2023-08

Development of early warning system for human wildlife conflict using deep learning, IoT and SMS

Ronoh, Emmanuel

NM-AIST

https://doi.org/10.58694/20.500.12479/2199

DEVELOPMENT OF EARLY WARNING SYSTEM FOR HUMAN WILDLIFE CONFLICT USING DEEP LEARNING, IoT and SMS

Emmanuel Kipchumba Ronoh

A Project Report Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

August, 2023

ABSTRACT

Human-wildlife conflict is a significant challenge to communities living in areas close to wildlife game parks and reserves. It is more evident in the United Republic of Tanzania whose economy depends on agriculture and wildlife tourism as a significant source of income for her citizens and foreign exchange respectively. The proposed system is a low-power and low-cost early-warning system using deep learning, Internet of Things (IoT) and Short Message Service (SMS) to support human-wildlife conflict response teams in mitigating these problems. The proposed system comprises three basic units: sensing unit, processing unit, and alerting unit. The sensing unit consists of a Global Positioning System (GPS) module, a passive infrared (PIR) sensor, and a Raspberry Pi camera. The PIR sensor module detects animal nearby using its heat signature, the GPS collects and records the current system location while the Raspberry pi camera takes an image after the PIR sensor has detected the animal nearby using its heat signature. The processing unit with the main unit uses a Raspberry microcomputer to perform image inferencing using the “you look only once” (YOLO) algorithm and data processing. The last unit is an alerting unit that uses Global System for Mobile Communications module to send an alerting SMS message to the community response team leader and the human-wildlife conflict response team whenever wild animals are detected near the park’s border. Therefore, the system detects, identifies, and reports wild animals detected using SMS. General Packet Radio Service cellular network provides internet connectivity for the purpose of data collection to enable monitoring and storage in the cloud. An online visualization system was developed using google maps to show the location of wildlife detected by the camera trap. The park rangers track the wildlife online to acquire important information before the wildlife wanders out of the park. This system was developed using the open-source Raspberry pi which is cost-effective even for low-income communities who are targeted by the system.

DECLARATION

I, Emmanuel Kipchumba Ronoh, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this project report is my original work and that it has neither been submitted nor being concurrently submitted for a degree award in any other institution.

Emmanuel Kipchumba Ronoh		23/08/2023
Name of Candidate	**Signature**	**Date**

The above declaration is confirmed by:

Dr. Mussa Ally		23/08/2023
Name of Supervisor 1	**Signature**	**Date**

Dr. Silas Mirau		23.8.2023
Name of Supervisor 2	**Signature**	**Date**

COPYRIGHT

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by the Nelson Mandela African Institution of Science and Technology, a project report titled ***"Early warning system for human-wildlife conflict using IoT and SMS"*** in partial fulfillment of the requirements for the degree of Master of Science in Embedded and Mobile Systems of the Nelson Mandela African Institution of Science and Technology.

Dr. Mussa Ally		23/08/2023
Name of Supervisor 1	**Signature**	**Date**

Dr. Silas Mirau		23.8.2023
Name of Supervisor 2	**Signature**	**Date**

ACKNOWLEDGMENTS

I would like to thank my supervisors Dr. Mussa Ally and Dr. Silas Mirau, Centre of Excellence for ICT in East Africa (CENIT@EA) and KUZA steam generation especially Rebbeca Ryakitimbo for the support they have provided me in preparing and writing this project report.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

CNN	Convolutional neural network
CSS	Cascading Style Sheets
Fast RCNN	Fast Region-based Convolutional Neural Network
Faster RCNN	Faster Region-based Convolutional Neural Network
GDP	Gross domestic products
GPS	Global positioning system
GSM	Global system for mobile communication
HTML	Hypertext markup language
HWC	Human-wildlife conflict
HWCRT	Human-wildlife conflict response team
IoT	Internet of things
MP	Mega Pixel
MySQL	My structured query language
PHP	Hypertext preprocessor
PIR	Passive infrared
RAM	Random access memory
RCNN	Region-based Convolutional Neural Network
SMS	Short messaging service
SSD	Single shot detector
TANAPA	Tanzania national parts authority
TPU	Tensor Processing Units
URL	Universal Resource Locator
YOLO	you look only once

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

Human-wildlife conflict (HWC) is an interaction between wildlife and humans with a negative outcome. It mainly occurs in agricultural areas bordering game parks and reserves. However, it can also happen in non-agricultural areas such as urban and peri-urban neighboring conservations (König *et al.,* 2020). From a human perspective, HWC is caused by crop-raiding, injuring, and the killing of domestic animals and community members by wild animals. This conflict occurs in a reciprocal process in nature. Therefore, both humans and wildlife are negatively affected by conflicts aggravated by scarce resources and habitat intrusion (Sharma *et al.,* 2021). The communities affected by HWC mostly retaliate by killing or encouraging poaching, which negatively affects the wildlife population, especially the endangered species with dwindling populations. The major causes of this problem are the growing human population and loss of wildlife habitat due to encroachment on wood logging and farming.

Tanzania's population increased from 11.7 million in 1965 to 44.9 million in 2012 (Mwakisisile *et al.,* 2019). This population increase caused more demand for agricultural production and agricultural areas. The country has recently increased land expansion to support agricultural production and demand from the increased population (Mkonda *et al.,* 2018). Tanzania's economy heavily relies on agriculture which accounts for over 25% of its Gross Domestic Product (GDP) (Chongela, 2015). In addition, agriculture is a significant source of income for many Tanzanian households. Therefore, the Tanzanian government had identified agriculture as a factor to consider in poverty reduction (Sarris *et al.,* 2006).

The availability of a wide variety of wildlife in Tanzania is also a significant source of tourism attraction. In addition, tourism is a substantial source of foreign exchange for the government of Tanzania (Weldemichel, 2020). According to Weldemichel (2020), wildlife-based tourism accounts for 17% of Tanzania’s GDP. An individual animal such as an elephant bull can generate more revenue for the country and communities through tourism throughout his lifetime than the value of his horn and meat combined. Tourism growth has also been shown to generally improve the economy in tourism-related sectors such as transport, hospitality, and banking and unrelated sectors such as farming (Kyara *et al.,* 2021). Besides, wildlife is part of nature whose main function is to keep the environment in check and balance. For example, the bat-eating crocodiles in Australia hold the population of bats in an environment-supported number.

Although wildlife protection is a national priority, communities living adjacent to protected areas should be considered to avoid negative impacts from conflicts. The conflict between humans and wildlife has been registered in different parts of Tanzania, particularly in the northern region. According to the statistics from the wildlife department in Tanzania, HWC is a growing concern, as shown in Table 1. The main wild animals in these conflicts are predatory carnivores and crop-raiding animals such as buffalos and elephants. Carnivorous animals such as lions predate livestock in the enclosed Maasai bomas and Maasai retaliate by killing them (Broekhuis *et al.,* 2017). These conflicts are caused by a rapid loss of natural wildlife habitats to croplands (Branco *et al.,* 2020). The protection mechanism of the bomas is vital; hence having strong fences and more guarding dogs to scare away the wildlife who wander into the villages or bomas is recommended.

Table 1: Human-wildlife conflicts statistics from the wildlife department of Tanzania

Year	Human deaths	Permanent injuries	Temporary injuries	Livestock deaths	Crop damage (acres)
2012/13	69	23	38	46	1518
2013/14	61	31	49	93	4046
2014/15	59	20	41	107	6786
2015/16	102	20	78	64	8924
2016/17	132	30	54	130	4567
2017/18	380	20	29	149	5016
2018/19	266	60	149	203	10 547
Total	**1069**	**204**	**438**	**792**	**41 404**

This conflict is destructive because it causes the loss of property, livestock, or human life. The government of Tanzania has devised ways of including the community in wildlife conservation. For instance, the establishment of Wildlife Management Areas on village land ensures that community members participate and benefit from wildlife management (Benjaminsen *et al.,* 2013). However, HWC has been a significant reason that undermines wildlife conservation among people living near wildlife conservation areas.

Typically, information about HWC comes in different ways, including the zonal hotline number reports to a given area's human-wildlife conflict response team (HWCRT). This method is efficient because the residents immediately report spotted wildlife to the human-wildlife response team. Another way is by collaring large-body wildlife such as elephants and rhinos to monitor and track their movements to geo-fence them. For instance, the current Ngorongoro conservation area system uses collaring of animals such as elephants for geofencing alerts. This system is installed on animals that tend to cause crop-raiding sprees. Although it is an effective method, some animals may go astray and raid crops. The Maasai around Ngorongoro has enclosed fences around manyattas but sometimes wildlife breaks these fences and raids the crops.

The methods described above have significantly improved efficiency in handling HWCs. However, continuous monitoring of wildlife using collaring has proven expensive to install and operate due to its networking requirements and the initial cost of equipment, especially in remote areas with poor network coverage and low population density (Giefer *et al.,* 2020). Similarly, the hotline system utilizes a reactive approach, hence the need for a cheaper and proactive approach. Technology has been improving and evolving. Technology has changed how sensors and actuators are integrated with communication mediums in the internet of things (IoT), thereby revolutionizing how industries work. This technology provides a way in which an embedded system gets data from sensors and transmits it to the cloud or online server. These embedded devices need slight improvement to conventional camera traps, such as using microcomputers such as Raspberry pi, which support machine learning algorithms to support early-warning systems.

The processing of images is laborious, expensive, and time-consuming but advances in deep learning methods have shown promising outcomes in object and image identification. For example, some deep learning models distinguished humans and animals by classifying the pictures (Şimşek *et al.*, 2018). Therefore, a low-cost camera trap was developed in the present study to notify the relevant authorities of the exact wildlife in the vicinity.

1.2 Statement of the Problem

It is uncommon for wildlife to stray away from the park and sometimes raid communities farms. Therefore, HWC is a critical factor affecting farming in areas bordering parks when wildlife wanders away from the park into people's farms where they raid crops and sometimes destroy property. Some wildlife also causes injuries or death to farm animals and human beings. The economic and emotional impacts on affected communities can be significant, with families losing their primary source of income, and sometimes suffering physical harm or even loss of life.

Currently, the early-warning systems in places involve wildlife collaring which is quite effective since it helps in knowing the location of collared wildlife at all times. The movement of wildlife is monitored and reported in real-time (Massawe, 2021). However, the problem with this system is that it tracks only collared wild animals. Therefore, it is expensive to use in tracking all wildlife. It is also challenging to implement the system on large and aggressive wild animals such as elephants and carnivores such as lions and leopards. The system is not also suitable for small-bodied wildlife such as wild dogs or wildlife which live in water, such as crocodiles.

The other solution is the use of zonal hotline numbers to report HWC and wildlife intrusion in the communities has shown success according to studies by Graham *et al.* (2012). However, this system is reactive because it reports incidents when they have already occurred.

1.3 Rationale of the Study

The objective of the project was designed and developed an early-warning system for HWC. Its primary purpose is to alert the park HWRT officers about wildlife wandering away from the park and notify the community HWC response association about the potentially dangerous wild animals roaming toward their farm and residences. Hence, the system was designed to complement and support existing HWC mitigation tools and methods used in areas bordering parks and other wildlife reserves.

1.4 Objectives of the Study

1.4.1 General Objective

To design and develop an early warning system for human wildlife conflict based on camera traps and short message service (SMS) to detect and alert park warders on wildlife escaping the Tarangire National Park and communities living adjacent to the park.

1.4.2 Specific Objectives

The study aimed to achieve the following specific objectives:

(i) To analyze the requirement for developing an early warning system for human wildlife conflict.

(ii) To develop an early warning system for human wildlife conflict.

(iii) To validate the developed system.

1.5 Research Questions

The study intended to answer the following questions:

(i) What are the requirements for developing an HWC early warning system using a camera trap?

(ii) How to monitor wildlife in the park using camera traps?

(iii) Is the system working as anticipated?

1.6 Significance of the Study

This project was developed with the hope of contributing to body of knowledge on the development and implementation of early warning system for human wildlife conflicts using camera traps and SMS technology, particularly in the context of African wildlife reserves.

The output of this early warning system for HWCs will ensure the safety of the residents as well as the wildlife by providing information about the animals getting out of the park in a timely.

1.7 Delineation of the Study

The scope of this study is focused on the development of an early warning system for human-wildlife conflicts, with specific limitations and conditions. Firstly, the system is designed to monitor wildlife in only one area facing a single direction, which means that it may not be suitable for use in more extensive or complex areas. Additionally, the wild animals detected by the system are determined by the trained model used, which means that it may not identify other wildlife not included in the training. The system is limited to working with specific wildlife found in the Tarangire National Park, including the big five wildlife and a few other selections.

Furthermore, the study is concerned with the power supply for the system, which is harvested from an external battery that is rechargeable or uses a solar panel. As a result, the placement of the system must be in a direction that can receive sunlight for the solar panel to function effectively. These limitations and conditions are crucial to the effectiveness and functionality of the early warning system and are necessary to consider during the design and development phases.

