A Major Project Final Report on

ALERT SYSTEM USING TENSORFLOW 'WATCHMAN'

Submitted in Partial Fulfillment of the Requirements for the Degree of **Bachelors of Engineering in Computer Engineering** under **Pokhara University**

Submitted by:

Amrit Sunar,14302

Binita Rai,14711

Prakash Yadav,14331

Rabin Rai,14333

Suraj Suwal,14345

Under the supervision of

Mr. Chandan Gupta

Date:

06/12/2018

Department of Computer

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

ACKNOWLEDGEMENT

First and foremost, we would like to thank our supervisor **Er. Chandan Gupta** for his guidance throughout the project. We would also like to thank **Er. Shiva Hari Acharya, Er. Saroj Bista** and **Dr. Roshan Chitrakar** for their proper guidance and support for counseling throughout our project. Without their invaluable advice and assistance, it would have not been possible for us to complete this part of our project on time.

Moreover, we would like to owe our special thanks to all subject teachers who provided us with necessary information during the course of studies which somehow helped us in various ways for the development of project. Similarly, we truly admire the effort of all persons who knowingly or unknowingly provided us with the concepts about the project that assisted us for the development of the system.

ABSTRACT

Here, in this project we tried to implement the object detection and recognition to detect and recognize the wild animals through the application of the TensorFlow library. Here we made a device that can detect and recognize what kind of animal is in its frame and can generate the sound to derive out them and also the device can generate alert sound to alert the peoples about these wild animals.

The propose for developing "**Alert system using TensorFlow 'Watchman'"** is to monitor the possible threats of the wild animals on human beings to those villages that are close to the jungles through the video surveillance. It facilitates the identification of possible threats/danger from different animals and notify the humans about that threat. It also provides the facility to track down what kind of possible loss may arise by animals on humans and their farming's.

We proposed to implementation of the CNN based object detection algorithm on Respberry PI. The model architecture is based on TensorFlow library to get image feature maps and to detect the wild animals.

Keywords*: Video surveillance, Object detection, Recognize, Respberry PI.*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Wildlife is one of the best attraction of the Nepal. Many peoples come here to see these animals and are still growing. Wildlife diversity is a notable feature of Nepal. Nepal has a large variety of plants and animals because of the variance in climate, from tropical to arctic. Wildlife tourism is one of the major source of tourism in Nepal. In Nepal a large number of rhododendron species can be found. They are no natural habitats in Nepal. Nepal has established numerous National Parks and reserves in order to protect its diverse fauna [1]. But there is also some negative part of these wildlife animals. People are being attacked by large mammal species such as tigers, common leopards, rhinoceros, elephants, bears, etc.

Rural area near the forest are venerable to wild animals. The danger might be for the human beings or for their occupation. The wild animals like tiger, elephant, rhino, etc. are threat to human society. Other animals like monkey, Boar, Deer, etc. are harmful and destroy Crops. So, we need to monitor and protect ourselves and the farms from these animals. For this, we have to watch and monitor the possible danger and threats by ourselves.

Here, we proposed the system that can continuously watch and monitor the possible dangers and give some notifications to the peoples on the village with minimum involvement of humans. The main aim of our project is to reduce the possible loss on human beings and their crops by monitoring the wildlife near the village and giving the result information to the peoples. Here our system exists in between the possible incoming path of the wild animals towards the village. The system detects the motion first and then detects the type of animal through camera surveillances. After that the system generate some sound to drive out these animals and it also alert the peoples on the village.

Deep learning-based object detection has been very successful in recent years. Especially the CNN (convolutional neural network) model has significantly improved the recognition accuracy on large data-sets. For the ImageNet benchmark data set, the CNN based model has been dominating the leader-board since it’s introduced by Krizhevsky in 2012 for the first time. [2]

The main motivation for us to go for this project was the continuous and inefficient traditional manual monitoring system for threats from wild animals. This made us to think why not make it automated, fast and mush efficient.

Injury and death from wildlife attacks may results in people feeling violent resentment and hostility against the wildlife involved and, therefore, support for wildlife conservation may not be topic of interest for public. Though Nepal, with rich biodiversity, is doing good in its conservation efforts, human-wildlife conflicts have been a major challenge in recent years. We analyze patterns of human injury and death caused by large mammals using data from attack events and their spatiotemporal dimensions collected from a national survey of data available in Nepal over five years (2010–2014). [3]

Table 1. 1: Patterns of human death and injury due to large-mammal attacks

Wildlife	**Contribution (%)**	**Avg. attacks/yr[a]**	**Avg. fatalities/ yr[a]**	**Avg. attacks/season[b]**	**Avg. fatalities/season[b]**
Elephant	30	27.4±7.7	18±4.6	34.2±16.5	22.5±11.7
Leopard	21	19.4±11.6	8±5.4	24.2±3.8	10±6.6
Rhinoceros	18	17±4.3	3±1.2	21.2±16	3.7±3.5
Bear	12	11±4.3	1±1.2	13.7±2.6	1.2±1.2
Tiger	10	8.8±5.4	4.8±3.3	11±4.8	6±1.4
[a] Observation Period= 5years [b] Number of seasons per year= 4 seasons					

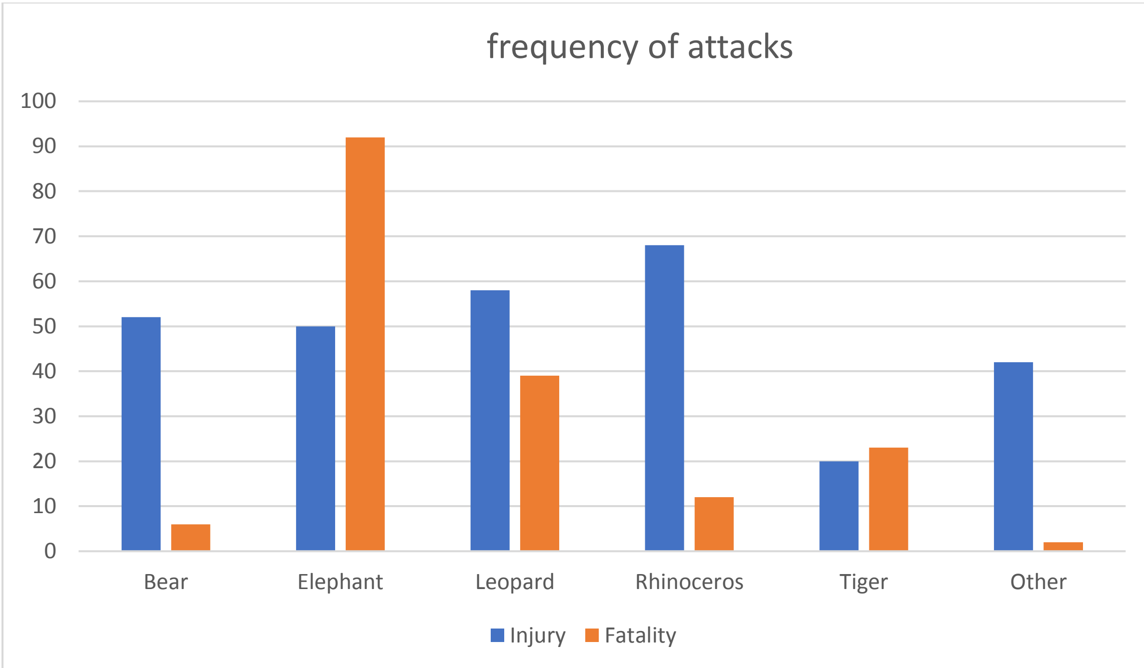


Figure 1. 1: Frequency of attacks

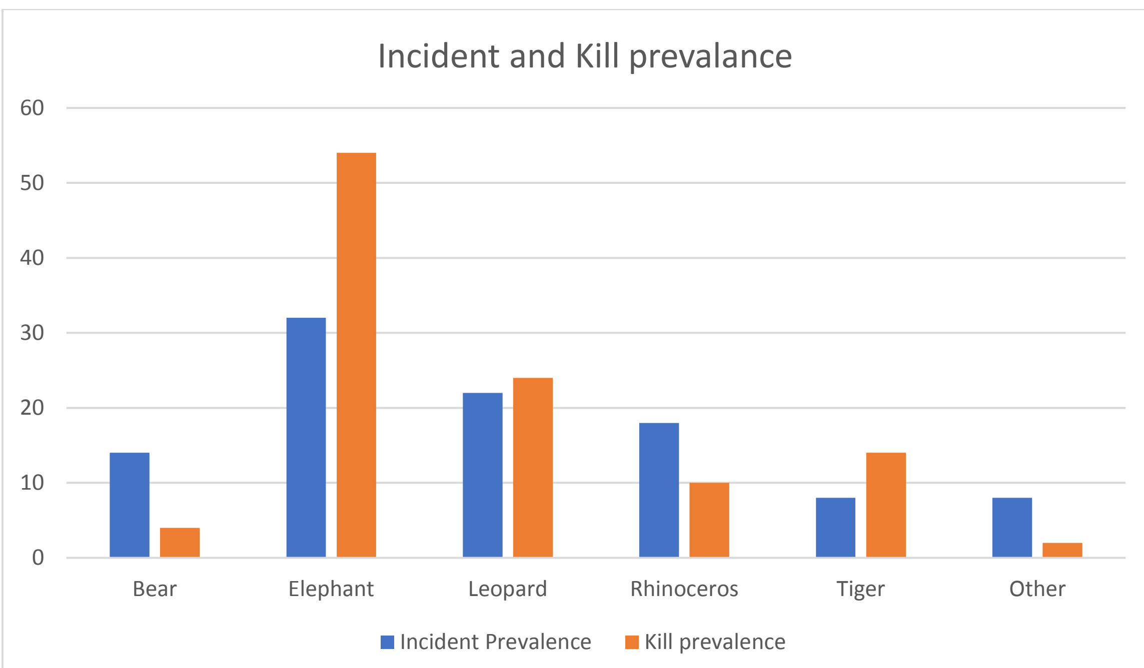


Figure 1. 2: Incident prevalence and kill prevalence

2. PROBLEM STATEMENT

The traditional manual methods of monitoring and protecting the human beings and their farming from possible danger and threats from wild animals is hard and tedious as the continuous monitoring is needed which have to fulfil by peoples themselves. This is hard, tedious, time consuming and prone to inaccuracies as people may not have time to monitor and track the possible threat continuously.

In Nepal, people are mainly attacked by large mammals such as tigers, common leopards, rhinoceros, elephants, bears and others. Other than these animals like Deer, Boar, Monkey, etc. harms the corps. So, we need continuous monitoring from these possible danger and threats.

Use of the object detection and recognition system in lieu of the traditional methods will provide a fast and effective method of monitoring the possible threats continuously while offering an easy, stable and robust service for tracking the possible danger and threat.

While developing the product, we have been faced various problems. The difficulty arose during gathering the large image data and train those images as the main objective of the project is to detect the animals from those images. Another problem that we have been faced is to integrate the project. Here we also have difficulties to gather requirement for sound, from which animal can drove out. Also, the system fails to differentiate the real object from the dummy object.

CNN based model can achieve higher accuracy, but they have following disadvantages:

High computation cost. In the CNN based model, each layer takes a lot of computation since are usually very deep with tens or hundreds of layers.

large memory demand. In the CNN based model, it has a lot of parameters that usually take hundreds of Megabytes of memory space.

Low efficiency. Most CNN based model are designed without efficiency improvement.

3. JUSTIFICATION

This project serves to automate the prevalent traditional tedious and time-wasting methods of monitoring and tracking wild animals. The use of “Alert system using TensorFlow ‘Watchman’” will increase the effectiveness of monitoring & tracking the wild animals and alerting the peoples about these animals.

This method could also be extended for use in similar fields such as animal tracking and counting’s for the purpose of wild animal conservations. Applications of object detection and recognition are widely spreading in different areas such as vehicle tracking and identification, face detection and recognitions, OCR, etc.

4. SCOPE OF THE PROJECT AND LIMITATIONS

The intention for developing this project is to computerize the traditional way of tracking and monitoring the possible threats from wild animals. Another purpose of this project is to generate the desired sounds to drive out the wild animals and alert peoples about those wilds.

The project is developed as a desktop application, and it will work for a particular area at a time.

This project serves to automate the prevalent traditional hard, non-efficient and continuous methods of monitoring and alerting peoples from the danger from wild animals to humans and their farms. The use of automatic watching and tracking the threats through object detection and recognition will increase the effectiveness of threat monitoring, management and alert system.

This method could also be extended for use in detecting the wild animals that may harm the corps and farming. Applications of object detection are widely spreading in areas such as autonomous driving, face detection, video surveillance, security systems, image and film processing, etc.

Although this product has various advantaged but also have some limitations. Here, after the product has been fully developed, we faced various problems. The system does not exactly detect those wild animals which may harm. The leading cause for this problem are heavy humidity, heavy rainfall, wildfire, distance, light sensitivity of the camera device etc. which are being problematic for device to capture the image. The other problem we faced is the system does not able to drive out / rout those harmful wild animals as expected by the system since it is not always possible to predict the behavior of wild animals.

5. PROJECT OBJECTIVE

The overall objective is to develop an automated Alert **system using TensorFlow “watchman**” comprising of a desktop application working to perform the following tasks:

- To identify motion of wild animals near the village.
- To detect & recognize the wild animals through video surveillance using CNN based algorithm.
- To generate the sound specific for the detected animals that drive out these animals and also to alert the people.
- To analyze the architecture and algorithm that can help in other similar fields.

