config_t config = {
        .url = url,
        .method = HTTP_METHOD_POST,
    };
    esp_http_client_handle_t client = esp_http_client_init(&config);
    esp_http_client_set_post_field(client, data, len);
    esp_err_t err = esp_http_client_perform(client);
    if (err == ESP_OK)
    {
        ESP_LOGI(TAG, "HTTP POST Status = %d, content_length = %d",
                 esp_http_client_get_status_code(client),
                 esp_http_client_get_content_length(client));
    }
    else
    {
        ESP_LOGE(TAG, "HTTP POST request failed: %s", esp_err_to_name(err));
    }
    esp_http_client_cleanup(client);
    return err;
}

void app_main()
{
    camera_init();
    gps_init();
```

```
    while (1)
    {
        uint8_t *frame;
        size_t len;
        if (capture_frame(&frame, &len) != ESP_OK)
        {
            ESP_LOGE(TAG, "Failed to capture frame");
            continue;
        }
        gps_data_t gps_data;
        if (gps_read(&gps_data) != ESP_OK)
        {
            ESP_LOGE(TAG, "Failed to read GPS data");
            continue;
        }
        char gps_json[128];
        snprintf(gps_json, sizeof(gps_json), "{" lat ":%f," lon ":%f}", gps_data.lat, gps_data.lon);
        char *data = malloc(len + sizeof(gps_json));
        memcpy(data, frame, len);
        memcpy(data + len, gps_json, sizeof(gps_json));
        if (http_post(SERVER_URL, data, len + sizeof(gps_json)) != ESP_OK)
        {
            ESP_LOGE(TAG, "Failed to send data to server");
        }
        free(data);
        free(frame);
    }
}

```