Finally, the system is also susceptible to the effects of strong winds and heavy rain, which can damage the equipment and affect the quality of the images captured. Therefore, these environmental factors must be considered when deploying and using the system to ensure its effectiveness in mitigating human-wildlife conflicts.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

Studies have shown that mobile phone ownership in Tanzania is high, with most people owning featured phones that support a standby mode for a long time and do not consume a lot of power. This presents an opportunity to use mobile phones for HWC early-warning systems (Lewis *et al.,* 2016). Furthermore, the recent COVID-19 pandemic has resulted in increased internet usage, with many activities, including learning, being done remotely. This has led to a significant increase in smartphone use in Tanzania, which has internet access (Pandey *et al.,* 2020). Therefore, most people had to switch to smartphones which have access to the internet.

Internet coverage in Tanzania is remarkable, with even remote areas such as the Tarangire National Park having access to a 2G network that can support calls and light browsing. This presents an opportunity to use internet-connected devices for HWC early-warning systems. Additionally, advancements in machine learning algorithms have allowed lightweight machine learning to run on small processors and memory. This has enabled the development of low-power-inferencing systems that can support upcoming systems.

2.2 Mitigating Human-Wildlife Conflicts

Human Wildlife Conflict (HWC) mitigating strategies are many but some include using an electric fence and trenches to stop animals from intruding into human settlements or homesteads, especially for small body wild animals that cannot break such barriers (Shaffer *et al.,* 2019). An electric fence is a common method in significant parks to discourage wildlife from escaping out of parks and nearby community livestock from entering the parks (Wijayagunawardane *et al.,* 2016). A study of electric fences effectiveness using collar data and camera traps to determine wildlife behaviours around the fences and proved to be effective (Branco *et al.,* 2020). However, due to the cost of power and breakage of fences by elephants using their tusk which does not conduct electricity (Massey *et al.,* 2014), different strategies to minimize cost have been used. For instance, activating power when a collared animal is near the fence using collaring data but its' use is limited due to the high subscription fee to transmit data and the challenges of capturing and collaring animals.

The other option is erecting a barrier using beehives to scare away wild animals such as elephants (O'Connell-Rodwell *et al.*, 2000). This system also benefits the local community because it provides a source of income through honey harvest. However, this option is cumbersome and laborious since it needs the installation and regular repairs after becoming operational (Karidozo *et al.*, 2015). The available solution is using ultrasonic repellent to scare away wild animals after sensing them (Giordano *et al.*, 2018). Nevertheless, different strategies should be developed and adopted since elephants easily be habituate to harmless methods according to prior study (O'Connell-Rodwell *et al.*, 2000).

2.3 Deep Learning Systems

Deep learning systems have been widely used in various applications related to wildlife conservation. In particular, they have been used for estimating animal populations (Connor *et al.*, 2022), studying animal behaviors (Znidersic, 2017) and analyzing survival rates of certain species (Williams *et al.*, 2020).

Computer vision is a crucial aspect of deep learning and enables the detection and identification of objects in images or videos. Object detection usually involves proposed region selection, feature extraction, and classification using a trained classifier. Traditional machine learning and deep learning are the two approaches used for object detection, with deep learning being more accurate with larger datasets (Kiruthika *et al.*, 2020). Convolutional Neural Network (CNN) is a powerful algorithm for data mining in deep learning classification problems, and there are two main types of object detection methods: Regression Object Detection and Region-Based detection. While Region-Based detection method have higher accuracy in object detection, they are slower in operation. In contrast, Regression Object Detection algorithm generate region proposals and classify them concurrently, making them faster than the Region-Based detection methods. The YOLO algorithm is a popular Regression Object Detection method that is faster than the SSD in object detection and is suitable for use in scenarios with small microcomputers that have low-quality cameras and require prompt results to save on power consumption. The YOLO has a radiant speed of processing 45 edges per second, making it a suitable option for wildlife conservation applications (Manjari *et al.*, 2019).

2.3.1 Image Classifications

Image is the visual perception depicted from an artefact (Vangorp *et al.*, 2011). Image classification refers to the process of categorizing images based on their visual features and content. This technique has been widely applied across diverse domains, including medicine, agriculture, and tourism, to enable accurate diagnosis and detection. In the realm of computer vision, researchers have explored various approaches to address a range of problems, such as object classification, detection, recognition, and segmentation (Wang, 2016). To enable effective image classification, the initial step involves the labeling of images, which allows the extraction of distinct features and characteristics associated with each label.

2.3.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is a widely used architecture in machine learning due to its high performance in image recognition tasks. It mimics the human brain's ability to interpret images by using hidden layers to analyze them. The CNN architecture is composed of various layers, including the input, convolutional, pooling, flattening, and output layers. The input layer prepares the image for processing and passes it to the convolutional layer, which filters the image to acquire emphasized features. The pooling layer then preserves these features, while the flattening layer prepares them for outputting. In a recent study by Premarathna *et al.* (2020), a CNN model accurately identified elephants with 94% accuracy, demonstrating the potential for automated wildlife identification and reporting.

2.3.3 Deep Learning on the Edge

The emergence of deep learning on edge has been made possible by advancements in microcontroller technology and improvements in deep learning techniques for smaller processors. This has allowed for object detection on microcomputers, which has proven useful in scenarios such as visually-based guidance for the visually impaired (Tepelea *et al.,* 2019) and security purposes, where individuals can be detected and any bags they may be carrying can be identified (Khan *et al.,* 2019). The accuracy and efficiency of these models are determined by the data used for training, as well as the quality of the camera and processing power of the microcomputer (Ghosh *et al.,* 2020). This improvement has resulted in faster processing speeds, as the need for data transmission and reliance on external computers or servers is eliminated.

2.4 Early Warning Systems for Human-Wildlife Conflicts

In Tanzania, the government has implemented various methods to mitigate HWC, including zonal hotline numbers that allow citizens to report incidents to the HWRT in real-time. The use of mobile phones has been particularly helpful for Maasai communities in reducing the severity and occurrence of HWC events (Lewis *et al.*, 2016). Ealy reporting facilitate collaboration between local official and farmers in driving away problematic wild animals (Graham *et al.*, 2012).

However, these methods are reactive in nature, as they rely on reports after the incident has occurred. A proactive approach such as collaring wildlife could provide real-time information to prevent HWC incidents. Wildlife collaring is an accurate method of monitoring wildlife movement, allowing for real-time geofencing of wildlife movement (Caro *et al.*, 2016). This approach has been helpful in overall wildlife monitoring. However, collaring has its limitations, such as its high cost for design, implementation, installation, and repair. Monitoring of wildlife using collars can be expensive, particularly in remote areas with low population density and poor network coverage (Giefer *et al.*, 2020).

2.5 Related Works

2.5.1 Early Warning System

To solve this issue of early warning for HWCRT and its surrounding communities' different strategies have been adopted. This system works effectively in different environments and conditions. The species of wildlife to be reported and the environmental factors determining the types and nature of sensors employed in the systems are determined. The animals also influence approaches to detecting it. Different methods and approaches have been adopted as early-warning systems. However, the costs and implications of various systems differ in terms of efficiency and practicality.

Kumar *et al.* (2021) developed a system using passive infrared (PIR) sensors and motion detector to detect wildlife moving to the human habited area. The system could notify the registered residents using the mobile app. It also included a loudspeaker system to chase the wild animals away using loud noise.

Giordano *et al.* (2018) developed a crop protection system using a PIR sensor to detect wild animals and an ultrasonic speaker to chase away the wildlife. This system used a low-power RIOT operating system and a tailored microcontroller. As a result, the system can detect wildlife but cannot identify each species.

Fazil *et al.* (2018) designed a system that uses a seismic sensor to detect groundwaves of elephants' feet trumping the ground and then alert people using the SMS and speakers. The only drawback of this system was that it could only detect wildlife, generating ground vibrations such as elephants.

2.5.2 Deep Learning Wildlife System for Early Warning

Divya *et al.* (2018) developed a wild animal intrusion detection system. The system incorporated camera traps consisting of a camera to capture the image of wildlife, a PIR sensor to sense the wildlife, and a computer to perform image processing. The images captured by the camera trap are sent to a powerful computer to process and identify the wild animals. However, this system requires good bandwidth to facilitate communication between the personal computer and the camera trap for effective operation.

2.6 Research Gap

The challenges of HWC have attracted different suggestions to mitigate them. The first step is to detect and report the wildlife escaping the park. Various methodologies have been adopted to detect wildlife. Some researchers have used a species-specific sensor that detects certain species, such as seismic sensors to detect elephants while others were using deep learning, but inferencing was done from the server which consumes a lot of internet bandwidth in transmitting image data to the server.

In the proposed system, deep learning at the edge was used to detect the wildlife and report the name of identified wildlife. The images of wild animals of interest were used to develop deep learning model. The developed YOLO deep learning algorithm was chosen because it supports low resources computers and detects tiny objects in images and the TensorFlow framework works in microcomputers such as Raspberry pi. The park of our study does not have a fence between the human population area and the park; therefore, the only barrier between the human population and the park is the expansive area used to control wildlife movement to human settlement.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Development of Designed System

The development model used was the scrum Agile SDLC model. It is a combination of iterative, collaborative, and incremental process models and decisions between requirement and solution teams with a focus on process adaptability and customer satisfaction by rapid delivery of working software products. The scrum agile system development lifecycle is as shown in Fig. 1. This is because most consumers will not wait for the system to be developed but take the solutions available at the moment and fast pace the coming up of new technologies. Therefore, some products may become outdated before becoming products in the market. Product improvements happen as the product is tested and deployed.

This methodology was developed in the early 90s as a framework for managing complex development projects (Schwaber, 2004). It focuses on monitoring software development cycles from Requirement Specifications to Integration Tests and provides support for the intermediary Design, Coding and Unit Testing Phases (Rajagopalan *et al.,* 2016).

The methodology has four major components (team structure). These components are as follows. Scrum master. This was my industrial supervisor who facilitate the project to ensure scrum rules and practice are followed. They are responsible for project success and helps product owner in selecting what to be included in priority backlog and development team to turn backlog into a functionality (Schwaber, 2004).

Product owner: its main responsibility was to decide which feature or function was build and decide what to be include in the priority backlog. This was a representative send from the company to ensure that the product was what they really wanted.

Development team: this a team responsible to implement functionalities given in the backlog into the system.

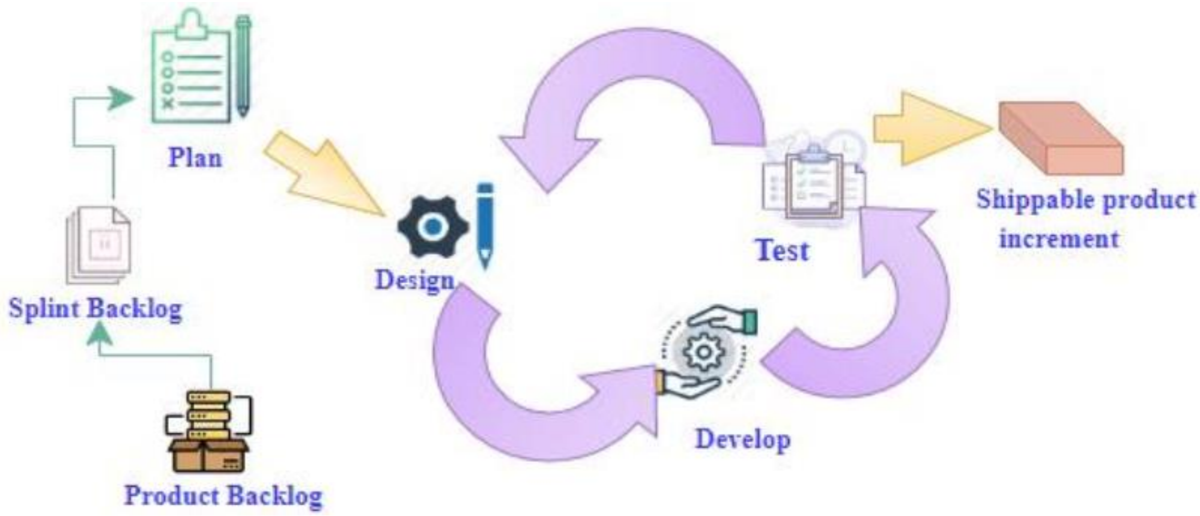


Figure 1: Scrum agile system development lifecycle

Agile development methodology was also used in development of artificial intelligence model. The model is suitable for interactive refine at each step; it saves time and effort and is used in Artificial Intelligence (AI) and machine learning applications. The agile development is shown in in Fig. 2

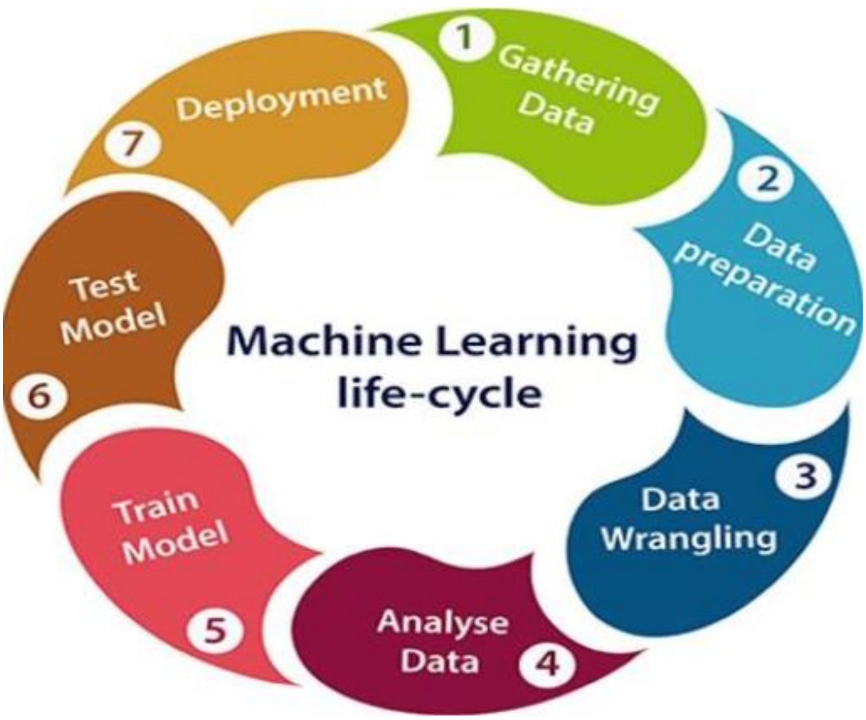


Figure 2: Machine-learning life cycle

Development of machine-learning projects have a cyclical lifecycle process called the machine-learning life cycle. The life cycle's main goal is to find an optimum solution for the project. Figure 2 shows the stages of the life cycle, and Table 2 describes the stages in detail.