6. SIGNIFICANCE OF THE STUDY

- First and foremost, the system reduces the threat level from the wild animals faced by people specially those living in the area near to the forest.
- This system also eliminates the traditional way of driving away the wild animals which may harms the crops as well other property along with human damages.
- Wild animals are also very known for their aggressive behavior towards people. So, this system provides security to the people by alerting them of the danger helping them to maintain sound mental health.
- Due to the various damages done by wild animal, there have been many conflictions about destroying them. So, this system helps to reduce these conflictions as it helps to control the destructions of wild animals.
- As there have been many illegal hunting as well as poaching of animals, this system helps to control these illegal activities.
- There has been great increment in the efficiency in the task of driving out animals as system works much more efficiently than that is done manually which saves the time of people.
- In future we can implement this type of animal detection systems in between the roads within the jungles, national parks etc. to avoid possible threats in human beings by these animals and vice versa.

7. FEASYBILITY STUDY

Economic Feasibility:

The system developed is economic with respect to National park, conservation authorities & office and the government point of view. It is cost effective in the sense that has eliminated the manual monitoring and tracking completely. Since it needs to be installed only one time and the system will work until the system crash or power failure.

Time period:

The system is time effective because the continuous manual monitoring is eliminated. The system minimizes the manual monitoring of the wild animals.

Technical feasibility:

The technical requirement for the system is economic & easily available since all the development software are open source and the system uses Hardware that can be easily available.

Behavioral Feasibility:

The system working is quite easy to use and does not take any longer time since once the system installed, it will work till any serious failure. Also, the operator requires no special training for operating the system.

8. LITERATURE STUDY

Many systems related to wild animal has been developed so far. These system works for the conservation as well as promotion of wild animals. Wild animal protection has been one of the main concern in today's world since many illegal hunting and poaching of an animal are being carried out. Although many systems to protect the wild animal has been developed we haven't found the system for protecting people who faces economic as well as physical damages.

8.1 EXISTING SYSTEM

Here existing system [6] consists various animal detection and tracking systems that are implemented to detect the animals on roads to prevent from possible threats on humans from these animals and vice-versa.

Kootenay, British Columbia, Canada

In June 2002 an animal detection system was installed along Hwy 93, in Kootenay National Park in British Columbia, Canada, about 60km north of Radium, immediately north of the Dolly Varden Day-Use. The system was designed to detect large animals, specifically white-tailed deer (Odocoileus virginianus). Two infrared cameras that detect heat and additional equipment were installed in the right of way. The software uses a combination of motion, speed and size to determine whether the warning system should indeed be triggered. The system, especially the cooling system of the cameras, experienced technical difficulties during the first year (June through October 2002).

A modified system with different infrared cameras was installed in May 2003 (Pers. com. Hillary Page, Sage Consulting). The road length covered by the system was cut in half (from 2,000m to 1,000m) because of the different cameras. The system has standard black-on-yellow deer warning signs with amber flashing lights on top to warn drivers. The system became operational in September 2003 (Pers. com. Nancy Newhouse, Sylvan Consulting). The system is currently only active from dusk to dawn. The system may eventually be operational 24 hours a day.

Seven Locations, Switzerland

Kistler (1998) and Tschudin (1998) reported on a study that covered seven locations in Switzerland. The systems were supplied by Calonder Energy AG in Dietikon, Switzerland. Each system consisted of a series of passive infrared sensors. The sites, their installation date, the width of the crossing area and number of sensors installed, are listed in table below. The passive sensors were designed to detect ungulates such as roe deer (Capreolus capreolus) and red deer (Cervus elaphus) within a 30-100m radius.

Rosvik, Sweden

In 1999 an animal detection system was installed along highway E4 near Rosvik in northern Sweden (between Piteå and Luleå). The system was designed by PIK AB, Karlskrona, Sweden, and it was installed by the manufacturer and the Road Administration. The system operates on a break-the-beam principle with infrared light. The system was installed in a 100-m-wide opening in a fence, and it was designed to detect moose. When an animal is detected lights are turned on that illuminate the highway at the crossing area. This should allow drivers to see the animal better. In addition, red warning lamps in the right-of-way are activated. A standard moose crossing sign with the text “wildlife passage” is located just before the crossing area. The electricity supply was a major problem, but that issue was solved in winter 2001/2002.

The other similar animal detection systems were implemented at different locations as shown in table below,

Table 8. 1: showing animal detection systems at different places.

ID #	Location	Target species	Distance covered	Fence	Cost system	Cost Install.	Evaluation Installed		Operational
a	7 loc., Switzerland	Roe & red deer	50-200m	No	$11,500	?	OVC	‘93-‘96	‘93/’96-present[1]
b	Box, Uusimaa, Finland	Moose	220m	Yes	$60,000	$40,000	OV	Sep ‘96	Dec ’96-present
c	Mikkeli, Finland	Moose	90m	Yes	$40,000	$30,000	O	‘99	‘99-present
d	Nugget Cany., WY, USA	Mainly mule deer	92m	Yes	$200,000[2]	?	OV	1 Dec ‘00	8 Dec ’00-21 May ‘01
e	KootenayNP, BC, Canada	White-tailed deer	1,000m	No	?	?	O	Jun ‘02	Sep ’03-present
f	4 loc. CH; 2 loc. D	Roe & red deer	?		±$20,000[3]		?	‘98-‘01/‘02?	?
h	Rosvik, S	Moose	100m	Yes	±30,000[4]	?	O	‘99	’00-present
i	Colville, WA, USA	Deer, elk	402m	No	$9,000[5]	$3,000	O	20 Jun ‘00	Taken down spring ‘02
j	Marshall, MN, USA	White-tailed deer	1,609m	No	$50,000	$7,000[7]	O	Jun ‘01	Turned off Nov ‘01
k	Wenatchee, WA, USA	Deer	213m	No	<$40,000[8]	?	O	Oct ‘02	Oct ‘02-present
l	Yellowstone NP, MT, USA	Elk	1,609m	No	$350,000[9]	$60,000	O	Oct/Nov ‘02	Not operational
m	South Bend, IN, USA	White-tailed deer	9,654m[10]	No	?	?	O	Apr ‘02	Not operational
n	Sequim, WA, USA	Elk	4,827m	No	$60,000[11], $13,000[12]	?	O	Apr ‘00	Apr ‘00-present
o	Harris, SK, Canada	Mostly Mule deer	5,000m	No	$36,000[13]	?	O	Apr ‘02	Apr ’02-present

[1] All in operation except Marcau site (road work Aug ‘97)
[2] Incl. operat. & maint., research, excl. WYDOT salaries
[3] In Switzerland
[4] Excl. ± $70,000 for electricity
[5] Excl. signage, batteries
[13] Excl. in kind contributions [7] Excl. salariesTable 2. Main characteristics of the seven systems located in Switzerland (Kistler 1998).
[8] Incl. research, design, installation
[9] Including research and development
[10] Divided over 6 sections (1 mile each)
[12] For herding and collaring

8.2 OBJECT DETECTION

Object detection involves in detecting instances of objects from a particular class in an image captured. The goal of object detection is to detect all instances/features of objects from a known class, such as people, cars or faces in an image. Usually, only a small number of instances of the object are present in image, but there are various possible locations and scales at which they can occur and that need to somehow be expressed.

Images of objects from a specific class are highly stochastic. One source of randomness is the actual imaging processing. Variations in illumination, variations in camera position as well as digitization artifacts, all produce significant variations in image appearance, even in a stable scene. Another source of variation is because of the intrinsic appearance variability of objects within a class, even assuming no variation in the imaging process. For example, people have variable shapes and wear a variety of clothes, while the handwritten digit 7 can be written with or without a line through the middle, with different slants, stroke widths, etc. The hard part is to develop detection algorithms that are invariant with respect to these variations and are computationally efficient.

Figure 8. 1: showing object(animal) detection

8.3 BASICS KNOWLEDGE OF DEEP LEARNING

The complete name is Machine Deep Learning; it's a branch of the learning based on a set of algorithms, that attempts to model high level abstractions in data, using multiple processing layer. These models are based on two fundamental concepts, the neural network and the back-propagation algorithm.

The first is a system that tries to approximate the human brain structure with programs and data structure combination. The network initially is trained on a large amount of data and rules about their relationship and then tested on the learned function, like object detection. The structure of the network is usually composed by three fixed layer typology: Input Layer, Hidden Layer, Output Layer. The ones that really learn are the hidden ones, that can be 1+n, in any combination and functionality, where their total number defines the length of the model; from here the adjective deep.

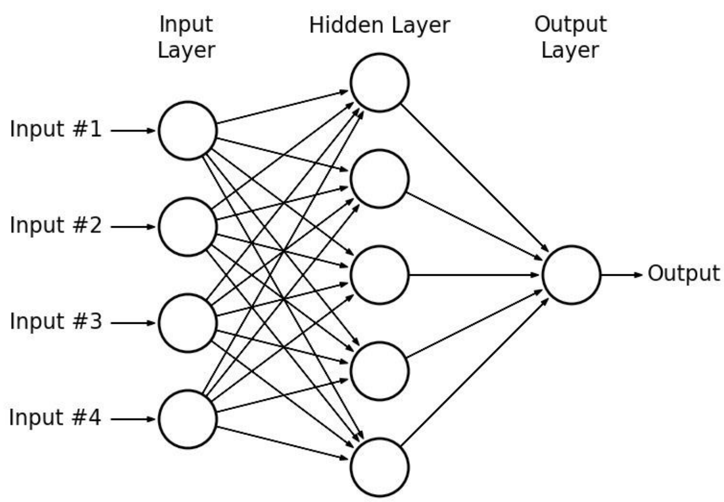


Figure 8. 2: Neural network showing its i/p, o/p and hidden layers

Then it's the Back-Propagation Algorithm that makes possible training the model respect to an optimization method. The system calculates the gradient of a loss function, respect to all the weights in the network and then it's passed to the optimization method that updates all the network weights trying to minimize the loss function. [5]

8.4 TENSORFLOW

Released on the 15th of November 2015 by Google, TensorFlow is the newest open source library written in Python and c++ for numerical computation.

It has immediately a great success in the Machine Learning community and in less than one year it also had a lot of support and development by Google itself, more over by many other projects, developed in any area of Deep Learning.

The peculiarity of TensorFlow is its work flux, made by data flow graphs. Where Nodes represent mathematical operations, edges represent the multidimensional data arrays communicated between them; the latter can be considered, as in electronics, a Tensor, from here its name. In May 2016, Google has revealed that it has used TensorFlow in AlphaGo project, with a special hardware dedicated to boost library’s performances.

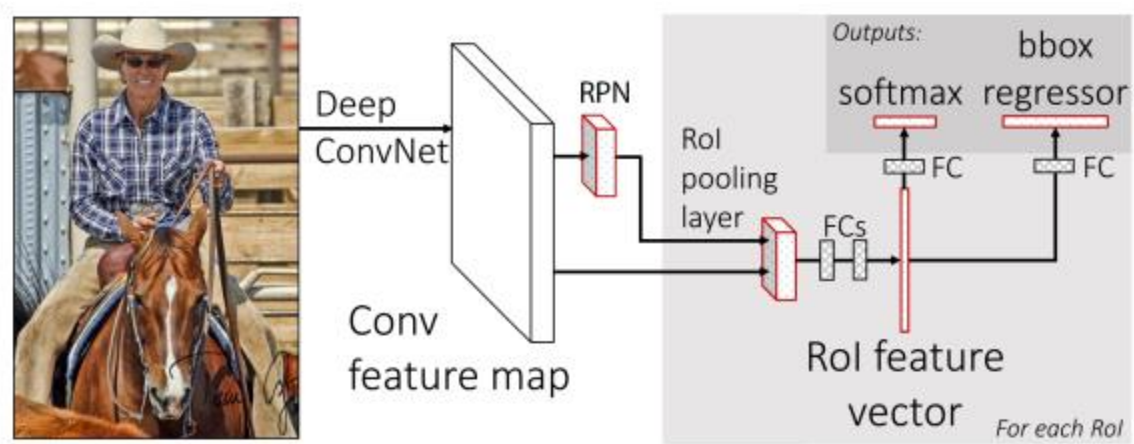
To reach rapidly this goal, Google dedicated a special attention to the user experience of TensorFlow, which arrives with a great basic support and a wellgrown GitHub community, the key of this quick improvement.

