VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590014

A Project Report on

"Crop Guard - Animal Detection in Crop Fields"

Submitted in partial fulfilment of the requirements for the award of degree

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Prathamesh Chougule	**2AG19CS047**
Shrinath Korajkar	**2AG19CS073**
Shriram Hebbar	**2AG19CS074**
Tejashwini V Rathod	**2AG19CS084**

Under the Guidance of

Prof. Dhanashree Kulkarni

Assistant Professor and Head,

Dept. of CSE, AITM, Belagavi.

ANGADI INSTITUTE OF TECHNOLOGY & MANAGEMENT

BELAGAVI-590009

2022-2023

ANGADI INSTITUTE OF TECHNOLOGY & MANAGEMENT

BELAGAVI -590009

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that Project entitled **"Crop Guard - Animal Detection in Crop Fields"** is bonafide work carried out by **Prathamesh Chougule (2AG19CS047), Shrinath Korajkar (2AG19CS073), Shriram Hebbar (2AG19CS074)** and **Tejashwini V Rathod (2AG19CS084)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Computer Science & Engineering under Visvesvaraya Technological University, Belagavi** during the year 2022-2023. It is certified that all the correction/suggestion indicated for internal assessment have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of major project work prescribed for the Bachelor of Engineering degree.

____________________	____________________	____________________	____________________
Signature of the Guide	Signature of Co-Ordinator	Signature of the HOD	Signature of the Principal
Prof. Dhanashree Kulkarni	Prof. Gautam Dematti	Prof. Dhanashree Kulkarni	Dr. Anand Deshpande
Assistant Professor and Head, Dept. of CSE, AITM	Assistant Professor, Dept. of CSE, AITM	Assistant Professor and Head, Dept. of CSE, AITM	Principal and Director, AITM, Belagavi

Name of the Examiner: **Signature with date:**

1.______________________________ ____________________

2.______________________________ ____________________

DECLARATION

We **Prathamesh Chougule (2AG19CS047), Shrinath Korajkar (2AG19CS073), Shriram Hebbar (2AG19CS074),** and **Tejashwini V Rathod (2AG19CS084)** studying in the 8^{th} semester of Bachelor of Engineering in Computer Science and Engineering at Angadi Institute of Technology and Management, Belagavi, hereby declare that this project work entitled **"Crop Guard - Animal Detection in Crop Fields"** which is being submitted by us in the partial fulfilment for the award of the degree of Bachelor of Engineering in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi is an authentic record of us carried out during the academic year 2022-2023 under the guidance of **Prof. Dhanashree Kulkarni,** Assistant Professor and Head, Department of Computer Science and Engineering, Angadi Institute of Technology and Management, Belagavi.

We further undertake that the matter embodied in the dissertation has not been submitted previously for the award of any degree or diploma by us to any other university or institution.

Place: Belagavi

Date: 03/05/2023

Prathamesh Chougule

Shrinath Korajkar

Shriram Hebbar

Tejashwini V Rathod

ACKNOWLEDGEMENT

It is our proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation, and guidance.

First and foremost, we wish to record our sincere gratitude to **Management of Angadi Institute of Technology and Management, Belagavi** and to our beloved, **Dr. Anand Deshpande,** Principal and Director, Angadi Institute of Technology and Management, Belagavi for his constant support and encouragement in preparation of this report and for making available library and laboratory facilities needed to prepare this report.

We express our sincere gratitude to our guide **Prof. Dhanashree Kulkarni,** HOD, Department of Computer Science and Engineering, Angadi Institute of Technology and Management, Belagavi for guiding us in investigations for this project and in carrying out experimental work. Our numerous discussions with her were extremely helpful. We hold her in esteem for guidance, encouragement and inspiration received from her.

Our sincere thanks to **Prof. Gautam Dematti**, Project Coordinator for having supported the work related to this project. His contributions and technical support in preparing this report are greatly acknowledged.

The project on **“Crop Guard - Animal Detection in Crop Fields”** was very helpful to us in giving the necessary background information and inspiration in choosing this topic for the project.

Last but not the least, we wish to thank our **parents** for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

Prathamesh Chougule
Shrinath Korajkar
Shriram Hebbar
Tejashwini V Rathod

Vision of the Department
To impart quality education in computer science and engineering with emphasis on professional skills to meet the global challenges in IT paradigm.
Mission of the Department
M1: Impart knowledge in current and emerging computing technologies by adopting various pedagogical approaches. **M2:** Develop a conducive environment to inculcate and nurture analytical and communication skills along with societal and ethical responsibility in all professional endeavours. **M3:** Enable students to be successful globally by being effective problem solvers and lifelong learners.