Table 2: Machine-learning life cycle

Name	Explanation
Gathering data	To understand the subject necessary data and domain knowledge is gathered through a study.
Data preparation	In this stage, the data set is identified and selected, it is important to understand quality, characteristics and data format required
Data wrangling	During this phase the following is done, cleaning of raw data and transforming them into usable data. Outliers are removed to improve data quality.
Analyze data	In this step, right techniques are chosen based on data acquired from data wrangling
Train the model	In this step, the model is trained on various machine learning algorithms using dataset to comprehend numerous features, laws and pattern.
Test the model	Focus on evaluating the model.
Deployment	The model is applying on real system in the field.

3.1.1 Planning

Each system development starts with planning on how to do requirement gathering and possibly how to design and deploy the product to the target customers. Next, feedback from users is used to plan for the system's development and improvement.

3.2 Study Area

The HWC is common in the northern region of Tanzania because it houses major popular parks like Tarangire, Serengeti, and Ngorongoro with a lot of wildlife that sometimes comes from Kenyan parks. The Tarangire National Park was chosen for this study because it receives significant complaints concerning HWCs which negatively affect the agricultural activities and livelihoods of the adjacent communities who sometimes retaliate by killing the wildlife which impacts the wildlife population, including endangered animals such as the rhino. The Tanzania national parts Authority (TANAPA) has been working to reduce and mitigate these conflicts. Together, the Ngorongoro conservation authority and TANAPA have devised different methods to solve these challenges, including collaring wildlife. The animals that were considered for this study were the elephant, lion, leopard, zebra, and buffalo to test the performance of the machine learning model and its ability to scale up to work on all animals and locals in the park.

3.3 Data Collection Methods

Different data collection methods were used in the present study. The methods include: the collection of images used to train the model and other data collections to establish existing methods and techniques in place to mitigate and report HWCs, Interview to collect views of stakeholders to express their views on how the system should work well to support them in resolving their problems (Bolderston *et al.,* 2012) and observation on the existing system and the environment where the system will work. These data collection methods included the following:

3.3.1 Interviews

A well-structured interview was done with a local village community near Tarangire National Park. The village elder who was responding was interested in a simple tool that can identify and report the wildlife spotted in real-time using simple technology to use such as SMS for planning to respond to the incident.

An interview was done with the human-wildlife response teams and the human-wildlife response team information officer. The officer provided more information about the system in place at the park. These systems included the collaring of wildlife, especially the elephants, and a hotline number that provided important information about animals that escaped the park. The officer needed a low-cost and power-effective system that was accurate in determining the wildlife escaping the park to cause HWCs and the locations where they were spotted.

3.3.2 Observations

Different early-warning systems and other systems that were working in Tarangire National Park officers were observed to determine their effectiveness. These systems included a zonal hotline system in which they receive calls from the neighboring communities about the wild animals that have to escape the park and arrange on how to solve it and collaring where the wildlife collared with tracker and GPS data is monitored from the park office. The other method observed was the use of barriers such as fences and trenches to prevent the movement of wildlife to people's communities and farms.

3.3.3 Image Collection

Images for developing a machine learning model were collected by taking pictures of selected wild animals in the parks and other photos were selected from Tarangire National Park authority album. Three hundred (300) photos of each selected species were gathered. The species of interest were elephants, lions, leopards, zebras, and buffalos. All the photos were preprocessed using labelling, rescaling, and augmenting. The images were sorted and labelled using a labelling tool. Each image was labelled to correspond to six classes of animals: elephants, lions, leopards, buffaloes, and zebras, and its localization in the image. The photos were labelled because the main objective was to detect and classify the animals under the supervised machine learning category whose input and output are known. The wildlife images that majorly caused these conflicts were collected to be used for training the camera traps model.

3.4 Requirements Gathering and Analysis

Data collected at this stage was summarized into definition of the system requirements. The data collected during the interview, observation and image collection had shown that the system should use simple technology that can provide information in advance as the wildlife escapes the park. The system was supposed to be low-cost, effective and accurate. Therefore, the system developers settle to come up with a camera trap since it had most of the qualities that were demanded by the user requirements which were real-time and its ability to send SMS and files data to the web server for visualization.

3.4.1 Software Requirements

The system requirements were dived into hardware requirements and software requirements. The software requirements needed to develop the system is as listed in the Table 3.

Table 3: Software requirements

S/N	Item name	Functionality
1.	Draw IO	Drawing Diagrams
2.	Raspberry Pi Imager	For OS installation in the Raspberry Pi
3.	Raspberry Pi OS	Operating system that used to configure Pi
4.	Notepad++	Writing codes for a website
5.	Jupyter notebook	For developing deep learning model for the system.
6.	YOLOV3	
7.	openCV	
8.	Python	
9.	Hypertext Preprocessor	It the server-side scripting language used to develop a website.
10.	Hypertext markup language	It is markup language used to develop websites.
11.	Apache webserver	It is server used to host and run a website.
12.	Database server	It is server whose functionality is to host database.

3.4.2 Hardware Components

The developed system uses a Raspberry pi board, Raspberry pi camera, PIR sensor, wireless modem, and IoT cloud database. The PIR sensor detects nearby wildlife and sends a signal to Raspberry pi to activate the camera to take the picture. After capturing the image, internal analysis is performed to detect and identify the objects used to decide the importance of sending a message about identified wildlife to relevant personnel. Finally, the data is sent to cloud platforms to allow real-time notifications.

(i) Passive Infrared Sensor

It is a widely used inexpensive motion sensor to detect motion (Frankiewicz *et al.,* 2013). Due to this feature of being passive is undetectable, consumes less power, and works well in all conditions including dark environments, which has led it to popularity (Sahoo *et al.,* 2017). The PIR sensor is a sensor that uses infrared heat signatures for detection (Saranu *et al.,* 2018). The advantage of this sensor is that it is small and easy to use. The PIR sensor senses the infrared signals from an object before activating the camera to take an image. The PIR used was as shown in Fig. 3.

Figure 3: Passive infrared sensor

(ii) Camera

The Raspberry pi camera module v2 is a high-quality camera with 8MP. It has a fixed focus feature. It is capable of taking 3280*2464 pixel still images but can support 1080p30. This camera is supported by all versions of the raspbian operating system and can capture both high-definition video and pictures (Rao *et al.*, 2020). The Raspberry pi camera is used to take still pictures and videos. The camera has holding holes for simple installation as shown in Fig. 4.

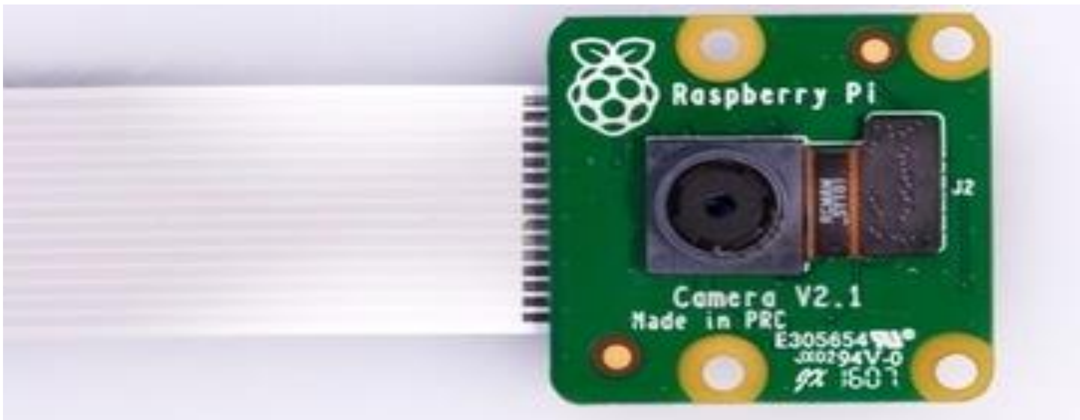


Figure 4: Raspberry Pi camera

(iii) The battery

A 12v solar battery (Fig. 5) since other systems in the park was using them to reduce maintenance logistics and support.

Figure 5: Rechargeable solar battery

(iv) The solar charge controller

A charge controller is used to convert 12 V from the battery to 5 V to be used by the Raspberry pi. The charge controller is suitable because it controls the flow of power from the solar to the battery and from the battery to the Raspberry Pi. Different charge controllers exist in the market and the one used was a charge controller for 12 V solar and 12 V battery with Raspberry Pi of 5 V. The charge controller is shown in Fig. 6.

Figure 6: Solar charge controllers

(v) The Global Positioning System sensor

It is a sensor used to detect the object's location. It accurately tracks the object's longitude and latitude as it moves (Listiyana, 2018). The GPS used is shown in Fig. 7, it works well with Raspberry Pi.

Figure 7: Global positioning system module

(vi) Internet modem

The modem supports the global system for mobile communication (GSM) because it supports sending of image SMS and supports data connection to send data to the cloud as shown in Fig. 8. The GSM is a mobile communication modem that operates in the 850 MHz, 900 MHz, 1800 MHz, and 1900 MHz frequency bands. It can perform operations like sending and receiving texts and making voice calls. It is interfaced with the Raspberry pi to send text messages using AT commands. According to the datasheet, the modem module requires a power supply of 4.5-6 V and a minimum current of 250 mA, and a maximum of 590mA (Sahoo *et al.*, 2017).

Figure 8: Modem module

(vii) Raspberry Pi

Raspberry pi was used as the main processing unit in the system since it provided simpler and affordable processing solutions. Raspberry pi 4 8 GB RAM processor speed is 1.2 GHz was used shown in Fig. 9. Its low-cost and low-power microcomputer were primary factors in choosing it (Bekaroo *et al.*, 2016). It is a tool that brings IoT programming to communities, enabling IoT devices for recreation, education, and personalized systems. Raspberry pi 4 comes with different designs and layouts. It supports various simple high-level programming languages such as python that provides more flexibility than other microcontrollers, hence suitable for this project. Raspberry pi supports different processes such as multi-threading which allows it to be a better tool in processing machine learning at the edge processes and activities. Therefore, it is suitable to reduce the amount of data sent to the server by sending only essential data from the edge.

Figure 9: Raspberry pi

3.5 Block Diagram

The system components were configured to the raspberry pi development board, a single-board microcomputer shown in Fig. 10. All the components were connected to the Raspberry pi. Those components include PIR sensors, a global positioning system (GPS) module, a Raspberry pi camera, and a GSM module.

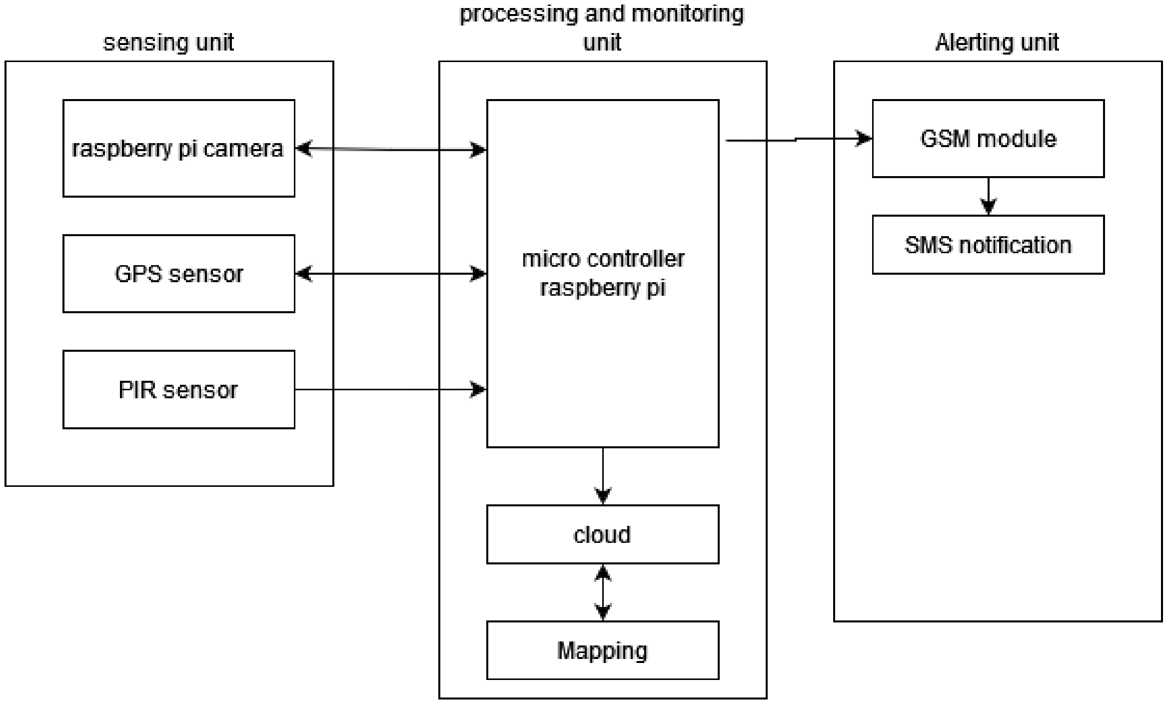


Figure 10: System block diagram

3.6 Flowchart

The system was developed using a python script to control all sensing components in collecting data until the alerting unit when data is sent through SMS and online update for visualizations. The processing module processes data using the YOLO deep learning algorithm packaged into the TensorFlow lite file. After the object has been detected and identified, the alerting unit sends data through SMS and updates online server for google map visualizations as shown in Fig. 11 shows the steps in which all actions are taken from the start of the system until it ends.