All these are the peculiarities that pushed us to choose TensorFlow, over more developed and bigger communities. The irruption made in the Deep Learning environment, shows up its great future potentialities, that are rolling out day by day. [5]

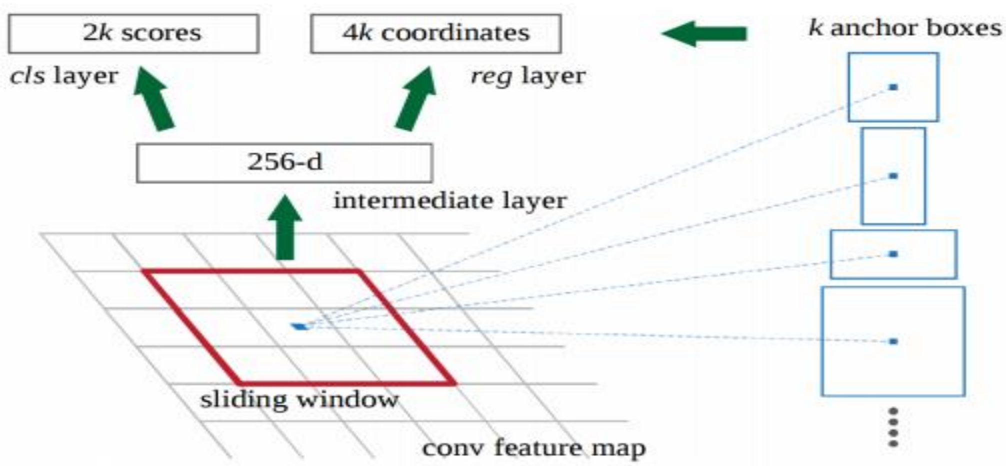
Figure 8. 3: TensorFlow logo

8.5 FASTER R-CNN

Faster R-CNN is built upon Fast R-CNN. It unifies proposal and classification parts into a single neural network which provides nearly cost-free region proposals and allows end-to-end training and testing. Its architecture consists of multiple convolutional layers, Region Proposal Network (RPN), a few fully-connected layers and two output layers – one for classification and one for bounding box regression. Either ZF model, which has 5 shareable convolutional layers, or VGG16 model, which has 13 shareable convolutional layers, is used. The convolutional layers are interspersed with ReLU and max-pooling layers.



RPN generates object proposals by sliding a small window over the feature maps. Each proposal is given by a bounding-box and objectness score. The bounding boxes are regressed from 9 anchor boxes that improve ability to detect objects of various scales and aspect ratios.



The feature maps and 300 proposals with the highest objectness score are passed to a region of interest (RoI) pooling layer. The RoI pooling layer uses max-pooling to crop and reshape the regions of feature maps defined by the proposals to achieve the same spacial size, typically 6 by 6. The resized regions are then processed by two fully-connected layers and the resulting feature vector continues to two sibling fully-connected layers. One is followed by softmax and generates probabilities of each class (including background). The other fully-connected layer outputs regression of proposed bounding-boxes. There are two proposed strategies to train the Faster R-CNN network, both of them use stochastic gradient descent. One alternates between training RPN alone and the rest of network, the other is end-to-end but the gradients for RPN are only approximate.

8.6 CREATING A SET OF TRAINING IMAGES WITH TENSORFLOW

The first place to start is by looking at the images you've collected, since the most common issues we see with training come from the data that's being fed in.

For training to work as we expect, you should collect at least a hundred photos in every possible surrounding condition of an object you want to recognize. The more you collect, the better the accuracy of your trained model is likely to be. You also need to make sure that the photos are of good representation of what your application will actually encounter. For example, you take all your photos indoors and your users are trying to recognize objects outdoors, you probably won't get good results.

Another risk to avoid is that the learning process will pick up on anything that the labeled images have in common with each other, and if you're not careful that might be something that's not useful. For example, if you capture kind of object in a blue room, and another in a green one, then the model will prediction on basis of the background color, not the characters of the object you actually care about. To avoid these problems, take pictures in a variety of situations as you can, at different times, and with different devices.

You may also want to think about the classes you use. It might be worth categorize big verities that cover a lot of different forms into smaller ones that are more visually distinct. For example, for ‘vehicle' we might use 'car', 'motorbike', and 'truck' etc. you also have to think about whether you have a 'closed world' or an 'open world' problem. In a closed world problem, things to categorize are the classes of object you know about. On contrasting, a roaming robot may see all types of different things through its camera as it wanders around the world. In those situations, you'd want the classifier to report if it was not sure about what it was seeing. This cannot be easy to do well, but if you collect a large number of specific 'background' photos with no relevant objects in them, you can do well. It is also good to make sure that all of your images are labeled correctly. Often user-generated tags are unreliable for our purposes. For example: pictures tagged `#daisy` might also include people and characters named Daisy. [7]

9. METHODOLOGY

9.1 SOFTWARE DEVELOPMENT PROCESS MODEL

The framework we used in development of this software is “Incremental model”, i.e. a method for software development where the product is designed, implemented, and tested incrementally. This model combines the elements of waterfall model with iterative philosophy of prototyping i.e. multiple development cycles are present here, that makes the life cycle a multi-waterfall cycle. In Incremental model, the whole task is divided into various increments.

In incremental model, the first increment is a core product. That means, basic requirements are incorporated, but many supplementary features (some known, others unknown) remain un-delivered. The core product is used by the end user or undergoes detailed review. As a result of use and/or evaluation of an increment, plans are developed for the next increments. The plan tries to incorporate the modification of the core product to meet the needs of the customer and the delivery of additional features and functionality. This process is iterative through the delivery of each increment, until the complete product is produced.

Here we used Incremental model since we gathered and analyzed all the requirements for the project i.e. requirement for the system are clearly defined and understood. Major feature of the project is also known. So, we planned and divided our whole task into four increments.

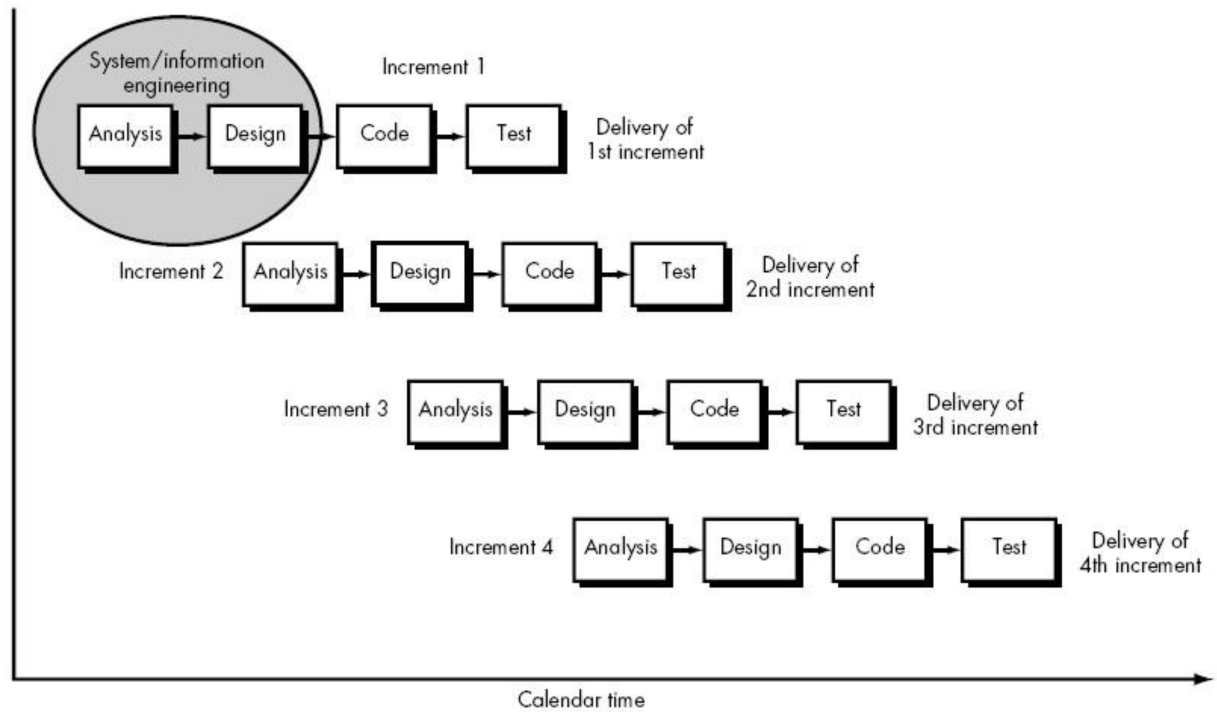


Figure 9. 1: Incremental Model

Analysis Phase: in this phase, gathering requirements and analyzing were done to find out the actual requirement of the system. The result of this phase is a SRS (System Requirement Specification). Here are some requirements for the project as,

Functional Requirements

System's functional requirement describes about the activities and services that must be provided i.e. the user/customer requirements.

- tracking wild animals in-front of device by object detection and recognition in specific time.
- Generate specific sound or any noise to drive out those animals.
- Generate sound or any alert type to alert peoples about detected wilds.

Here, the requirements are fulfilled as expected.

Non-Functional Requirements

The Non-functional Requirements are the characteristics or attributes of the system/project that can judge its operation. The following points clarify them:

- **Accuracy and Precision:** the system performed its process in accuracy and Precision as expected to avoid problems.
- **Security:** the system is secure since there are no external interferences that can be arose once the system installed.
- **Usability:** the system is easy to deal with and simple to understand since it does not need much interactions.
- **Speed and Responsiveness:** Execution of operations should be fast but does not respond as fast as we expected due to the processing power of Respberry Pi.

Design Phase: In this phase the SRS is translated into the systems is design. Context Diagram, DFD, use case, activity diagram and class diagram have been developed. The modeling diagrams are presented at system development section.

Coding Phase: In this phase code needed to map designs were done as per the design and a working system is developed by the end of this process.

Testing Phase: In this phase, the system is tested through different testing methods. With each testing a list of changes to the system were developed, suggested and the changes were applied to the software and the software is delivered as a successive increment until a satisfying system is achieved. The test results are shown below in each increment.

Advantages of Incremental model:

- gives core software quickly and early during the software life cycle.
- is less costly, more flexible and easy to change scope and requirements.
- easier to test and debug for a smaller iteration.
- customer can respond to each iteration.
- Lowers initial delivery cost.
- Easy to manage risks because risky pieces are identified and handled during iterations.

Disadvantages of Incremental model:

- Needs good planning and design.
- Needed a clear and complete specifications of the whole project before it can be broken down and built incrementally.
- Total cost is higher than waterfall.

When to use the Incremental model:

- This model can be used if the requirements of the system are clearly defined & understood.
- Major features must be defined; however, some details can evolve with time.
- If there is a need to get a product to the market in less time.
- If resources with needed skills sets are not available.

9.2 TOOLS AND TECHNOLOGIES USED

To realize project successfully, it is necessary to select the right tools and technologies, Designs should be made with the end-product in mind and keep in mind that which tools and applications should be used to build the applications and how to make the environment scalable enough for the final product. Tools and Technologies that are used in this project/system are as folloews.

Development Hardware required:

Table 9. 1: showing hardware required

Section	**requirements**
Display	Minimum resolution 800 x 600 pixel 1024 x 768 pixel recommended
RAM	4 GB or more recommended especially for Microsoft Windows 7, 8 and 10.
CPU	1.7 GHz processor speed or higher
GPU	GTX 960 or higher GTX 1050 Ti or higher recommended
Webcam	Minimum 360p webcam required (recommended 720p, 5MP or higher)

Development Software required:

Table 9. 2: showing software required

Section	**requirements**
Supported Operating System	Resbian OS
Additional Software Library Requirements	Lxml, anaconda, Cython, jupyter, matplotlib, pandas, cuda, cudnn
Development software requirements	Python 3+, OpenCV library, TensorFlow lib.

Component Used

Table 9. 3: showing components required

Section	**requirements**
Respberry pi	Pi model 3 B^+
Webcam	Pi camera module / webcam (Logitech c310)
Motion sensor	Ultrasonic Sensor (HC-SR04)
Arduino	Arduino Yun
Speaker	Jy-25

9.3 SYSTEM DEVELOPMENT

As we proposed the system that can continuously watch and monitor the possible dangers and give some notifications to the peoples on the village with minimum involvement of humans, it has been developed as expected. The main aim of our project is to reduce the possible loss on human beings and their crops by monitoring the wildlife near the village and giving the result information to the peoples. Here our system exists in between the possible incoming path of the wild animals towards the village. The system detects the motion first and then detects & recognize the type of animal through camera surveillances. After that the system generate some sound to drive out these animals and also alert the peoples on the village.