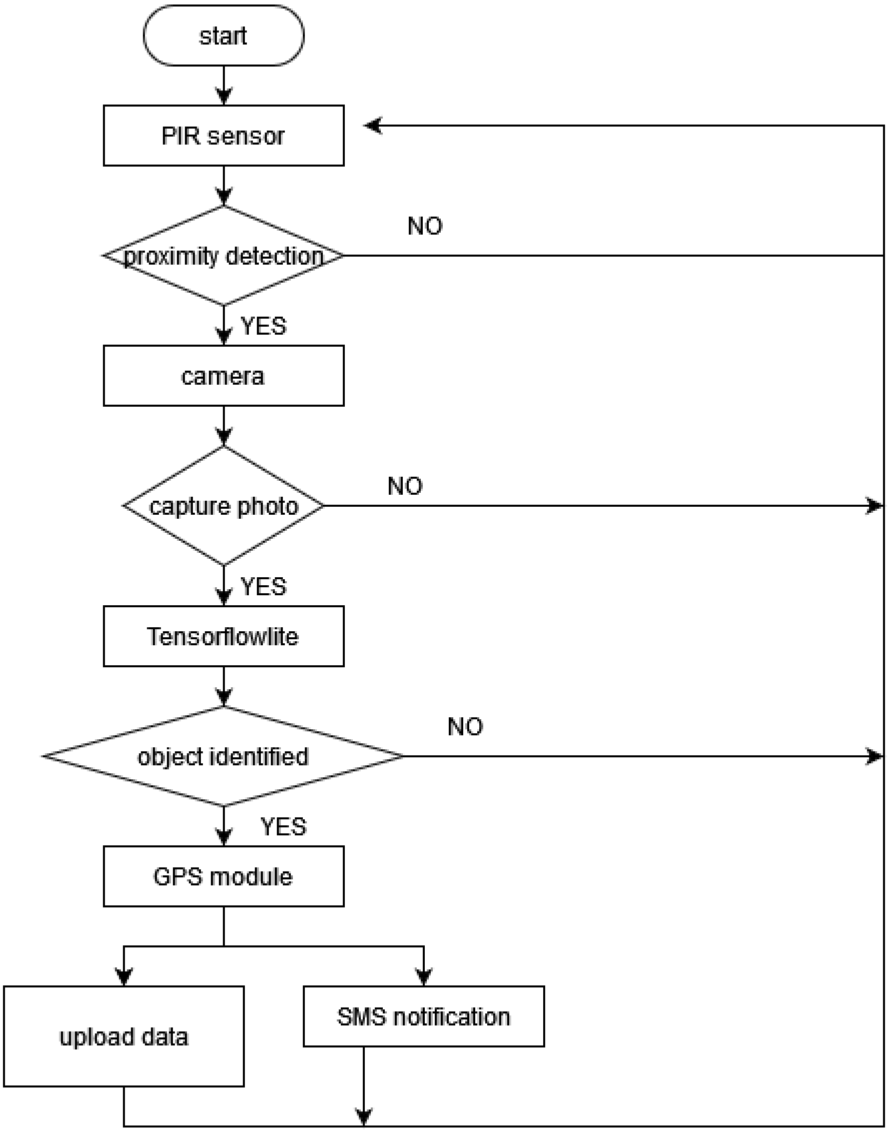


Figure 11: Flowchart

In the first stage, the PIR sensor senses the availability of wildlife's heat signatures near it and starts the program that switches on the Raspberry Pi camera and captures the image. The processing part uses TensorFlow lite to determine the object in the image, if an animal of interest is detected, the GPS module is activated to give device current location. Then information from the GPS module on location and the wild animal's name received from the processing unit is preprocessed and made ready for sending as SMS and forwarded to the modem or GPRS module to send SMS to selected numbers. The same message is packaged and sent to the online server to provide data for google map visualization and monitoring from the park's authority offices as shown in Fig. 11.

3.7 System Overview

3.7.1 System Architecture and Design

The system was designed and developed into three stages: sensing, processing, alerting, and visualization as shown in Fig. 12. The sensor module comprises the PIR sensor, Raspberry pi camera, GPS module, and GSM module, and Raspberry pi is the main processing unit handling all processes including data management until sending them to a server which keeps and processes data for visualization and mapping. The PIR sensor senses the wildlife's heat signatures near it and starts the program that switches on the Raspberry Pi camera and captures the image. The process shown in Fig. 13 is from preprocess the image, extract features, detect, identify and classify the images to identify wildlife of interest, and then send data to the server and SMS to HWCRT. The data in the server shall be used to monitor and visualize the location of each identified wild animal on the map.

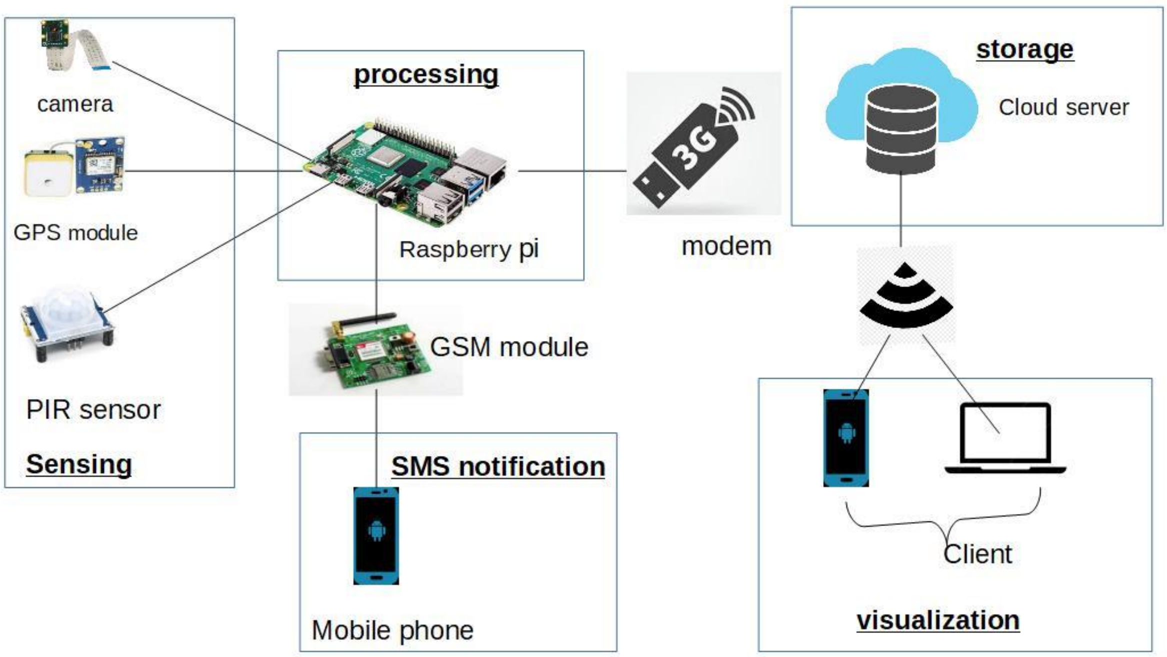


Figure 12: Proposed system architecture

3.7.2 Sensing Unit

The central sensing unit is a PIR sensor, Raspberry pi camera, and GPS module. The PIR sensor was used to detect wild animal in its proximity. The PIR uses heat signatures (Saranu *et al.,* 2018) to achieve its purpose, thus can work under all lightness conditions (Sahoo *et al.*, 2017), while the Raspberry pi camera (Rao *et al.*, 2020) main purpose is to take pictures after its activation following the sense of wild animal. The camera is the foremost essential component that collects images from the environment in which it is focused after activation by the PIR sensor. The GPS module is used to accurately estimate the system's current location using GPS coordinates (Listiyana, 2018). The GPS data is used for mapping and further analysis.

3.7.3 The Processing Unit

The Raspberry pi uses a python script to extract information from the image taken using the tensors flow lite model. It was used to examine the image taken by the Raspberry pi camera to detect objects in the picture and identify the available wildlife. It is a low-power-consuming microcomputer (Bekaroo *et al.*, 2016). The details of detected wildlife are coupled with GPS location information and sent to the cloud for visualization. The identified wildlife data details in the image are combined with the location data details from GPS and then transmitted to the cloud for visualization. Finally, the wild animals identified were used to determine if the SMS is sent to the HWCs response team using GSM. The cloud server will be storing the data collected by sensors processed by Raspberry Pi for visualization and future references.

3.7.4 Visualization and Alerting

The system used google Maps to visualize data reported from the nodes containing GPS coordinates, name of the wildlife, node id, and time received. The HWCRT allocates action to each incident received. The task issued includes assigning one team to check the incident or ignoring the incident if one team is in the area or already allocated. The SMS sending job is ignored if the wildlife cannot cause conflict or is not an endangered species. The command centre receives field outcome reports and files them into the system. After finishing the assigned field activity, its outcome is recorded and compiled into reports.

3.7.5 Deep Learning

Computer vision is a crucial area of study in artificial intelligence that gives machines the ability to recognize and detect objects. Region-Based detection and Regression Object Detection are two categories of object detection techniques. The three stages of traditional object detection algorithms are proposed region selection in the graphic using bounding boxes, feature extraction, and classification using a trained classifier. They thus perform better in terms of object detecting precision but operate slowly. There are three types of region-based algorithms: RCNN, fast RCNN, and faster RCNN. While Regression Object Detection techniques like SSD and YOLO create a region proposal network and simultaneously classify the region. The YOLO technique is ideal when the object placement is unpredictable in the field of view since it is quick and accurate at detecting small things.

3.7.6 Image Preprocessing

The photograph collected were scaled to 214 x 214 before feeding them to the model. Augmentation is a method for generating new images by a random transformation of existing images. It reduces the chances of model overfitting. Augmentation can create new data up to 50 times existing one hence a powerful tool. It involves variety of transformations like rotation, padding, saturate, greyscale, center crop, flipping, and many others.

3.7.7 Proposed Model

The YOLO model usually divides an image into grids and then do prediction to determine the confidence and location of the object in the image. The next step is to predict the bounding box to find the truth box using intersection-over-union. Finally, the final results were determined using the bounding box’s threshold is determined. The architecture used was darknet-53. ReLU activation, a batch size of 512, and 40 epochs was used in training of the model.

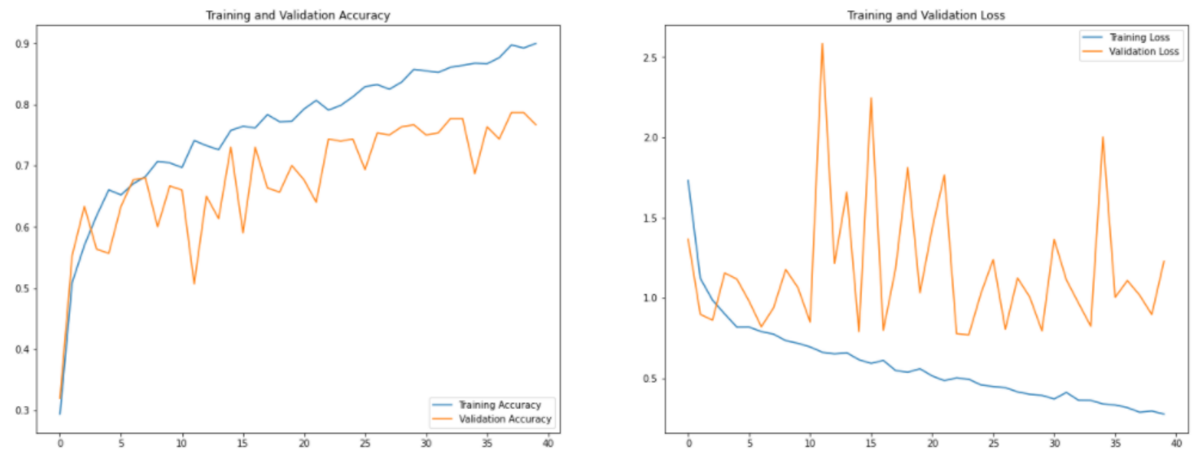


Figure 13: Training and validation plots for the model

The input image size was 214 x 214 x 3 pixels and after 3 pooling and 6 convolutions, the size was shrunk to 21x 21 x128 pixels for the output features maps. Further convolutions produce a neuron of 1 x 1 x 4096 (Alsheikhy *et al.*, 2020). As a result, the model achieved a training accuracy of 90% and a validation accuracy of 78, and the loss graph is shown in Fig. 14.

3.7.8 TinyML Model

With the help of the YOLO model and 1500 pictures of notable species known to cause HWCs, a deep-learning model was developed. The model was trained using these photos and achieved a 74% accuracy utilizing the low-power Raspberry Pi microcontroller. As a result, it could identify six important species in the park. After collecting the images for the developing model, the images were sorted and labeled well so that the data were clean to achieve better object detection of the trained model. The data used to develop the model was divided into 70-20-10, 70% training data set, 20% validation dataset, and 10% testing dataset. The training was done in the Kaggle environment using the python language and cloud Tensor Processing Units (TPU) accelerator to develop TensorFlow. The model was created using a normal stochastic descent gradient optimizer for minimizing error. Since the problem was single-class classification ReLU activation was used. The trained model was converted to TensorFlow lite using the TensorFlow library so that it can be loaded by Raspberry pi and exported to the TensorFlow Lite file.

3.7.9 Deep Learning Model Testing

The YOLO model achieved 98% accuracy in detecting wildlife as shown in Table 4 when tested using data from the park. The model identified the wildlife of interest with a reasonable accuracy that could be worth identifying and reporting.

Table 4: Model testing results

	Buffalo	Elephant	Leopard	Lion	Rhino	Zebra	Uncertain
Buffalo	100	0	0	0	0	0	0
Elephant	1.7	94.8	0	0	3.4	0	0
Leopard	0	0	100	0	0	0	0
Lion	0	0	0	100	0	0	0
Rhino	0	3.1	0	0	95.3	0	1.6
Zebra	0	1.1	0	0	0	98.9	0
F1 score	0.99	0.95	1.00	1.00	0.96	0.99	

3.8 System Testing

In the lab, the system was tested by directing the camera to an image of wild animal on the computer monitor and then swapping it with different photograph of objects and wildlife. Next, it identified the wildlife and sent data to the cloud for visualization using google maps.

3.8.1 Unit Testing

Unit testing is a method used to verify functional correctness of each component and unit of the system (Dybå *et al.*, 2008). For this developed prototype system, the following unit were tested: a GPS sensor, a passive infrared (PIR) sensor, and a Raspberry Pi camera, to collect data about the environment, a Raspberry Pi equipped with TensorFlow Lite mode to perform object detection in the image and identify the wild animal in the image and GSM to send SMS to mobile officers who are using a phone.

3.8.2 System Testing

System testing involves testing the entire system as a unified entity to verify if it functions as intended and fulfills the desired requirements of the end users. It examines the collective performance of all modules during this phase. Unlike code-level scrutiny, system testing focuses on assessing the visible functional accuracy of the final product.

During the interfacing process, the hardware was connected to an online cloud database to facilitate data storage and virtualization. Subsequently, the entire system underwent testing, and it demonstrated successful operation in alignment with the predetermined objectives.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results

To improve the mode and system used in early warning of human wildlife conflicts, a camera trap was developed with the support of online server. The communication between a camera trap and server was done GPRS models to ensure each camera was independent.

4.1.1 Results from Requirements Gatherings

This report outlines the results of a project with the main objective of developing an early warning system for human-wildlife conflict (HWC) using IoT, SMS, and mapping. The specific objectives included analyzing system requirements, designing and developing the system, and validating and implementing it in Tarangire National Park.

Through engagement with villages and park wardens, it was determined that a simple camera trap was the preferred tool for reporting wildlife escaping the park, and the HWC response team suggested a real-time tool to identify and report wildlife with location data.

The system requirements were defined based on data collected through interviews, observations, and image collection. The system needed to use simple technology, be low-cost, effective, and accurate. Therefore, a camera trap was chosen as it met most of the user requirements, including real-time detection and the ability to send SMS and file data to the web server for visualization.

4.1.2 Functional Requirements

The functional requirements for the proposed system are summarized in Table 5 and include automatically sensing approaching wild animals, taking pictures of focused objects and location coordinates, using deep learning to determine wildlife in images, sending SMS with location and animal identification data, using Google Maps to show the location of the animal, and generating reports of all HWC incidences on the server.

Table 5: Functional requirements for the proposed system

S/N	Requirements	Descriptions
1.	The system should automatically sense approaching wild animal.	The system should be able to detect wild animal that has come near to the system.
2.	The designed system to take picture of focused object and take location coordinates.	The system to activate camera to take a picture and also GPS sensor to collect location coordinates details.
3.	Use deep learning to determine wildlife in the image.	The system should be able to detect and identify wild animal sported.
4.	Use SMS to send data of location and the name of animal identified.	The system should be able to send SMS upon detecting and identifying the wild animal
5.	Use google map to show location where the animal was sported.	The system should be able to show map of locations where wild animals were sported.
6.	To generate report of the animals populated on the server.	The system should be able to populate all data of HWC incidences and report them in google map for better visualization.

4.1.3 Non-Functional Requirements

Non-functional requirements are more concerned with quality and how the system should operate to please the end user than they are with the core functions of the system or what the program should do. The non-functional criteria for the suggested system are shown in Table 6.

Table 6: Non-functional requirements

Description	Requirement description
Performances	to be able to identify and report wildlife within a period of less than 1 minute
Robustness and Reliability	the system should be able to run all time when it is required
Usability	the system should have an easy to interface that can be learn to use easily

4.1.4 Results of Prototype Design and Development

The system proposed in this project was composed of three main parts: sensing, processing, and notification. The sensing part consisted of three sensors: a passive infrared (PIR) sensor, a Raspberry Pi camera, and a GPS sensor, which were used to gather data about the wildlife and their location. The processing part was performed by a Raspberry Pi, which was equipped with TensorFlow Lite to identify the wild animals in the images captured by the camera. The notification part used GSM to send SMS alerts to mobile officers and an API from an SMS-enabled server to send data to the web server for monitoring and visualization. The system prototype during development and as finishing development is shown in Figs. 14 and 15.

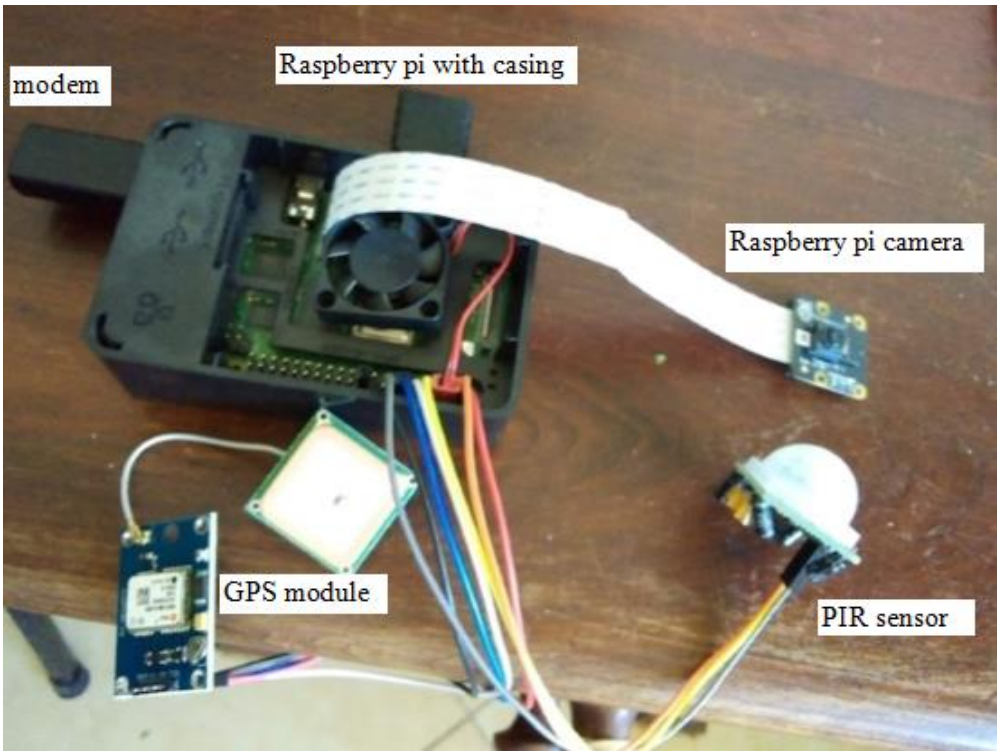


Figure 14: System prototype development

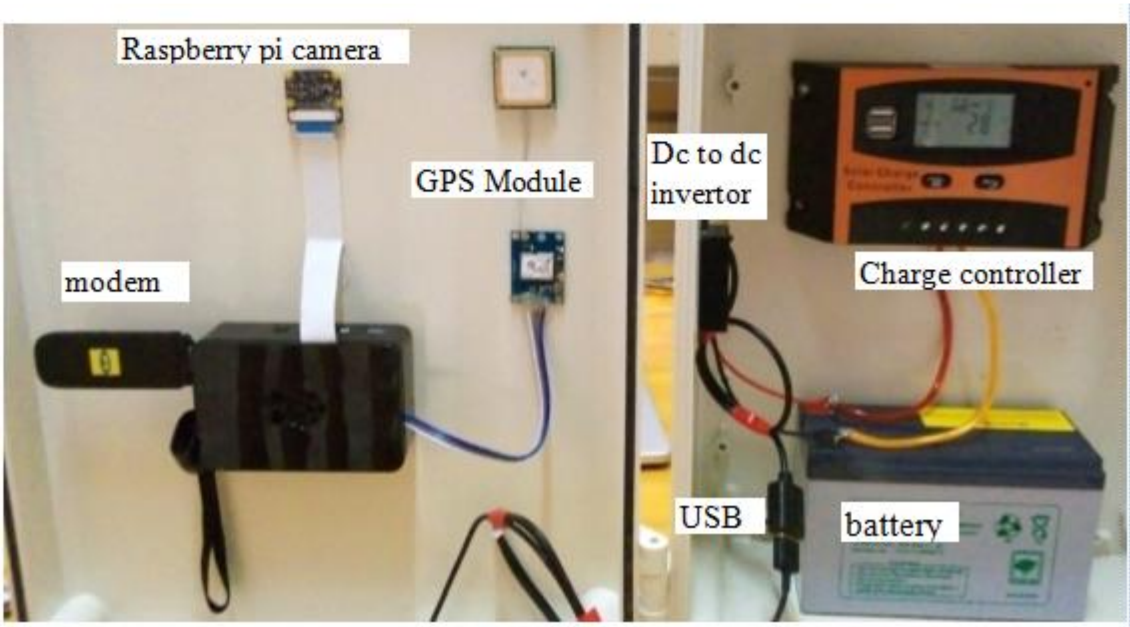


Figure 15: System prototype

4.1.5 Results of Prototype Development

The system was implemented on a Raspberry Pi running a Linux environment and a Python script that executed machine learning models from the TensorFlow Lite file to process the images. The identified images were then used to activate the GPS module to collect location data. The SMS notification system was used for fast and simple communication, while a web system provided visualization and reporting of the data and maps of the escaped animals.

The notification part uses an API from the SMS enable server capable of receiving a request and sending SMS over a telecommunication network through a special registered sender ID. To use this service a developer needs to use a third-party API provided by a company that renders this service such as which was used in this project.

The SMS notification was used as an easy and simple way of communicating the escape of animals but visualization and report were accessible through a web system containing data and maps of the wild animals reported.

The system was developed to work in the park setting whereby many camera traps were placed on the animal part so that it can economize the camera trap used while maximizing the opportunity of detecting and reporting most animals escaping the park as shown in Fig. 16.