Block diagram

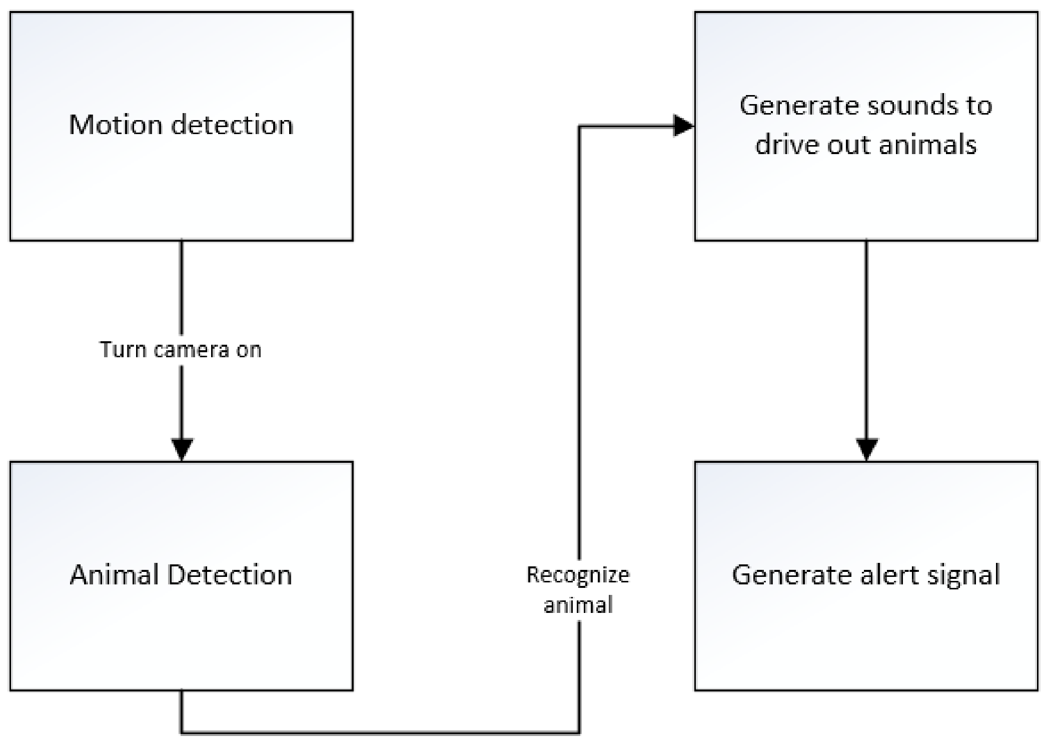


Figure 9. 2: Block diagram for the project

Flowchart

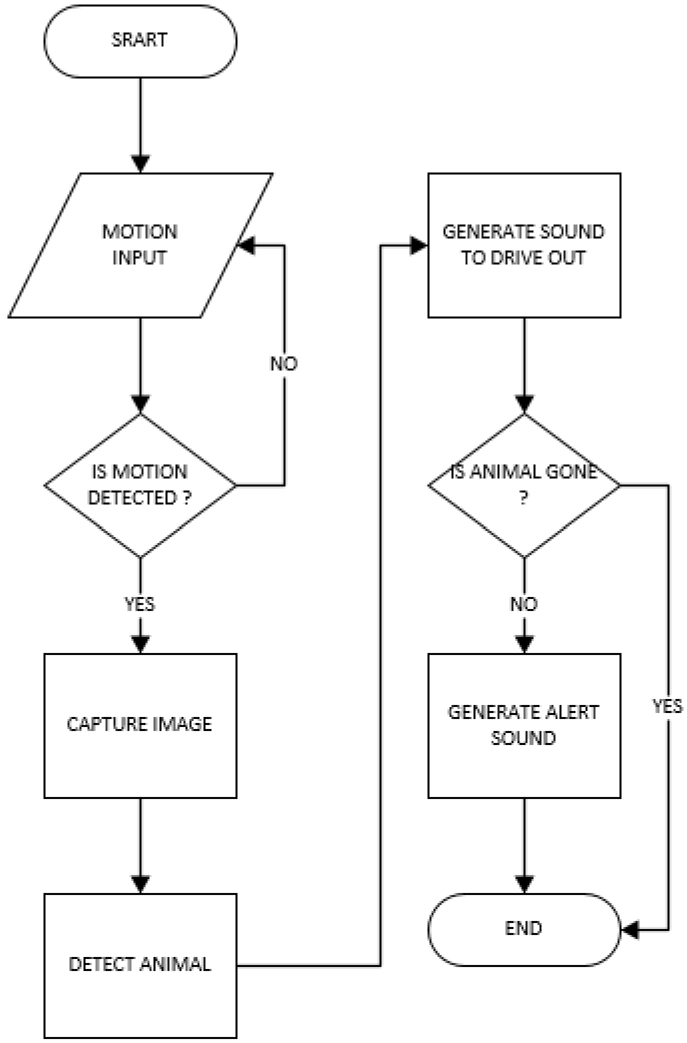


Figure 9. 3: flow chart for the system

Use case diagram

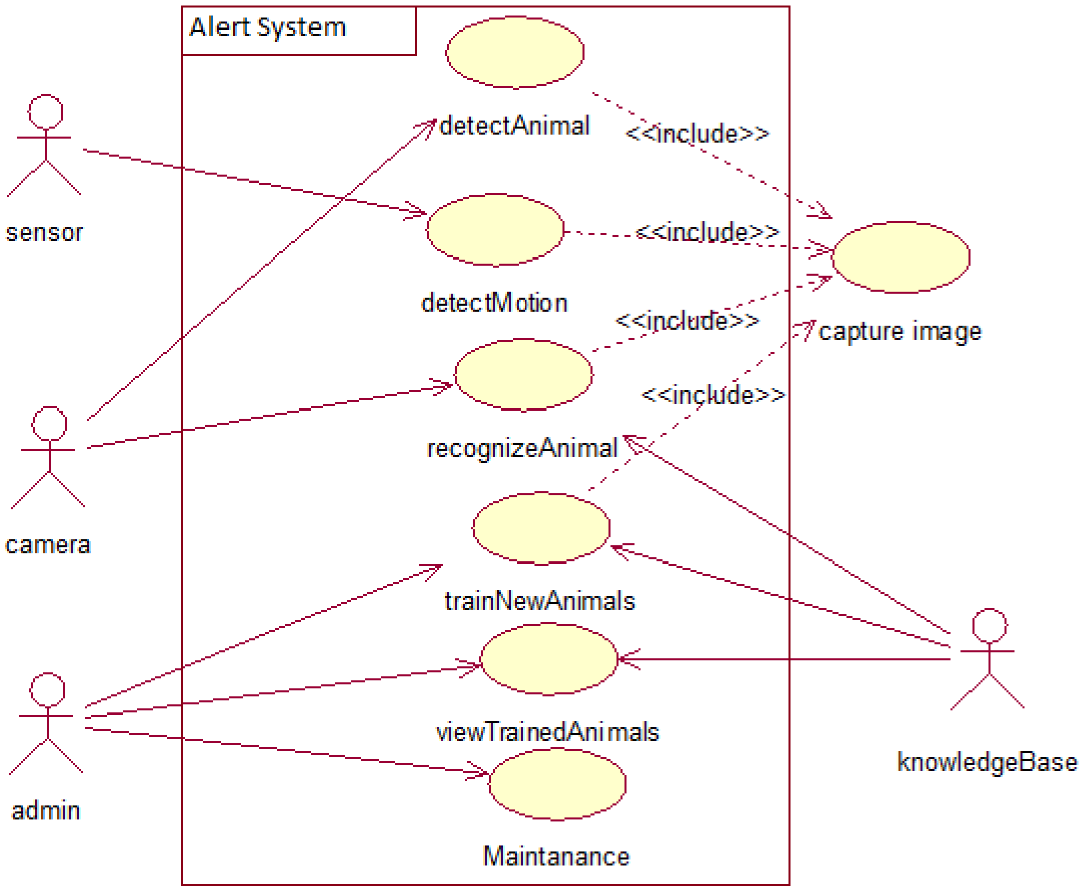


Figure 9. 4: use case diagram for the system

Activity diagram

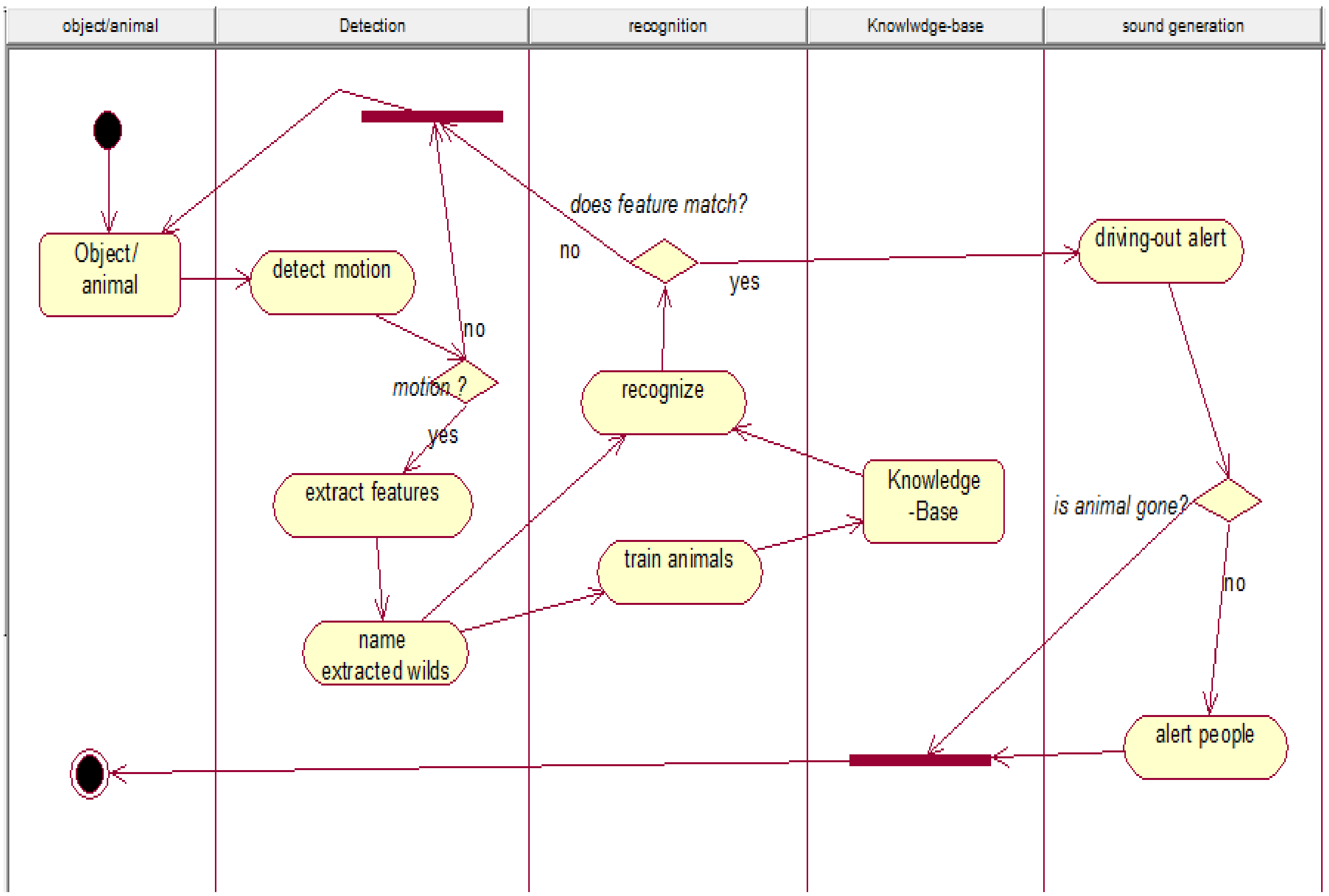


Figure 9. 5: showing activity diagram for the project

Class Diagram

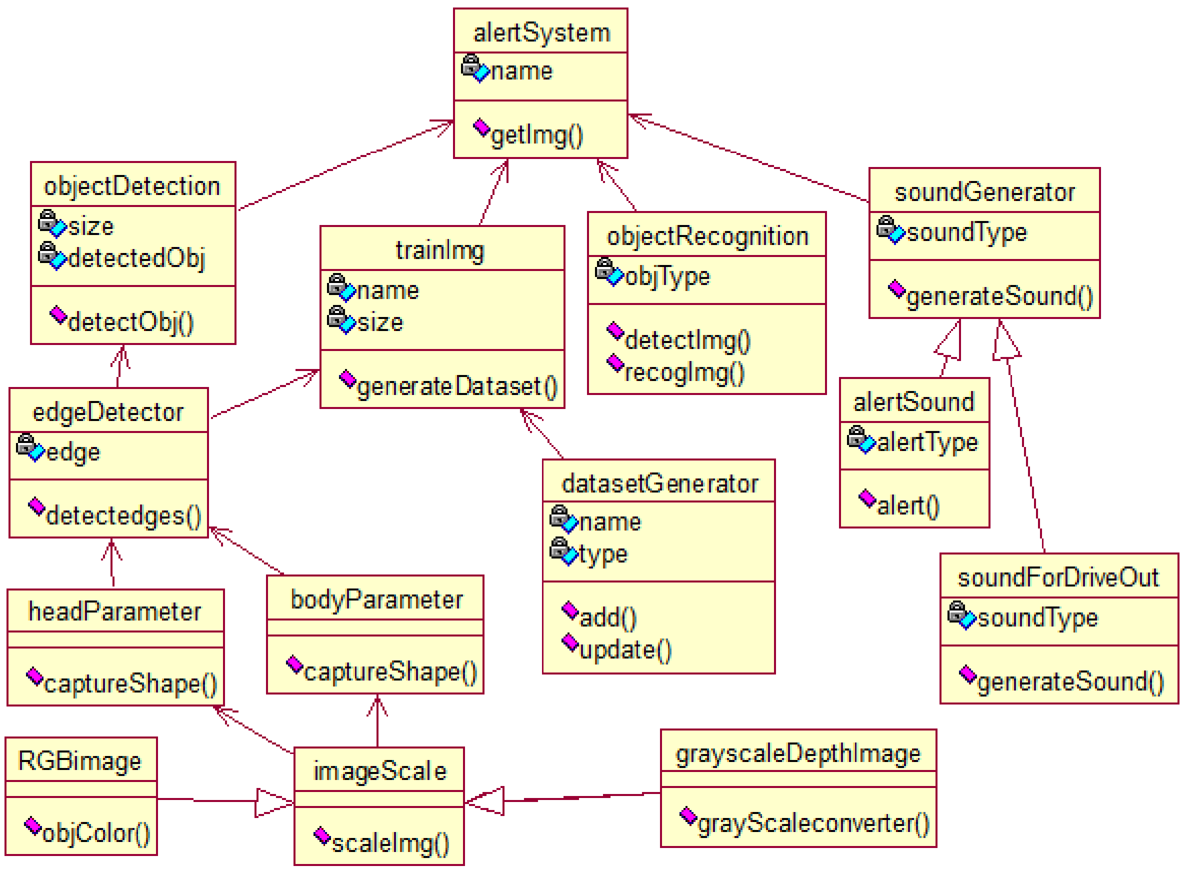


Figure 9. 6: showing class diagram

Context diagram

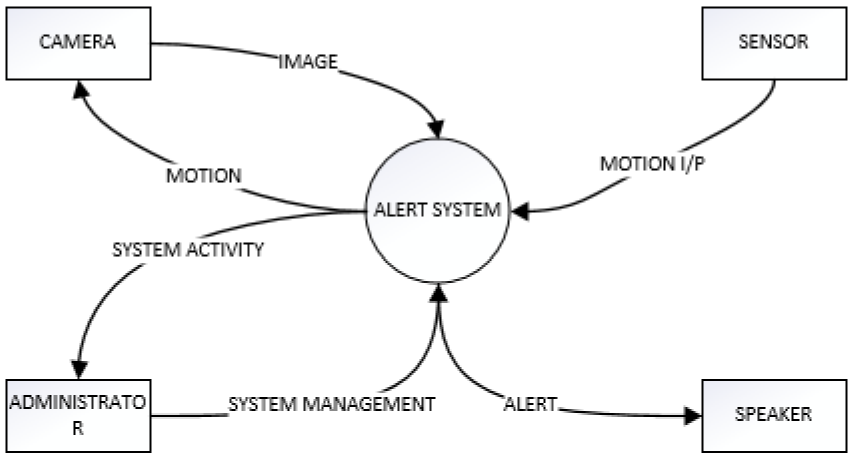


Figure 9. 7: context level DFD

Level 1 DFD

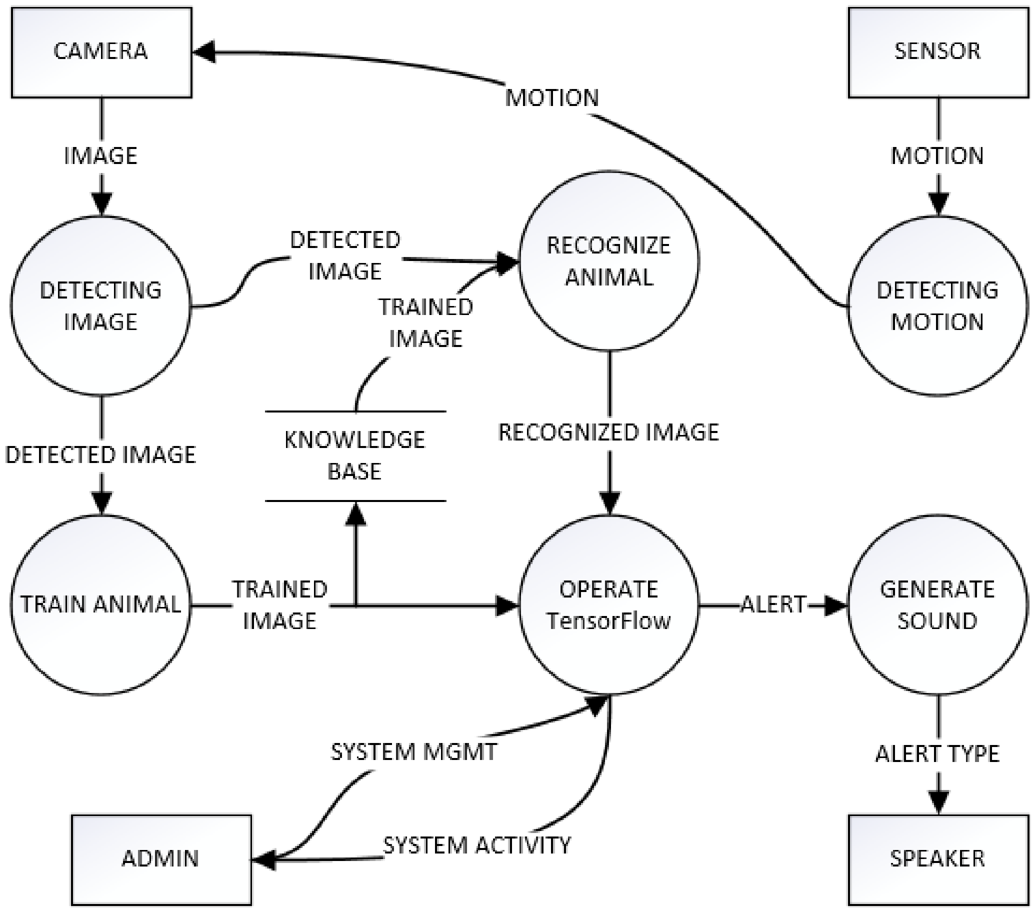


Figure 9. 8: level-1 DFD for the system

9.3.1 INCREMENT-1: object detection

Object detection involves in detecting instances of objects from a particular class in an image captured. The goal of object detection is to detect all instances/features of objects from a known class, such as people, cars or faces in an image. Usually, only a small number of instances of the object are present in image, but there are various possible locations and scales at which they can occur and that need to somehow be expressed.

The goal was to produce a working system for tracking objects in 3-dimensional space. The aim of this iteration was to begin from this spec and design a solution to the problem. After approximate models were designed, the task came to implement the models. Throughout the iteration, many problems arose. These problems were varied from performance issues related to the code, and from implementation issues related to the limitations of software technologies used in the project. All attempts at overcoming these problems are discussed in this section. TensorFlow library is used to detect objects in the image.

A. Analysis