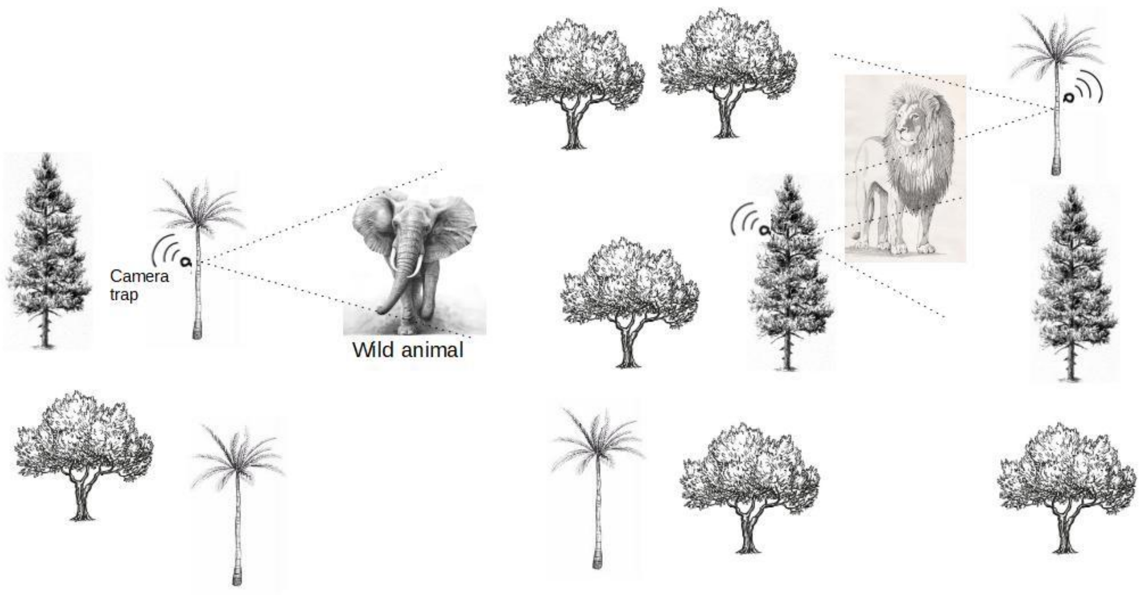


Figure 16: System field architecture

System specifications include hardware and software requirements. The project should be successful and should be implemented as proposed following the steps in sections 4.1.1 to 4.1.5.

4.1.6 Sensing Subsystem

The sensing subsystem was correctly interfaced to the Raspberry pi and it functioned well to produce desired results as proposed in the system. The PIR sensor and GPS sensor are connected to serial ports which Raspberry Pi camera is connected to the Raspberry Pi camera port. The most crucial module was the PIR sensor which initializes all processes after detecting the presence of the wildlife within its vicinity. The Raspberry pi camera took a picture after it is activated. The other sensor was a GPS sensor that determined the current location of the devices, therefore showing the location of the data captured.

4.1.7 The Processing Unit and Deep Learning at the Edge

The Raspberry pi equipped with the YOLO model was used to analyze the image taken by the raspberry pi camera to detect objects in the picture and identify the available wildlife. It is a low-power consuming microcontroller (Bekaroo *et al.,* 2016). The reason for choosing Raspberry pi was its low power consumption with a reasonable processing speed. After identifying the wildlife in the image, these data and the location information were sent using SMS to indented persons to visualize using Google Maps from the link sent (Javed *et al.*, 2020).

4.1.8 Deep Learning Model

After collecting the images for the developing model, the images were sorted and labelled well so that the data were clean to achieve better object detection of the trained model. The data used to develop the model was divided into 70-20-10, 70% training data set, 20% validation dataset, and 10% testing dataset. The training was done in the Kaggle environment using python language using cloud TPU accelerator to develop tensors flow. The model was created using a normal stochastic descent gradient for minimizing error. Since the problem was single class classification ReLU activation was used. The training model was converted to tensor flow lite that was able to use in Raspberry pi and exported to the TensorFlow Lite file.

Table 7: Validation accuracy

	Buffalo	Elephant	Leopard	Lion	Rhino	Zebra
Buffalo	91.3	2.2	0	0	6.5	0
Elephant	10.9	73.4	0	0	14.1	1.6
Leopard	0	0	100	0	0	0
Lion	20	0	0	60	20	0
Rhino	5.7	0	0	0	94.3	0
Zebra	0	0	0	0	0	97.8
F1 Score	0.85	0.84	1.00	0.75	0.85	0.98

The performance of the model was evaluated based on the training dataset. The metric used for training and validation was accuracy as shown in Table 7. The model was able to detect and identify wildlife with remake accuracy. It detected different wildlife like elephants, lions, zebra, and buffalos as shown in Fig. 17 (a), (b), (c) and (d) respectively.

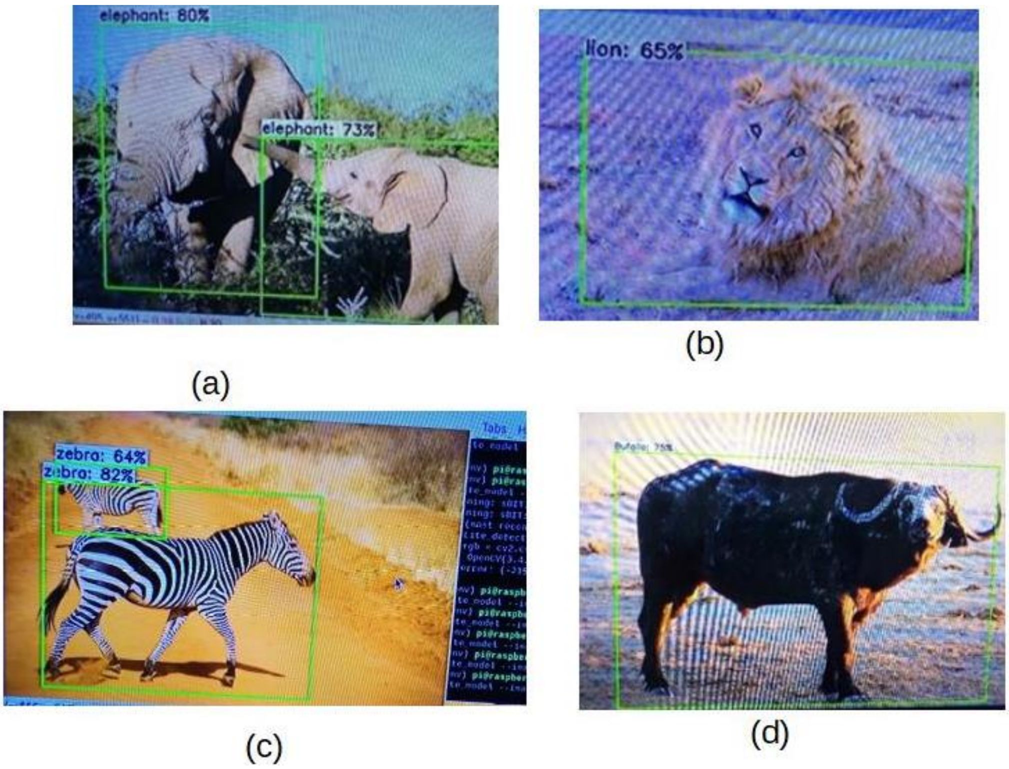


Figure 17: Model object detections and identification results

4.1.9 Notification

The system has a GSM module that sends an SMS message to the interest parties when the wildlife of interest is detected. The danger determines the interest parties that posed the wildlife to the communities around the park. If the animal was not dangerous or not endangered wildlife, then no notification was made. For dangerous animals like crop-raiding wildlife such as elephants, notification was sent to the game park rangers and the community elders. If the animal is not dangerous but on the endangered list such as the zebra, the report was conveyed to the game park rangers only. The sample of the message is shown in Fig. 18 showing the wildlife reported, the location using GPS coordinates, and the universal resource locator (URL) of maps for the GPS coordinates in google maps.

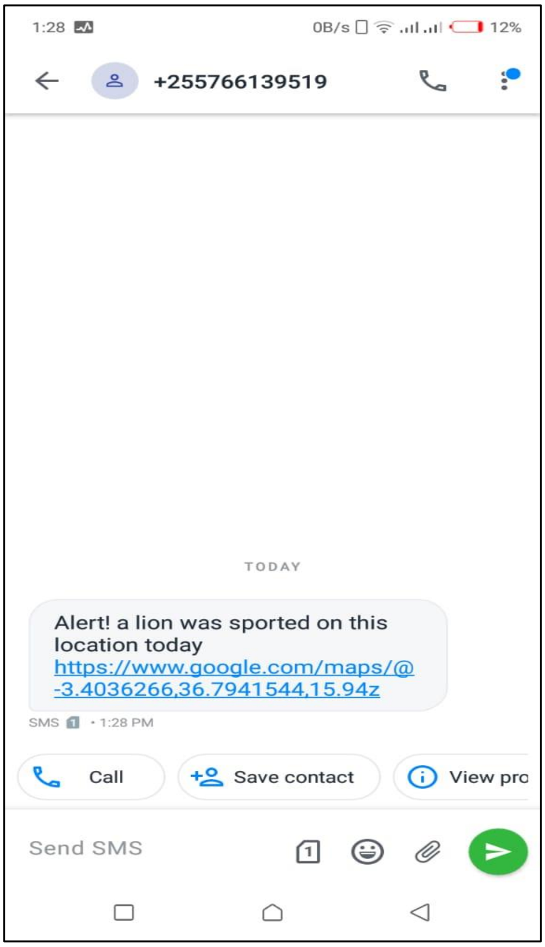


Figure 18: Testing SMS receipt

4.1.10 Web-Based Monitoring

The web-based system was developed primarily for sensor node data management and monitoring, user management, dashboard reports, and monitoring reports. It is accessible by use of a web browser. Park rangers and park managers are the only privileged people to use the system to access the park's wildlife spotting data from the database. Fig. 19 shows part of the web-based monitoring system with the location where animals were sported and the time they were sported. The website is used to manage HWC in coordinating, resolving, and preparing reports.

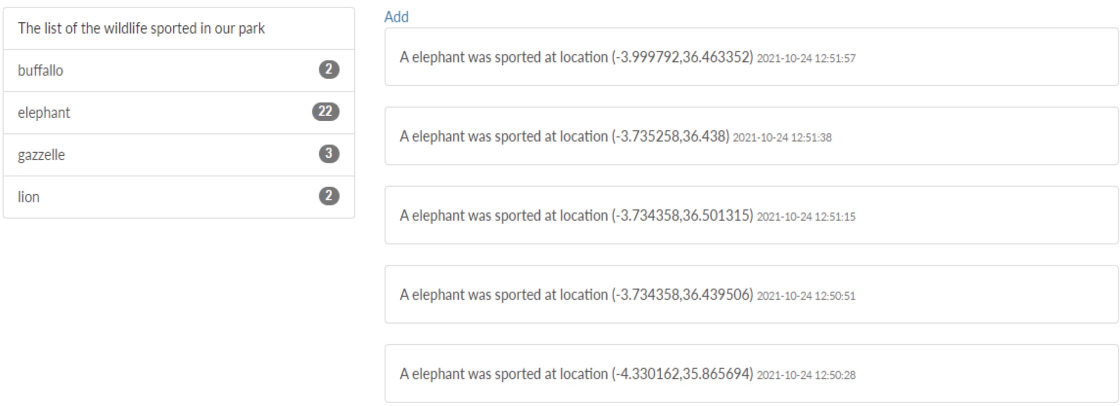


Figure 19: Web monitoring system

4.1.11 Mapping

An online web system was developed, which takes data from different camera traps indicating wildlife identified, the time at which it was identified, and the location where it was sighted. The system updated data to an online database. The data in the system was populated such that the game park officer will populate them into a map as shown in Fig. 20. The map had two parts to monitor wildlife spotted in the park and the other part where the wildlife is sported moving to the edge of the parks. A part of the map tab involved just wildlife moving to the edge of the parks and that could cause HWCs

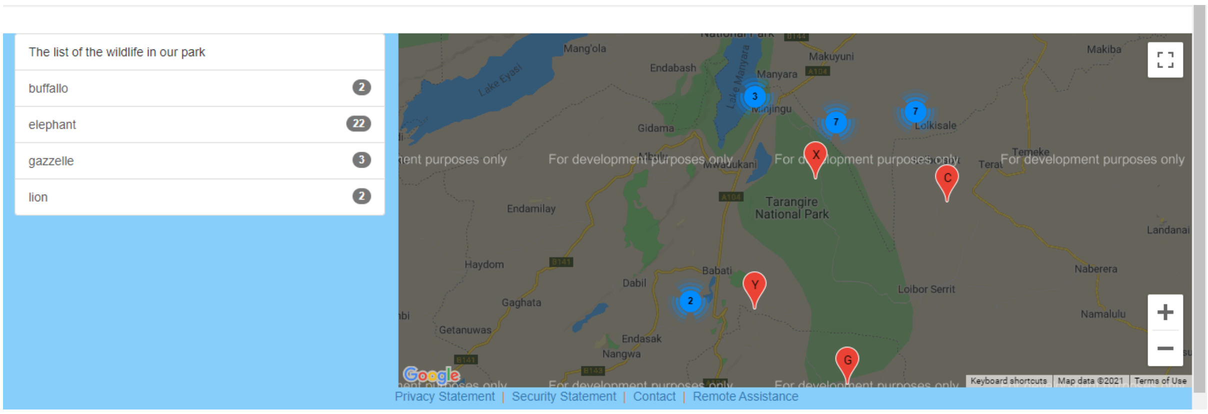


Figure 20: Mapping system of the wildlife

4.1.12 Results from System Testing and Validation

The system testing and validation stage was an essential step to ensure that the developed system was functional, efficient, and effective. The system testing and validation process involved several stages.

First the hardware components of the system were tested to ensure that they were functioning correctly. This involved checking the PIR sensor, Raspberry Pi camera, GPS sensor, and GSM module for any defects or malfunctions. The hardware components worked well to provide desired output.

The next was testing software components of the system were tested to ensure that they were working correctly. This included testing the machine learning models, the SMS notification system, and the web server for data visualization.

The different components of the system were integrated and tested to ensure that they were working together correctly. This included testing the communication between the PIR sensor, Raspberry Pi camera, and GPS sensor, as well as the communication between the Raspberry Pi and the GSM module.

The system was tested in the field to validate its effectiveness in detecting and reporting wildlife escaping the park. The camera traps were set up in various locations in Tarangire National Park, and the system was left to run for several days. The data collected during this testing phase were used to evaluate the system's accuracy and effectiveness. During the testing, the system was able to detect the wildlife approaching the camera trap and capture their images with the Raspberry Pi camera. The image captured was processed using TensorFlow lite model for object detection, which enabled the system to identify the wild animal in the image accurately. The GPS sensor was also able to collect the current location data of the detected animal.

The notification part of the system was also tested, and it was found to be efficient in sending SMS notifications to the mobile officers and sending collected data to the storage server for monitoring and visualization. The SMS notification was received within a few seconds after the detection of the animal, which indicated that the system was reliable in real-time reporting.

The system was also evaluated for its non-functional requirements, such as performance, robustness, reliability, and usability. The system's performance was satisfactory, as it was able to identify and report wildlife within less than a minute. The system was found to be robust and reliable, as it was able to run continuously without any issues during the testing period. The system's usability was also evaluated, and it was found to have an easy-to-use interface that could be learned quickly by the users.

4.1.13 User Acceptance Results

The system was tested by end-users, including park wardens and community members, to evaluate its usability and ease of use. Feedback from the users was used to improve the system's design and functionality. The results of user acceptance as shown in Table 8.

Table 8: User acceptance results

Validations features	Strong agree	agree	Not sure	disagree	Strong disagree
The system was easy to use and understand, and users were able to effectively navigate and interact with its various components.	60%	20%	3%	7%	10%
Users appreciated the real-time functionality of the system, and find that it provided valuable information about wildlife escaping the park.	78%	9%	6%	4	3
The system accurately detects and identifies wildlife in images, and users finds that it was reliable and consistent in its performance.					
Users were able to quickly and easily access data and maps through the web system, and find that this information was helpful in addressing human-wildlife conflict (HWC) incidents.	85%	5%	0	4%	6%
The system meets the needs of both park wardens and local communities, and is seen as an effective tool for preventing HWC incidents and promoting coexistence between humans and wildlife.	76%	4%	10%	5%	5%

4.2 Discussion

The implementation of this project is indeed a significant contribution to solving the problem of human-wildlife conflict and protecting endangered species. The use of low-power microcomputers such as Raspberry pi makes the system affordable and easily deployable in remote areas with limited power supply. The integration of various sensors such as PIR sensor, GPS sensor, and Raspberry pi camera enables the system to collect real-time data about the environment and wildlife, process the data using machine learning algorithms, and notify the relevant authorities of any detected wildlife.

The system's ability to detect and identify wildlife in real-time and notify the relevant authorities through SMS is a significant improvement over traditional wildlife monitoring methods, which are often time-consuming and require significant resources. The web application dashboard also provides a useful tool for tracking and visualizing wildlife movements, allowing authorities to take appropriate measures to prevent human-wildlife conflicts.

However, there are still some limitations to the system that need to be addressed. For example, the system's accuracy in detecting and identifying wildlife may be affected by environmental factors such as weather conditions and lighting. The system's ability to operate in remote areas with limited internet connectivity also needs to be improved to ensure reliable data transmission.

In conclusion, the implementation of this project demonstrates the potential of deep learning and edge computing to solve real-world problems such as human-wildlife conflict and wildlife conservation. Further research and development in this area could lead to more effective and efficient wildlife monitoring and management solutions that can help protect both human and animal populations.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The camera trap could send data to the server after automatically identifying the species and report the correct notification to the relevant people in case of species of interest. They automatically identified the species in the question, therefore, reducing the need for a person to follow up and identify species in the images manually. Since object detection and identification on the server can cause a delay due to network issues and processing issues, object detection on the edge is the best way.

5.2 Recommendations

wildlife monitoring and reporting of HWC requires a model with high performance to detect and identify an object of different wildlife with high precision and accuracy to prevent miss identification of the wildlife. Due to limited time and scope, the system was developed to identify four wildlife only.

The camera traps have made the process of monitoring wildlife and warning of potential HWC easier, safer, and more real-time. This process will always need an accurate process to ensure that the required species triggers the camera to avoid missed images. The other problem to be overcome by the camera trap is to be energy efficient, save power by switching off the camera and other unnecessary components during the sleeping mode, and be fast enough when required to wake up and facilitate taking pictures or video.

The battery used for the system was a simple recharge battery that works short period of a few days; therefore, the system should be improved to include a solar system and a high-capacity battery to run for a long time. The accuracy of a sensor distance was just about a hundred metres within line of sight and to improve the performance of the system sensor having higher power should be used.

The web and visualization system needs improvement to better integrate with other of the Tarangire National Park web systems to enable faster learning of using the system and seamless operation of the park.

REFERENCES

Alsheikhy, A., Said, Y., & Barr, M. (2020). Logo Recognition with the Use of Deep Convolutional Neural Networks. *Engineering, Technology & Applied Science Research, 10*(5), 6191-6194.

Bekaroo, G., & Santokhee, A. (2016). Power consumption of the Raspberry Pi: A comparative analysis. *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies,* 361-366.

Benjaminsen, T. A., Goldman, M. J., Minwary, M. Y., & Maganga, F. P. (2013). Wildlife management in Tanzania: State control, rent seeking and community resistance. *Development and Change, 44*(5), 1087-1109.

Bolderston, A. (2012). Conducting a research interview. *Journal of Medical Imaging and Radiation Sciences*, *43*(1), 66-76.

Branco, P. S., Merkle, J. A., Pringle, R. M., King, L., Tindall, T., Stalmans, M., & Long, R. A. (2020). An experimental test of community-based strategies for mitigating human–wildlife conflict around protected areas. *Conservation Letters, 13*(1), e12679.

Broekhuis, F., Cushman, S. A., & Elliot, N. B. (2017). Identification of human–carnivore conflict hotspots to prioritize mitigation efforts. *Ecology and Evolution*, *7*(24), 10630-10639.

Caro, T., & Davenport, T. R. (2016). Wildlife and wildlife management in Tanzania. *Conservation Biology, 30*(4), 716-723.

Chongela, J. (2015). Contribution of agriculture sector to the Tanzanian economy. *American Journal of Research Communication, 3*(7), 57-70.

Connor, T., Division, W., Tripp, E., Bean, W. T., Saxon, B. J., Camarena, J., Donahue, A., Sarna-Wojcicki, D., Macaulay, L., Tripp, W., & Brashares, J. (2022). Estimating Wildlife Density as a Function of Environmental Heterogeneity Using Unmarked Data. *Remote Sensing, 14*(5), 1087.

Divya, U. K., & Praveen, M. (2018). IoT-based wild animal intrusion detection system. *International Journal on Recent and Innovation Trends in Computing and Communication, 6*(7), 06-08.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology, 50*(9-10), 833-859.

Fazil, M., & Firdhous, M. (2018). IoT-enabled smart elephant detection system for combating human elephant conflict. *2018 3rd International Conference on Information Technology Research*, 1-6.

Frankiewicz, A., & Cupek, R. (2013). Smart passive infrared sensor-Hardware platform. *2013 39th Annual Conference of the IEEE Industrial Electronics Society*, 7543-7547.

Ghosh, A., Al Mahmud, S. A., Uday, T. I. R., & Farid, D. M. (2020). Assistive technology for visually impaired using tensor flow object detection in raspberry pi and coral USB accelerator. *2020 IEEE Region 10 Symposium*, 186-189.

Giefer, M., & An, L. (2020). Synthesizing Remote Sensing and Biophysical Measures to Evaluate Human–wildlife Conflicts: The Case of Wild Boar Crop Raiding in Rural China. *Remote Sensing, 12*(4), 618.

Giordano, S., Seitanidis, I., Ojo, M., Adami, D., & Vignoli, F. (2018). IoT solutions for crop protection against wild animal attacks. *2018 IEEE International Conference On Environmental Engineering*, 1-5.

Graham, M. D., Adams, W. M., & Kahiro, G. N. (2012). Mobile phone communication in effective human elephant–conflict management in Laikipia County, Kenya. *Oryx, 46*(1), 137-144.

Javed, S., Ghazala, S., & Faseeha, U. (2020). Perspectives of Heat Stroke Shield: An IoT based Solution for the Detection and Preliminary Treatment of Heat Stroke. *Engineering, Technology & Applied Science Research, 10*(2), 5576-5580.

Karidozo, M., & Osborn, F. V. (2015). Community based conflict mitigation trials: Results of field tests of chilli as an elephant deterrent. *Journal of Biodiversity & Endangered Species, 3*(1), 144.

Khan, F., Ghatge, G., & Hemaya, S. (2019). Object detection and human identification using raspberry pi. *2019 1st International Conference on Advances in Information Technology*, 135-139.

Kiruthika, M., & Bindu, S. (2020). Classification of Electrical Power System Conditions with Convolutional Neural Networks. *Engineering, Technology & Applied Science Research, 10*(3), 5759-5768.

König, H. J., Kiffner, C., Kramer-Schadt, S., Fürst, C., Keuling, O., & Ford, A. T. (2020). Human–wildlife coexistence in a changing world. *Conservation Biology, 34*(4), 786-794.

Kumar, M. S., Janaki, G., Hariharan, S., & Priya, P. S. (2021). IoT based warning system from the threat of wild animals. *2021 6th International Conference on Communication and Electronics Systems*, 550-553.

Kyara, V. C., Rahman, M. M., & Khanam, R. (2021). Tourism expansion and economic growth in Tanzania: A causality analysis. *Heliyon, 7*(5), e06966.

Lewis, A. L., Baird, T. D., & Sorice, M. G. (2016). Mobile phone use and human–wildlife conflict in northern Tanzania. *Environmental Management*, *58*(1), 117-129.

Listiyana, M. S. (2018). Tracking Object based on GPS and IMU Sensor. *2018 5th International Conference on Information Technology, Computer, and Electrical Engineering*, 214-218.

Manjari, K., Verma, M., & Singal, G. (2019). CREATION: Computational constRained travEl aid for objecT detection in outdoor eNvironment. *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems*, 247-254.

Massawe, E. (2021). *Development of an intelligent tracking system for monitoring rhinos and elephants: A case study of Ngorongoro conservation area* (Doctoral dissertation, NM-AIST).

Massey, A. L., King, A. A., & Foufopoulos, J. (2014). Fencing protected areas: A long-term assessment of the effects of reserve establishment and fencing on African mammalian diversity. *Biological Conservation, 176*, 162-171.

Mkonda, M. Y., & He, X. (2018). Agricultural history nexus food security and policy framework in Tanzania. *Agriculture & Food Security, 7*(1), 1-11.

Mwakisisile, A., & Mushi, A. (2019). Mathematical model for Tanzania population growth. *Tanzania Journal of Science, 45*(3), 346-354.

O'Connell-Rodwell, C. E., Rodwell, T., Rice, M., & Hart, L. A. (2000). Living with the modern conservation paradigm: Can agricultural communities co-exist with elephants? A five-year case study in East Caprivi, Namibia. *Biological Conservation, 93*(3), 381-391.

Pandey, N., & Pal, A. (2020). Impact of digital surge during Covid-19 pandemic: A viewpoint on research and practice. *International Journal of Information Management, 55,* 102171.

Premarathna, K. S. P., Rathnayaka, R. M. K. T., & Charles, J. (2020). An elephant detection system to prevent human-elephant conflict and tracking of elephant using deep learning. *2020 5th International Conference on Information Technology Research,* 1-6.

Rajagopalan, S., & Mathew, S. K. (2016). Choice of agile methodologies in software development: A vendor perspective. *Journal of International Technology and Information Management*, *25*(1), 3.

Rao, B. N., & Sudheer, R. (2020). Surveillance camera using IOT and Raspberry Pi. *2020 Second International Conference on Inventive Research in Computing Applications,* 1172-1176.

Sahoo, K. C., & Pati, U. C. (2017). IoT based intrusion detection system using PIR sensor. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology,* 1641-1645.

Saranu, P. N., Abirami, G., Sivakumar, S., Ramesh, K. M., Arul, U., & Seetha, J. (2018). Theft detection system using PIR sensor. *2018 4th International Conference on Electrical Energy Systems,* 656-660.

Sarris, A. H., Savastano, S., & Christiaensen, L. J. (2006). The role of agriculture in reducing poverty in Tanzania: A household perspective from rural Kilimanjaro and Ruvuma. *Research in Agricultural & Applied Economics,* 1-37.

Schwaber, K. (2004). *Agile project management with Scrum*: Microsoft Press.

Shaffer, L. J., Khadka, K. K., VanDenHoek, J., & Naithani, K. J. (2019). Human-elephant conflict: A review of current management strategies and future directions. *Frontiers in Ecology and Evolution, 6*, 235.

Sharma, P., Chettri, N., & Wangchuk, K. (2021). Human–wildlife conflict in the roof of the world: Understanding multidimensional perspectives through a systematic review. *Ecology and Evolution, 11*(17), 11569-11586.

Şimşek, E., Özyer, B., Bayındır, L., & Özyer, G. T. (2018). Human-animal recognition in camera trap images. *2018 26th Signal Processing and Communications Applications Conference,* 1-4.