In this phase, all the requirements for detecting objects were gathered. The required environments were installed on PC. The requirements were analyzed whether it is possible to implement or not. After analysis, planning for the increment was done and tasks for the iteration were divided. Finally, the design concept for the increment was figured out.

Table 9. 4: showing planning for increment-1

Activity	Period	Comment
Requirement Gathering	39d	Requirement gathering has took placed through searching on internet & by taking the information from various animal conservation related organizations and taking the ideas, sharing the views among group members.
Planning	37d	Planning has done by Reviewing of literatures, existing systems and other similar systems on internet, by taking the walkthrough.

Design	36d	Designing has done by creating UML diagram and by creating Charts
Coding & Implementation	34d	Implementation has done by creating module by module.
Testing	39d	Testing has done by perfuming unit testing, black-box testing and system testing.

B. Design

Design is the first step in the software development phase for any tools, techniques and principles for the purpose of defining a device, a process or a system. Once the software requirements and planning's have been analyzed and specified, the software development involves three major activities design, coding, implementation and testing that are required to build & verify the software. Here are some model diagrams as followings,

Flowchart

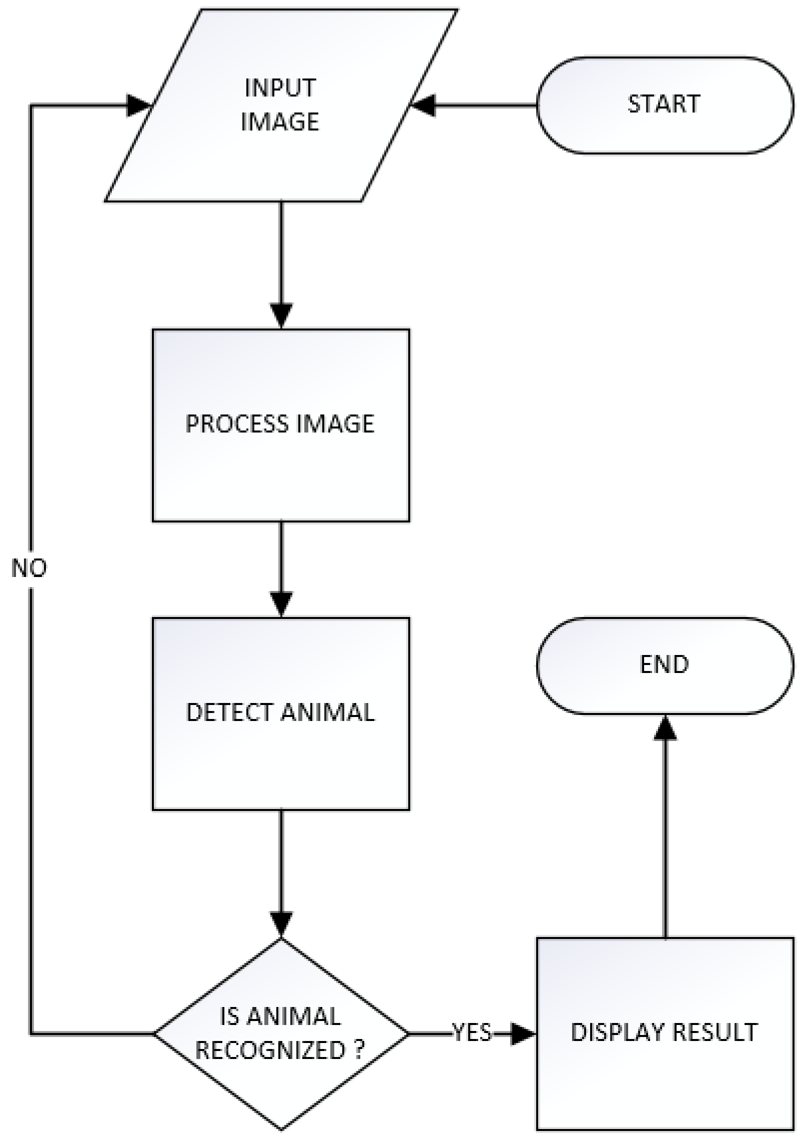


Figure 9. 9: flowchart for animal detection

Use case diagram

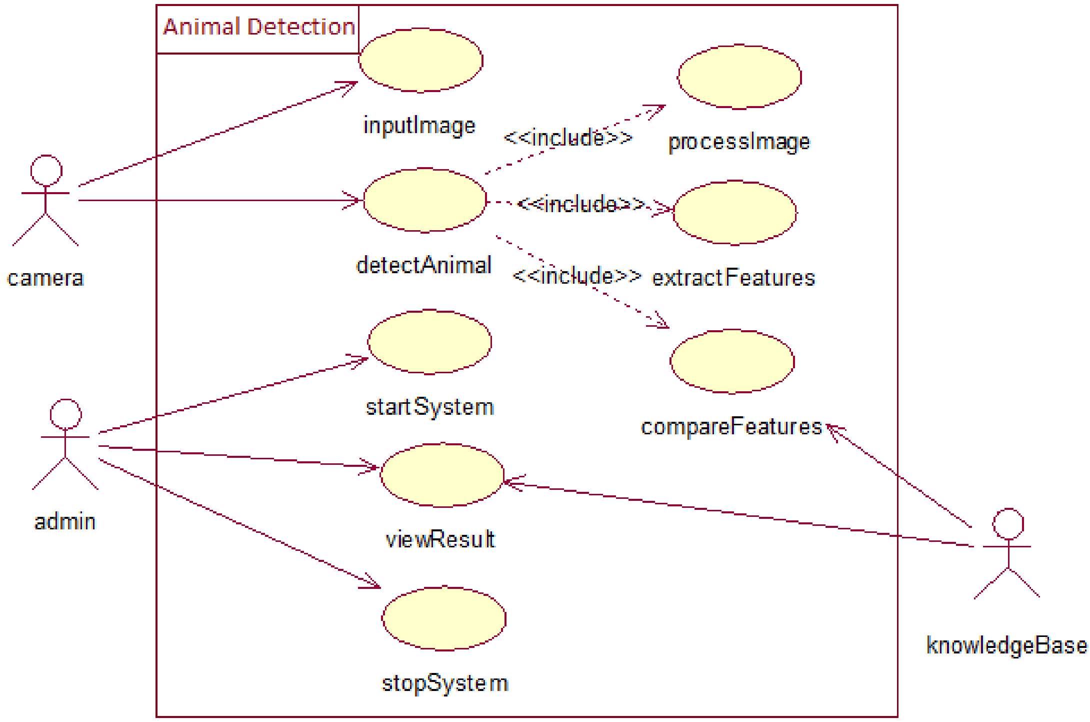


Figure 9. 10: use case diagram for animal detection

Level-1 DFD

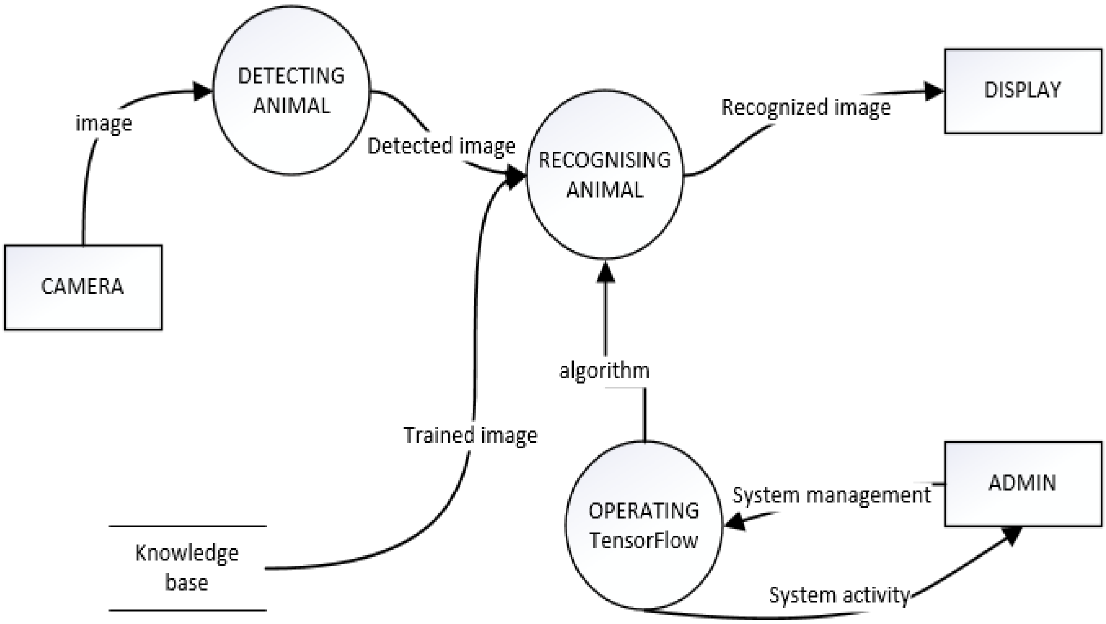


Figure 9. 11: level-1 DFD for animal detection

C. Coding

In this phase coding was done as per the design and a working system was developed by the end of this process.