Tepelea, L., Buciu, I., Grava, C., Gavrilut, I., & Gacsádi, A. (2019). A vision module for visually impaired people by using Raspberry PI platform. *2019 15th International Conference on Engineering of Modern Electric Systems,* 209-212.

Vangorp, P., Chaurasia, G., Laffont, P. Y., Fleming, R. W., & Drettakis, G. (2011). Perception of visual artifacts in image-based rendering of façades. *Computer Graphics Forum, 30*(4), 1241-1250.

Wang, X. (2016). Deep learning in object recognition, detection, and segmentation. *Foundations and Trends® in Signal Processing, 8*(4), 217-382.

Weldemichel, T. G. (2020). Othering pastoralists, state violence, and the remaking of boundaries in Tanzania's militarised wildlife conservation sector. *Antipode, 52*(5), 1496-1518.

Wijayagunawardane, M. P., Short, R. V., Samarakone, T. S., Nishany, K. M., Harrington, H., Perera, B., Perera, B. V. P., Rassool, R., & Bittner, E. P. (2016). The use of audio playback to deter crop-raiding Asian elephants. *Wildlife Society Bulletin, 40*(2), 375-379.

Williams, P. J., Schroeder, C., & Jackson, P. (2020). Estimating reproduction and survival of unmarked juveniles using aerial images and marked adults. *Journal of Agricultural, Biological and Environmental Statistics, 25*(2), 133-147.

Znidersic, E. (2017). Camera traps are an effective tool for monitoring Lewin's Rail (Lewinia pectoralis brachipus). *Waterbirds*, *40*(4), 417-422.

APPENDICES

Appendix 1: Interview Guide Questionnaire

The Nelson Mandela African Institution of Science and Technology

Research Title: Development of early warning system for human wildlife conflict using IoT and SMS

Dear Respondent

As part of my Master's research at Nelson Mandela African Institution of Science, I am surveying to investigate ways of reporting human-wildlife conflict incidence on a real-time basis using the Internet of Things in Tarangire National park. The results of this study shall be used to develop an early warning system for human-wildlife conflict. I would appreciate it if you could complete the following questionnaire. Any information obtained in connection with this study that can be identified with you will remain confidential. In any written reports or publications, no one will be identified. Participating in this study is voluntary.

If you have any questions about the research, please call Mr. Emmanuel Ronoh (+255717 654 246) or E-mail ronohe@nm-aist.ac.tz. at the School of Computational, Communication Science and Engineering, The Nelson Mandela African Institution of Science and Technology, Arusha-Tanzania.

Thank you very much for your feedback.

Interview Questions/Guide

1. Do you experience human-wildlife conflicts?
2. Which type of conflicts and how do you manage or solve them?
3. What is the main purpose of the camera traps? (to spot poachers, monitor wildlife, study wildlife or prevent human-wildlife conflicts)
4. Why did you choose camera traps instead of other technologies?
5. What are the challenges you are facing in using camera traps?
6. What improvement would you like to see with the use of camera traps?
7. Are you using camera traps with artificial intelligence?

Appendix 2: Website Codes

```
<?php
session_start();
if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
  header("location: login.php");
  exit;
}
require_once "config.php";
?>
<!DOCTYPE html>
<html>
 <head>
  <title>Wildlife monitoring system</title>
 <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">


 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  <script src="https://polyfill.io/v3/polyfill.min.js?features=default"></script>
  <script
src="https://unpkg.com/@google/markerclustererplus@4.0.1/dist/markerclustererplus.min.js"></script
>
  <style type="text/css">
   #mapx {
    height: 100%;
   }
   html,
   body {
    height: 400px;
    margin: 0;
    padding: 0;
```

```
 }
</style>
<script>
 function initMap() {
  const map = new google.maps.Map(document.getElementById("mapx"), {
   zoom: 9,
   center: { lat: -4.024, lng: 35.9887 },
  });
  // Create an array of alphabetical characters used to label the markers.
  const labels = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  // Add some markers to the map.
  // Note: The code uses the JavaScript Array.prototype.map() method to
  // create an array of markers based on a given "locations" array.
  // The map() method here has nothing to do with the Google Maps API.
  const markers = locations.map((location, i) => {
   return new google.maps.Marker({
    position: location,
    label: labels[i % labels.length],
   });
  });

  // Add a marker clusterer to manage the markers.
  new MarkerClusterer(map, markers, {
   imagePath:
    "https://developers.google.com/maps/documentation/javascript/examples/markerclusterer/m",
  });
 }

 const locations = [
```

```
<?php
if($_SERVER["REQUEST_METHOD"] == "GET"){
        if(isset($_GET["wildlife"])){
        $sql = "SELECT * FROM `incidences` WHERE id='".$_GET["wildlife"]."'";
        }else if(isset($_GET["category"])){
        $sql = "SELECT * FROM `incidences` WHERE wildlife='".$_GET["category"]."'";
        }
        else{
        $sql = "SELECT * FROM `incidences`";
        }
}
else{
        $sql = "SELECT * FROM `incidences` ORDER BY time DESC LIMIT 10";
}

                        if($result = $mysqli->query($sql)){
                                if($result->num_rows > 0){
                                        while($row = $result->fetch_array()){
                                                echo "{lat: ".$row['latitude'].", lng:
".$row['longitude']."},";
                                        }}}

          ?>
    ];
   </script>
  </head>
  <body>

   <div class="jumbotron" id="myBanner">
```

```
   <ul class="nav navbar-nav">

    <li ><a href="welcome.php">Home</a></li>

    <li class="active"><a href="map.php">map</a></li>

    <li><a href="#">Page 2</a></li>

   </ul>

      <ul class="nav navbar-nav navbar-right">

     <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#"><span
class="caret"></span> <?php echo htmlspecialchars($_SESSION["username"]); ?></a>

                                  </a>

                       <ul class="dropdown-menu">

       <li><a class="btn btn-primary ml-3" href="profile.php"><span class="glyphicon glyphicon-
user"></span>account</a></li>

       <li><a href="reset-password.php" class="btn btn-warning ml-3">reset password</a></li>

       <li><a href="logout.php" class="btn btn-danger ml-3">Logout</a></li>

      </ul>

            </li>

   </ul>

<?php

//include "menu.php";

?>

</nav>

  </div>

  <div class="col-sm-4">

                       <div class="list-group">

                                        <p class="list-group-item" id="header">The list of the wildlife sported in
our park</p>

                                        <?php

                                  $sql = "SELECT DISTINCT wildlife ,count(*) as num from incidences group by
wildlife;";
```

```
if($result = $mysqli->query($sql)){
        if($result->num_rows > 0){
                while($row = $result->fetch_array()){
                $wd=$row['wildlife'];
                $wl=$row['num'];
                ?>
                 <a href="map.php?category=<?php echo $wd; ?>" class="list-group-item "><?php echo $wd; ?><span class="badge badge-primary badge-pill"><?php echo $wl; ?></span></a>
<?php }}}
?>
    </div>
</div>
<div class="clearfix visible-xs"></div>
 <div id="mapx" class="col-sm-8"></div>
       <div style="background-color: #87CEFA;color:#CD853F;">
<p class="text-center"><a href="#">Privacy Statement</a>  |  <a href="#">Security Statement</a><a href="#"></a>  |  <a href="">Contact</a><a href="#"></a>  |  <a href="#">Remote Assistance</a></p>
<p class="text-center">&copy; 2021 human wildlife management system| Design by Emmanuel ltd. All Rights Reserved.</p>
</div>
 <script
  src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBrf9q0nN2qxp1ceAVM0nOU6V-K8-HuUbs&callback=initMap&v=weekly"
  async
 ></script>
</body>
</html>
```

Appendix 3: Raspberry pi Codes

```
#import board
#import digitalio
import serial
import time
import string
import pynmea2
import time
import numpy as np
import picamera
from PIL import Image
from tflite_runtime.interpreter import Interpreter
import json  # optional - for debugging json payloads
# init Raspberry Pi Camera
camera = picamera.PiCamera()
# specify paths to local file assets
path_to_labels = "labels.txt"
path_to_model = "model2.tflite"
path_to_image = "image.jpg"
# confidence threshold at which you want to be notified of a new bird
prob_threshold = 0.4
def main():
    """ check to see if PIR sensor has been triggered """
    #if pir_sensor.value:
    check_for_wild()
    time.sleep(0)  # only check for motion every 30 seconds!
def check_for_wild():
    """ is there a bird at the feeder? """
    labels = load_labels()
    interpreter = Interpreter(path_to_model)
    interpreter.allocate_tensors()
    _, height, width, _ = interpreter.get_input_details()[0]['shape']
    camera.resolution = (height, width)  # ML model expects 224x224 image
    camera.start_preview()
    time.sleep(20)  # give the camera 2 seconds to adjust light balance
    camera.capture(path_to_image)
    image = Image.open(path_to_image)
    results = classify_image(interpreter, image)
    label_id, prob = results[0]
    print("bird: " + labels[label_id])
    print("prob: " + str(prob))
    camera.stop_preview()
    if prob > prob_threshold:
        bird = labels[label_id]
        bird = bird[bird.find(",") + 1:]
        prob_pct = str(round(prob * 100, 1)) + "%"
        send_note(bird, prob_pct)
def load_labels():
    """ load labels for the ML model from the file specified """
```

```
    with open(path_to_labels, 'r') as f:
        return {i: line.strip() for i, line in enumerate(f.readlines())}
def set_input_tensor(interpreter, image):
    tensor_index = interpreter.get_input_details()[0]['index']
    input_tensor = interpreter.tensor(tensor_index)()[0]
    input_tensor[:, :] = image
def classify_image(interpreter, image, top_k=1):
    """ return a sorted array of classification results """
    set_input_tensor(interpreter, image)
    interpreter.invoke()
    output_details = interpreter.get_output_details()[0]
    output = np.squeeze(interpreter.get_tensor(output_details['index']))
    # if model is quantized (uint8 data), then dequantize the results
    if output_details['dtype'] == np.uint8:
        scale, zero_point = output_details['quantization']
        output = scale * (output - zero_point)
    ordered = np.argpartition(-output, top_k)
    return [(i, output[i]) for i in ordered[:top_k]]
def send_note(animal, prob):
    gps="home"
    #a="Latitude=" +str(len(gps))
    #print(a)
    while len(gps)<8:
        port="/dev/ttyS0"
        ser=serial.Serial(port, baudrate=9600, timeout=0.5)
        dataout =pynmea2.NMEAStreamReader()
        newdata=ser.readline()
        newdata = newdata.decode("utf-8","ignore")
        if newdata[0:6]== "$GPRMC":
            newmsg=pynmea2.parse(newdata)
            lat=newmsg.latitude
            lng=newmsg.longitude
            gps="Latitude=" +str(lat) + "and Longitude=" +str(lng)+ " and Wild animal=" +str(animal)+ " and
prob=" +str(prob)
            print(gps)

#    """ upload the json note to notehub.io """
#    req = {"req": "note.add"}
#    req["file"] = "bird.qo"
#    req["start"] = True
#    req["body"] = {"bird": bird, "prob": prob,
#                "from": sms_from, "to": sms_to}
#    rsp = card.Transaction(req)
#    # print(rsp) # debug/print request
while True:
    main()
```

Appendix 4: Poster Presentation

Early Warning System for Human Wildlife Conflicts Using deep learning, Internet of Things and SMS

Background.

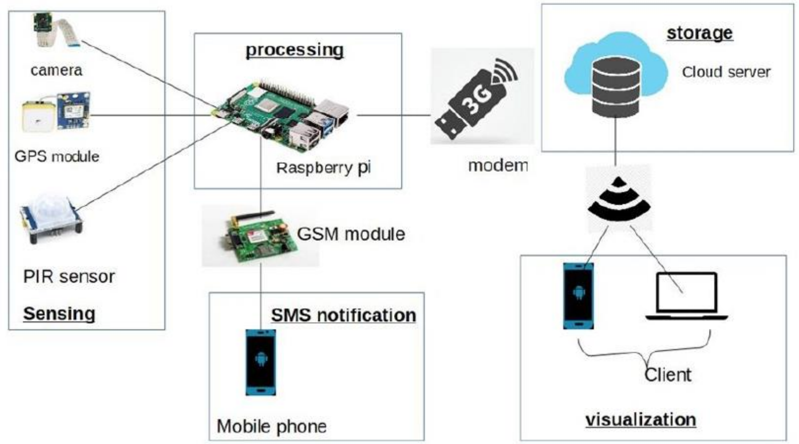
Northern Tanzania hosts major parks in the country. The country majorly depends on agriculture as a significant source of income for her citizens and wildlife tourism as major source of foreign exchange. Wildlife from the park may sometimes escape the park and raiding people's crops, even injuries or killing livestock and a time people.

Currently, they have mitigation solutions that includes wildlife collaring which is quite expensive and hotline numbers which is reactive in nature.

Objectives.

The main objective of this project is to develop low cost human wildlife conflict early warning system for Tarangine national park.

System architecture



Solution.

The proposed system utilizes raspberry pi as main system processing unit, sensing unit have PIR sensor, camera module and GPS module, while notifications is done through SMS and online map cloud notification.

Industrial application.

- Camera trap for monitoring wildlife in the park.
- Camera trap for monitoring farm intrusion.