D. Testing

At this section of work the testing strategies are applied and discussed the result. The unit testing approach was applied to debug the coding errors, the white-box testing approach was done to test whether the system gives output or not and finally system testing was done to test the The result is shown below,

Table 9. 5: showing test results

Data sets	iterated object	No of successfully detected object	% correct result
Centered light	10	10	100%
Left light	15	15	100%
Right light	12	12	100%
Covered object	10	3	30%
Unveiled/uncovered object	10	9	90%
Fast moving object	20	8	40%
Similar objects	20	16	80%
White background	20	19	95%
Random background	20	15	75%
Multiple objects	10	8	80%

9.3.2 INCREMENT-2: image training

For training to work as we expect, we should collect at least a hundred photos in every possible surrounding condition of an object we want to recognize. The more we collect, the better the accuracy of our trained model is likely to be. we also need to make sure that the photos are of good representation of what our application will actually encounter. For example, we take all our photos indoors and our users are trying to recognize objects outdoors, we probably won't get good results. So, by keeping this in mind we have done the trainings & the discussions and results were discussed in this section.

A. Analysis

In this phase, all the requirements for training the objects were gathered. The required environments were installed on PC. The requirements were analyzed whether it is possible to implement or not. After analysis, planning for the increment was done and tasks for the iteration were divided and concept for the design are gathered.

Table 9. 6: showing planning for increment-2

Activity	Period	Comment
Requirement Gathering	13d	Requirement gathering for training has took placed through &sharing the views among group members.
Planning	16d	done by Reviewing of literatures, existing systems and other similar systems.
Design	16d	done by creating UML diagram and by creating Charts
Coding & Implementation	30d	Done by creating module by module.
Testing	63d	Testing has done by perfuming various tests strategies.

B. Design

Once the software requirements and planning have been analyzed and specified, the software development involves three technical activities design, coding, implementation and testing that are required to build and verify the software. Here are some model diagrams as followings,

Flowchart

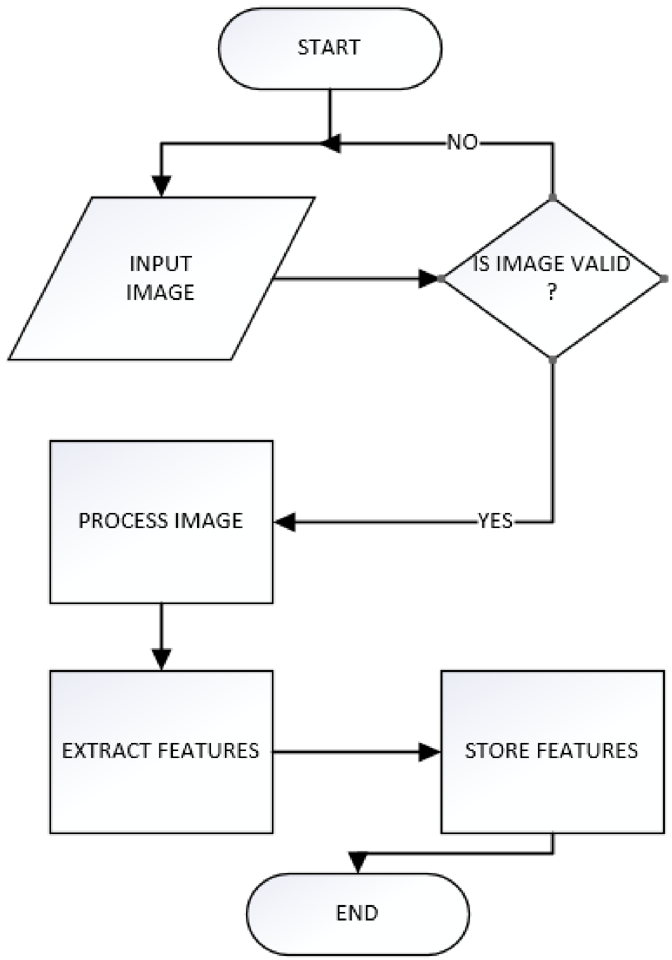


Figure 9. 12: flowchart for training animals/objects

Use case diagram

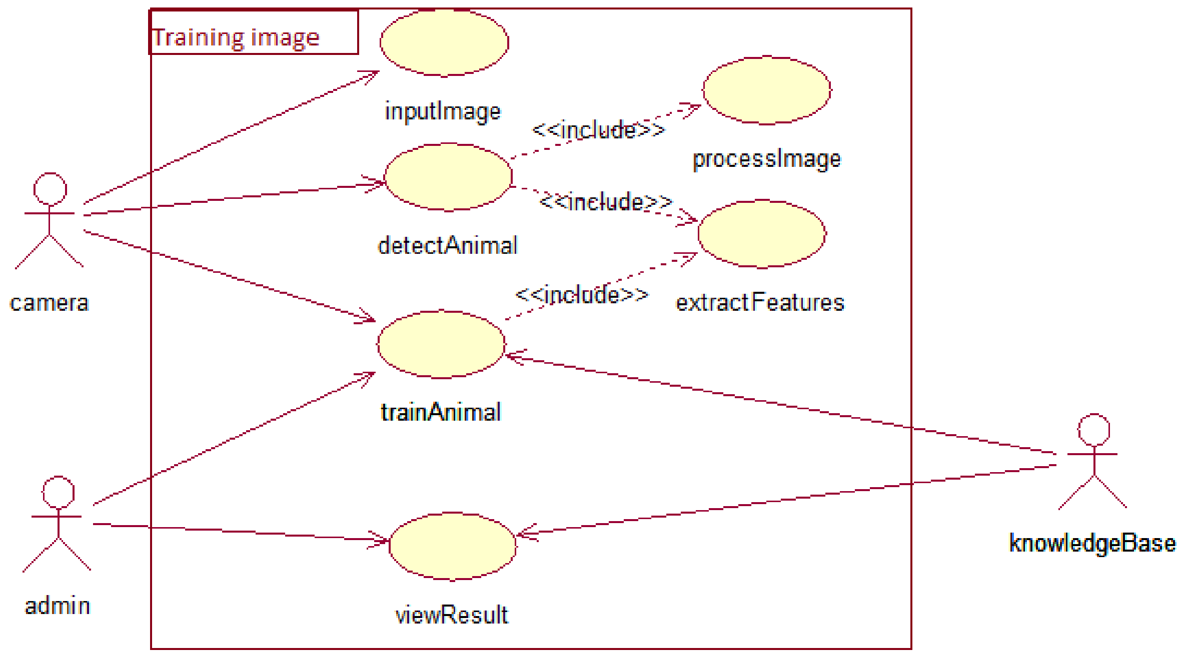


Figure 9. 13: use case diagram for training animals

Level-1 DFD

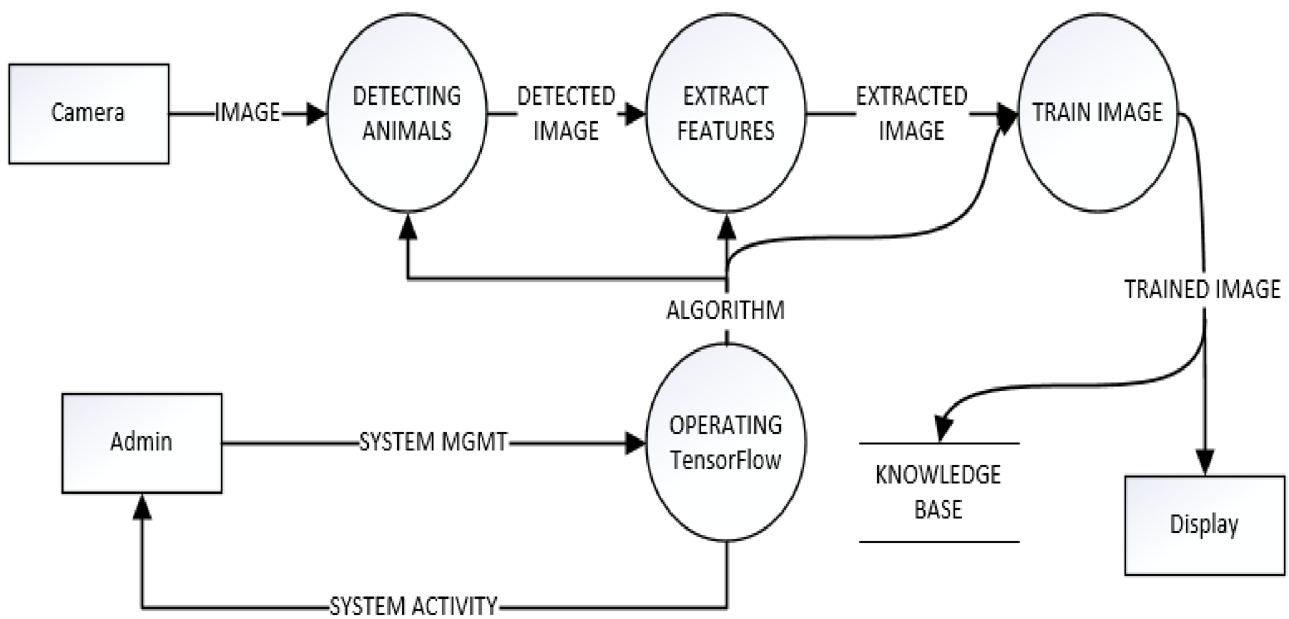


Figure 9. 14: level-1 DFD for training animal

C. Coding

In this phase coding was done as per the design and a working system was developed by the end of this process.

D. Testing

At this section of work, the different testing strategies are applied and the result are discussed. Here we have done testing on the training sets in different modes so that the accuracy ca be increased.

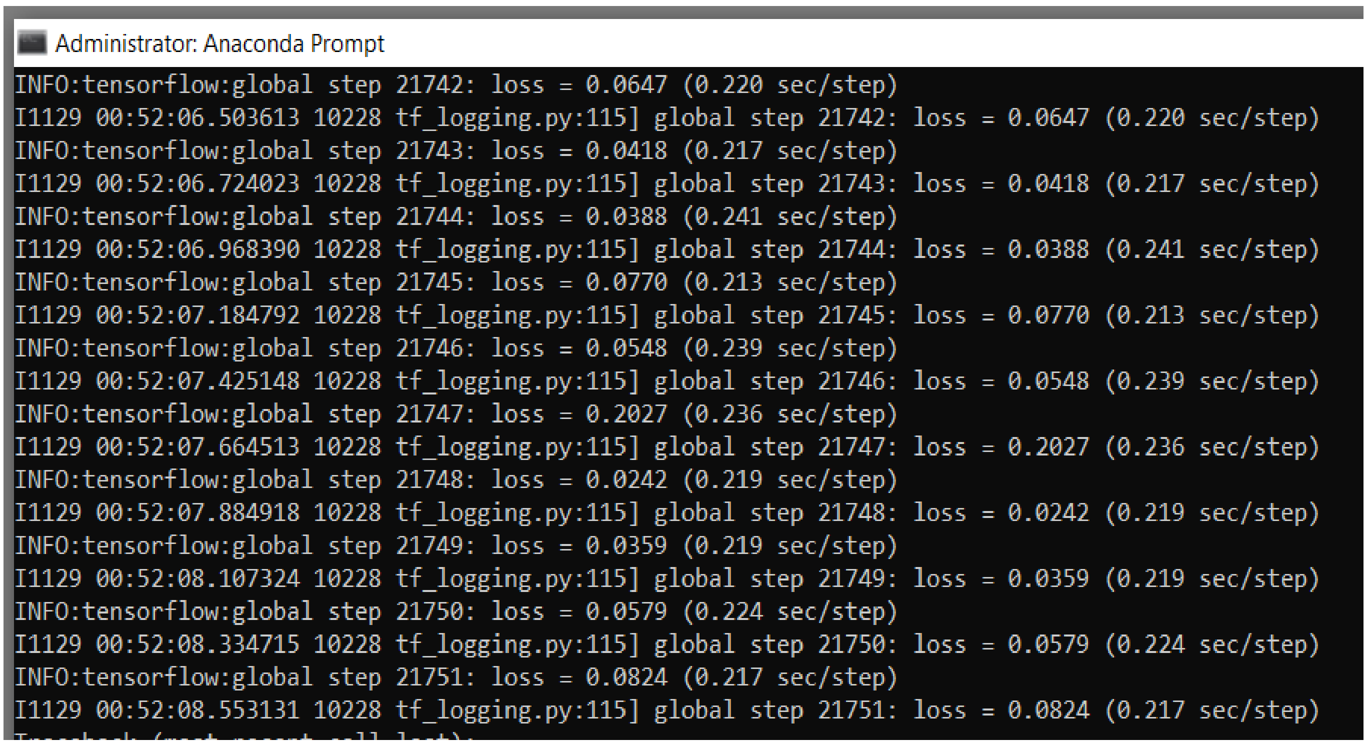


Figure 9. 15: training Log

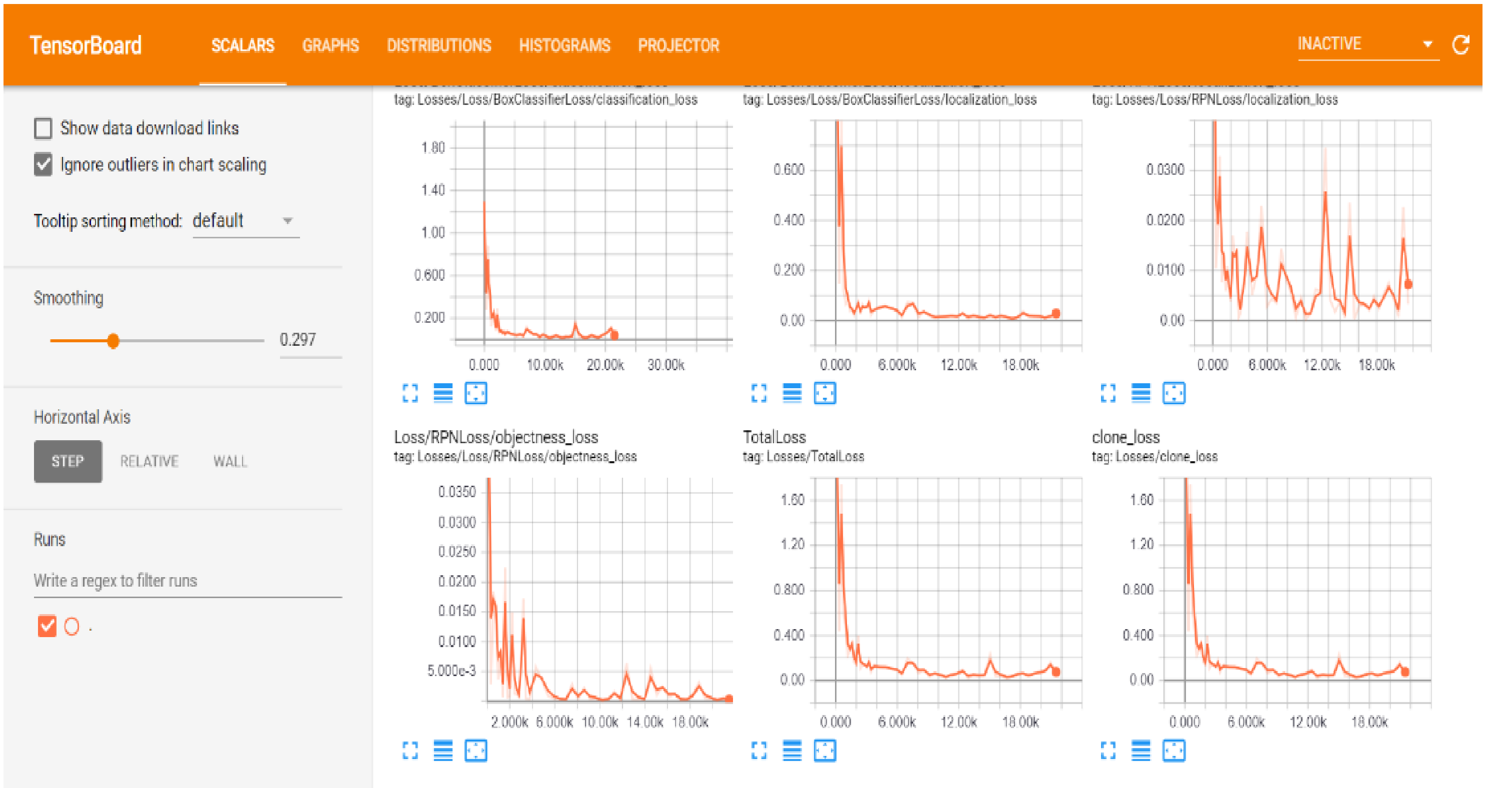


Figure 9. 16: Loss graph

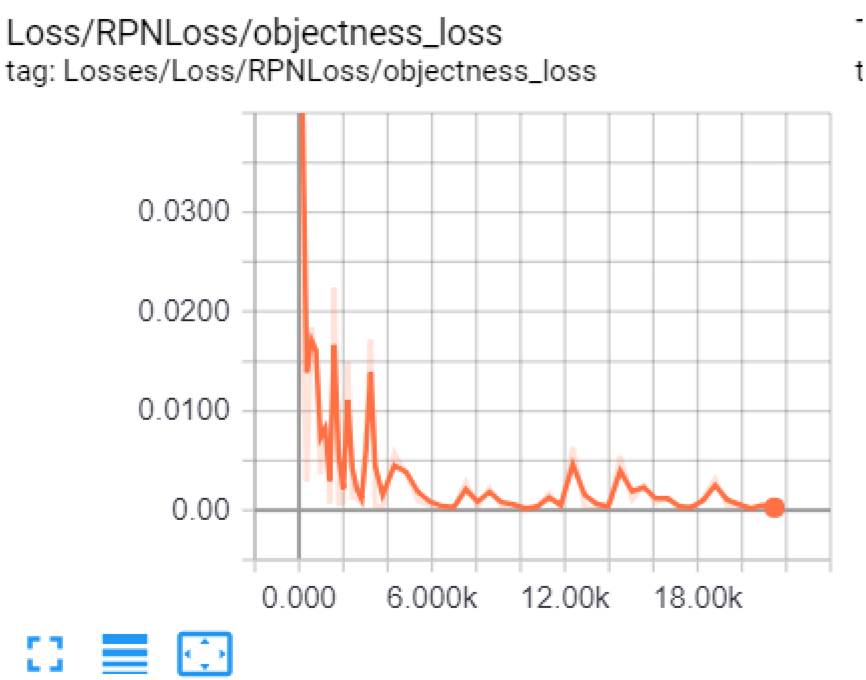


Figure 9. 17: loss vs steps graph

9.3.3 INCREMENT-3: hardware implementation

Hardware implementation involves various planning, task, cost & time estimations, coding and testing. Here in this increment of the project, we have gathered various requirements about the hardware devices and their interconnections. After gathering of the requirements, we designed the device and implement the code that have already prepared in pc console. Finally, we tested the system device whether it works or not.

A. Analysis

Here, all the requirements for h/w implementations were gathered. The requirements were analyzed whether it is possible to implement or not. After analysis, planning for the increment was done and tasks for the iteration were divided. Finally, the design concept for the increment was gathered.

Table 9. 7: showing planning for increment-3

Activity	Period	Comment
Requirement Gathering	14d	Requirements were gathered, estimated and the views were shared among group members.
Planning	15d	Planning has done by Reviewing of literatures, existing systems and other similar systems on internet, by taking the walkthrough.
Design	12d	Designing has done by creating diagrams and Charts
Coding & Implementation	12d	Implementation has done by integrating the hardware components.
Testing	36d	Testing has done by perfuming system testing.

Table 9. 8: showing hardware cost estimations

SN	Components	Quantity	Rate	Price (Rs)
1	Pi model 3 B^+	1	3920	3920
2	Arduino YUN	1	8400	8400
3	HC-SR04	2	440	880
4	Pi camera module	1	3575	3575
5	Jy-25	1	1050	1050

Hardware Description

Pi model 3 B+

The Raspberry Pi 3 Model B+ contains a wide range of improvements and features that will benefit the designers, developers, and even engineers who are looking to integrate Pi systems into their products. Here are some of the Pi model 3 B+'s specs:

- Quad core 64-bit processor clocked at 1.4GHz
- 1GB LPDDR2 SRAM
- Dual-band 2.4GHz and 5GHz wireless LAN
- Bluetooth 4.2 / BLE
- Higher speed ethernet up to 300Mbps
- Power-over-Ethernet capability (via a separate PoE HAT)

Figure 9. 18: Pi model 3 B+

pi camera module v-2

The Raspberry Pi Camera Module v2 is a high quality 8-megapixel custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464-pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. Pi camera attaches to Pi by way of one of the small sockets on the board (upper surface) and uses the dedicated C Si interface, designed especially for interfacing to pi cameras.

Figure 9. 19: Pi camera module v.2

Arduino YUN

AVR Arduino Microcontroller

- -ATmega32u
- -Input Voltage: 5V
- -20 Digital I/O Pins
- -7 PWM Channels
- -12 ADCs
- -16MHz Clock Speed
- -32 KB Flash Memory

Figure 9. 20: Arduino YUN

B. Design

Designing is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the hardware requirements have been analyzed and specified the software design involves three technical activities design, coding, implementation and testing that are required to build and verify the hardware.

C. Coding

In this phase coding was done as per the design and a working system was developed by the end of this process.

D. Testing

At this section of work the testing strategies are applied and discussed the result. The unit testing approach was applied to debug the coding & connection errors, the white-box testing approach was done to test whether the system gives output or not and finally system testing was done to test the system. Hence the system device tested successfully.

9.3.4 INCREMENT-4: integration of the project

After working on the various parts of the project separately we integrated them as one to give out the final product. We have done the various task such as object detection (recognizing the object specification), image training for the purpose of detecting the object as well as hardware implementation of the project.

The foremost work of our project is to identify the object for the detection of an object to carry out the further task. Object detection involves the detecting instances of an object from a particular class in an image that has been captured in real time. The main objective of an object detection is to detect the features of an object and recognize it. Image training is another important part of the project.

To detect the object firstly some input should be given to detect which type of object is present. Images of animals are given as input for our project. For the image training we have collected at least hundred pictures of the specific object from every possible angle and surrounding so that they can be recognized in any random conditions. The large number of pictures of a specific object from different perspective has helped to increase the object detection accuracy of our system in a great amount.

As we proposed we need a device. So, we have to implement our works in hardware components to achieve the project goal.

All these task that were carried out separately are integrated at final stage to provide the final system. After achieving the final product, we ran the test for the system to check whether it performs as expected or not. We performed various test for different possible situation to find out the results. The results obtained from all these tests were discussed among the team members and supervisor.

A. Analysis

Here, all the requirements for increment integration were gathered. The requirements were analyzed whether it is possible to implement or not. After analysis, planning for the final increment was done and tasks for the iteration were divided.

B. Integration Testing

We integrated all the task that were done separately to give out the final product. The integrated product was monitored and tested. The result achieved through the test were discussed and the conclusion was drawn from it. We obtained the output from the system and found the system is working. Though the system is working but process is slow which is limiting the scope of project.

Due to the limited processing power of Raspberry Pi, real time object detection process couldn't be achieved as expected. Not enough memory space also contributed to the slow processing resulting to the disappointment in the expected outcome of object detection. High resolution picture led to the decrease in the frame rate of the system and vice versa. But when the resolution was decreased the detail of the picture could not be reached. So, another limitation was encountered when the expected frame rate per second was not achieved by the system.

Sound generation process encountered the error causing incompleteness to the system. It was also difficult to integrate the motion sensor with our product. The main theme of our project to alert the people through sound generation was achieved in the desktop but we found it very difficult to implement in the hardware and couldn't achieve it as a final product.

10. TASK DISTRIBUTION

Table 10. 1: showing task division

Member	Activity	Comment
M1-M5	Requirement Gathering	M2, M1 and M4 has performed the searching for project requirement on the internet by reviewing the related literature and by analyzing the related project which is already available in the market. Regularly inform to the other member of team.
M1-M5	Analyzing Of the requirement	Whole team done the requirement analyzing of project by sharing the ideas, and by discussing on related information which is gather by the M1, And M2 and M3. M1 and M3 has created the list of requirements after every meeting
M1-M5	Finalizing requirements	Whole team finalize the requirement. M1, M2 and M3 has created a list of finalize requirement.
M3, M4	Installation of tools and technology for object detection	M3 and M4 installed the all the require tools which is use for object detection. And M1 installed required UML and diagrammatic tools. Also informed to M1.
M1, M4	Implementing TensorFlow Library to detect object	M1 performs various modeling diagrams and M4 implements the TensorFlow library to detect object (i.e. wild animals) and informed to M1 & M3
M4	Testing of object detection	M3 done various test techniques and gathered different test cases. Also informed to M1.

M3	Installation of tools and technology for training	M3 installed all the require tools which is use for training objects. And M1 installed UML and other diagrammatic tools.
M1, M3	Implement training module	M1 design model diagrams and M3 implemented the training algorithm (i.e. TensorFlow library) and discuss on it with other team members.
M3	Implement of Dataset & trainer module	M3 implemented the dataset module and discussed to the other team members.
	Testing and verifications of trained dataset	M3 tested the implemented module in various cases and discussed with other members. Also informed to M1.
M5, M1	Gathering requirements for hardware implementation	M5 and M1 gathered all requirements to implement hardware for the project and informed to other members.
M5, M1	Integrating Hardware modules	M5 and M1 integrated all the modules and discuss on it with M3 and to each other regularly.
M5	Integrated hardware testing	M5 performed the unit testing and noted down results and discuss with M1 and other members of team.

M3, M4, M5	Requirement gathering	M3, M4 and M5 gathered requirements for project integration and discussed with other members.
M1- M5	Implementing system integration	Whole members were gathered for integration and system integration was done successfully
M1-M5	Unit testing	All team members performed the functional testing and noted down results and discussed the result of testing with each other.
M1-M5	Black box testing	All team members performed black box testing and discussed the result
M1-M5	System testing	Performed system testing and result was discussed with each other.
M1, M2	documentation	M1 perform documentation portion of the project and M2 helped M1 at each phase.
M1= Amrit Sunar **M2= Binita Rai** **M3= Prakash Yadav** **M4= Rabin Rai** **M5= Suraj Suwal**		

11. DELIVERABLES

- A device that can detect and recognize wild animals.
- The device that can generate sounds to drive out those animals.
- The device that can generate the alert sounds to alert the peoples.
- project document and related demos. Furthermore, answers for the queries.
- review, feedback and overall judgement on the project.

12. PROJECT TASK AND TIME SCHEDULE

The time schedule has been designed as per the requirements. Requirement analysis has been given more emphasis. System Design is to be done first. Testing and Debugging is to be done along with the development of the project. Finally, documentation is done throughout the project.

ID	Task Name	Start	Finish	Duration
1	Requirement gathering for object detection	6/22/2018	8/15/2018	39d
2	Planning for task and schedule	6/29/2018	8/20/2018	37d
3	Designing model diagrams	7/12/2018	8/30/2018	36d
4	Coding and implementation	7/26/2018	9/11/2018	34d
5	Testing and verification	7/30/2018	9/20/2018	39d
6	Documentation	6/29/2018	9/24/2018	62d

Figure 12. 1: Showing planned duration for tasks of increment-1

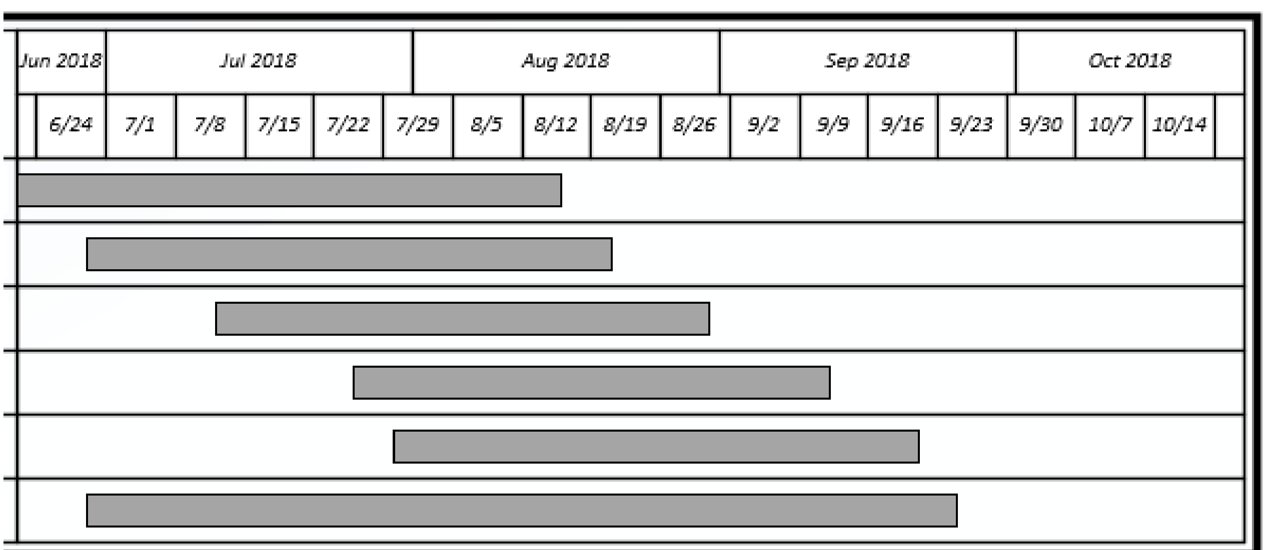


Figure 12. 2: showing Gantt chart for increment-1

ID	Task Name	Start	Finish	Duration
1	Requirement Gathering for training objects	8/16/2018	9/3/2018	13d
2	Planning for task division, schedule & training	8/20/2018	9/10/2018	16d
3	Designing and architecting model diagrams	8/27/2018	9/17/2018	16d
4	Coding and implementing	9/3/2018	10/12/2018	30d
5	Testing and verification	9/5/2018	11/30/2018	63d
6	Documentation	8/28/2018	11/28/2018	67d

Figure 12. 3: Showing planned duration for tasks of increment-2

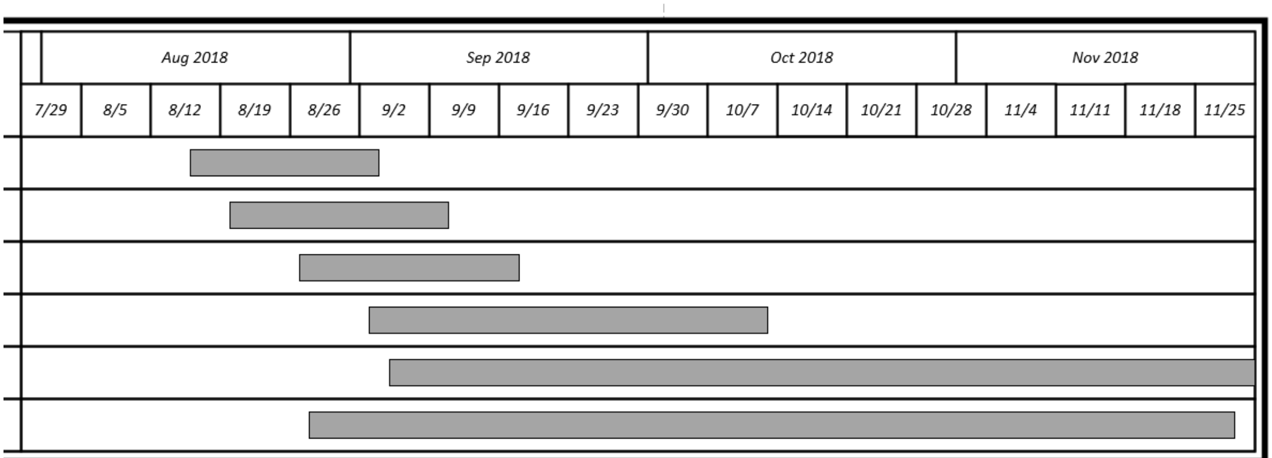


Figure 12. 4: showing Gantt chart for increment-2

ID	Task Name	Start	Finish	Duration
1	Requirement gathering for h/w implementation	9/24/2018	10/11/2018	14d
2	Planning	10/8/2018	10/26/2018	15d
3	Designing and architecting	10/12/2018	10/29/2018	12d
4	Connection and implementation	10/18/2018	11/2/2018	12d
5	Testing and verification	10/25/2018	11/9/2018	12d
6	Documentation	9/28/2018	11/16/2018	36d

Figure 12. 5: showing task schedule for increment-3

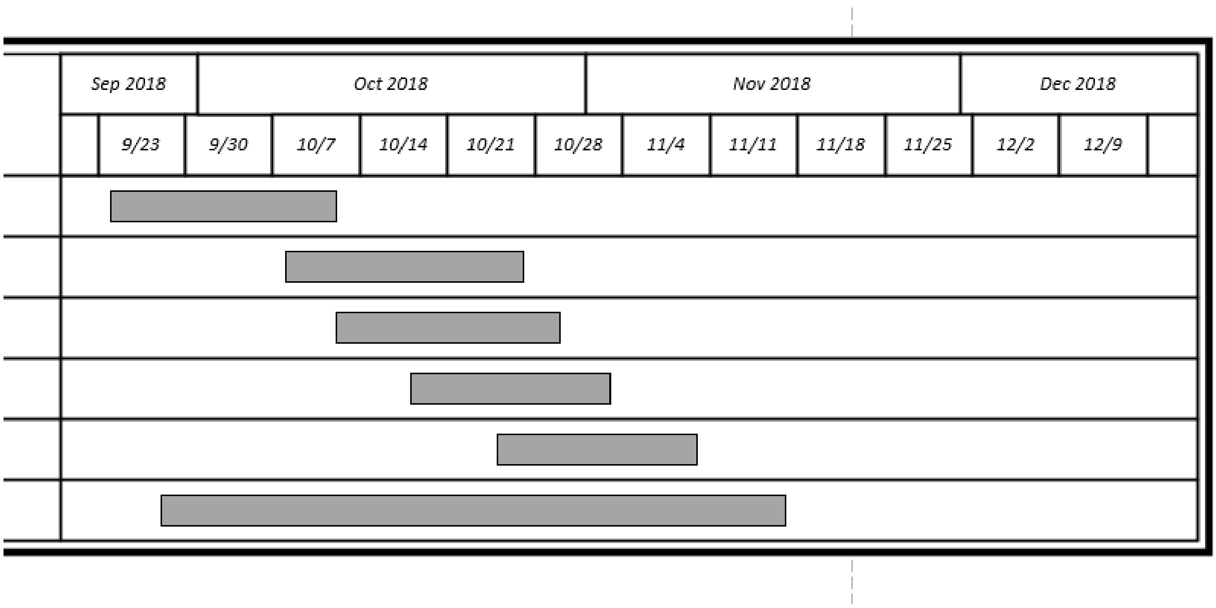


Figure 12. 6: showing Gantt chart for increment-3

ID	Task Name	Start	Finish	Duration
1	Requirement gathering for integration	11/6/2018	11/13/2018	6d
2	Planning	11/9/2018	11/19/2018	7d
3	Design and disscuss	11/16/2018	11/26/2018	7d
4	Implementation	11/22/2018	11/30/2018	7d
5	Testing and verification	11/26/2018	12/3/2018	6d
6	Documentation	11/12/2018	12/6/2018	19d

Figure 12. 7: showing task schedule for increment-4

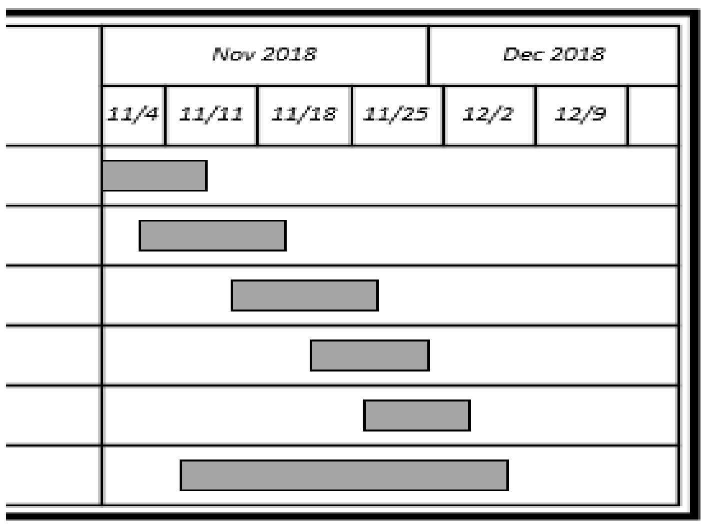


Figure 12. 8: showing Gantt chart for increment-4

13. CONCLUSION AND FUTURE SCOPE

13.1 CONCLUSION

In order to monitor and alert the peoples this system has been proposed. It replaces the manual system with an automated system which is fast, efficient, cost and time saving as replaces the continuous, tedious and manual monitoring and tracking of wild animals.

Although we had tried our best to achieve the expected result, we were still bounded by some limitations. Due to the limited processing power of Raspberry Pi, real time object detection process couldn't be achieved as expected. Not enough memory space also contributed to the slow processing resulting to the disappointment in the expected outcome of object detection. High resolution picture led to the decrease in the frame rate of the system and vice versa. But when the resolution was decreased the detail of the picture could not be reached. So, another limitation was encountered when the expected frame rate per second was not achieved by the pi.

Hence this system is expected to give desired results and in future could be implemented in other various fields. Also, the efficiency could be improved by integrating other techniques with it in near future.

13.2 FUTURE SCOPE

It can be easily implemented at any places where there is danger from wild animals. A method could be proposed to illustrate robustness against the variations that is, in near future we could build a system which would be robust and would work in undesirable conditions too. Here it is proposed for an area where there is some danger from wild animals but in future it can be used to do the same work at various areas that has same problem. Authors are working to improve the object detection and recognition effectiveness to build more efficient systems in near future.

In the future if the processing power as well as memory of Raspberry Pi is increased then we can obtain the expected outcomes. Also, if the TensorFlow is made compatible with the Raspberry Pi, we can carry this project to have better performance.

14. BIBLIOGRAPHY

[1]. *https://en.wikipedia.org/wiki/Wildlife_of_Nepal* [accessed on 28-06-2018]

[2]. Zhongjie Li and Rao Zhang, *"Object Detection and Its Implementation on Android Devices",* feb 2017.

[3]. Krishna Prasad Acharya, Prakash Kumar Paudel, Prem Raj Neupane, Michael Köhl *"Human-Wildlife Conflicts in Nepal: Patterns of Human Fatalities and Injuries Caused by Large Mammals",* Sept 9-2016.

[4]. *https://en.wikipedia.org/wiki/Object_detection* [accessed on 29th jun 2018]

[5]. Andrea Ferri, *"Object Tracking in Video with TensorFlow"* October 2016.

[6]. Marcel P. Huijser and Patrick T. McGowen, *"Overview of Animal Detection and Animal Warning Systems in North America And Europe"* 2003

[7]. https://www.tensorflow.org/tutorials/image_retraining [accessed on 30-06-2018]