Program Educational Objectives (PEOs)
Graduates will be able to: **PEO1:** Utilize the strong foundation in mathematics, programming, scientific and engineering fundamentals necessary to formulate, analyse and solve IT related engineering problems, and endeavour themselves for higher learning. **PEO2:** Demonstrate an ability to analyse the requirements and technical specifications of software to articulate novel engineering solutions for an efficient product design. **PEO3**: Adapt to emerging technologies, and work as teams on multidisciplinary projects meeting the requirements of Indian and multinational companies. **PEO4:** Pursue professional career adopting work values with a social concern to bridge the digital divide and develop effective communication skills and leadership qualities, and **PEO5:** Understand the efficacy of life-long learning, professional ethics, and practices, so that they may emerge as global leaders.

ABSTRACT

The project titled "Crop Guard - Animal Detection in Crop Fields" aims to develop a software tool that detects wild animals entering the farm or crop fields and alerts farmers using an alarm system. The objective of this project is to provide a simple, fast, and cost-effective solution to the problem of crop damage caused by animal attacks, which is a major threat to crop yield. To achieve this, the proposed system leverages an image enhancement technique based on LSGAN (Least Square Generative Adversarial Network) to generate more realistic and diverse training images. The enhanced dataset is then used to train an object detection algorithm YOLO (You Only Look Once) version 7 for animal detection. The proposed system monitors the entire farm using cameras that record the surroundings and play appropriate sounds when animals are detected to drive them away from the fields. The system aims to ensure the safety of both human and animal by diverting animals without causing them any harm. The project is an attempt to develop an economical, time-saving solution that supports farmers and helps protect their crops from damage caused by animals.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1. Problem Statement

Agriculture is a crucial sector that forms the backbone of most economies worldwide. It involves cultivating crops and rearing livestock to provide food and other essential products. In the case of crops, different varieties are grown, such as cereals, pulses, fruits, vegetables, and oilseeds, among others. The choice of crops is determined by factors such as climate, soil, and market demand, among others. However, one of the main challenges facing farmers is the intrusion of wild animals into crop fields, which poses a significant threat to crop yields.

1.1.1. Animal Intrusion into Crop Fields

Animals can enter crops for various reasons. In many cases, animals are looking for food or water, and they may be attracted to the crops that are growing in the fields. Animals such as deer, rabbits, and rodents are known to feed on crops such as corn, wheat, and soybeans. These animals can cause significant damage to crops, which can result in reduced yields and financial losses for farmers. In addition to feeding, animals may enter crops for shelter and protection from predators. Fields provide cover for animals to hide from predators, especially during the breeding season. Some animals also use crops as a pathway to move from one habitat to another. For example, deer may use fields to travel from wooded areas to open fields in search of food and water.

Human activities such as construction and urbanization have also contributed to the intrusion of animals into crop fields. As human settlements expand, they encroach upon natural habitats of animals, pushing them into agricultural areas in search of food and shelter. Furthermore, climate change has also played a role in the intrusion of animals into crop fields. Changes in temperature and precipitation patterns have altered the timing of crop growth, which can affect the feeding patterns of animals. As a result, animals may enter crop fields earlier or later in the season, causing more damage to crops than in the past.

Animal intrusion in crop fields can result in significant crop damage and low crop yield. Wild animals, such as deer, rabbits, and birds, can consume or damage the crops by feeding on the leaves, stems, and fruits of plants. This can lead to stunted growth, reduced yield, and sometimes complete destruction of crops. For example, deer can cause damage to crops by browsing on young trees and shrubs, while rabbits can chew on the stems and leaves of plants.

Similarly, birds can cause damage to fruits, such as grapes and cherries, by pecking and eating them. In addition to direct damage to crops, animal intrusion can also lead to other problems. For example, animals can trample on crops, damaging the soil and causing compaction, which can reduce water infiltration and root growth. Moreover, animal droppings and urine can contaminate the crops, leading to food safety concerns. Animal intrusion can also attract other pests, such as insects and rodents, which can further damage the crops.

1.1.2. Traditional Techniques

To address the problem of animal intrusion into crop fields, farmers have traditionally used various techniques. These techniques and their limitations are given in fig 1.1.2.1:

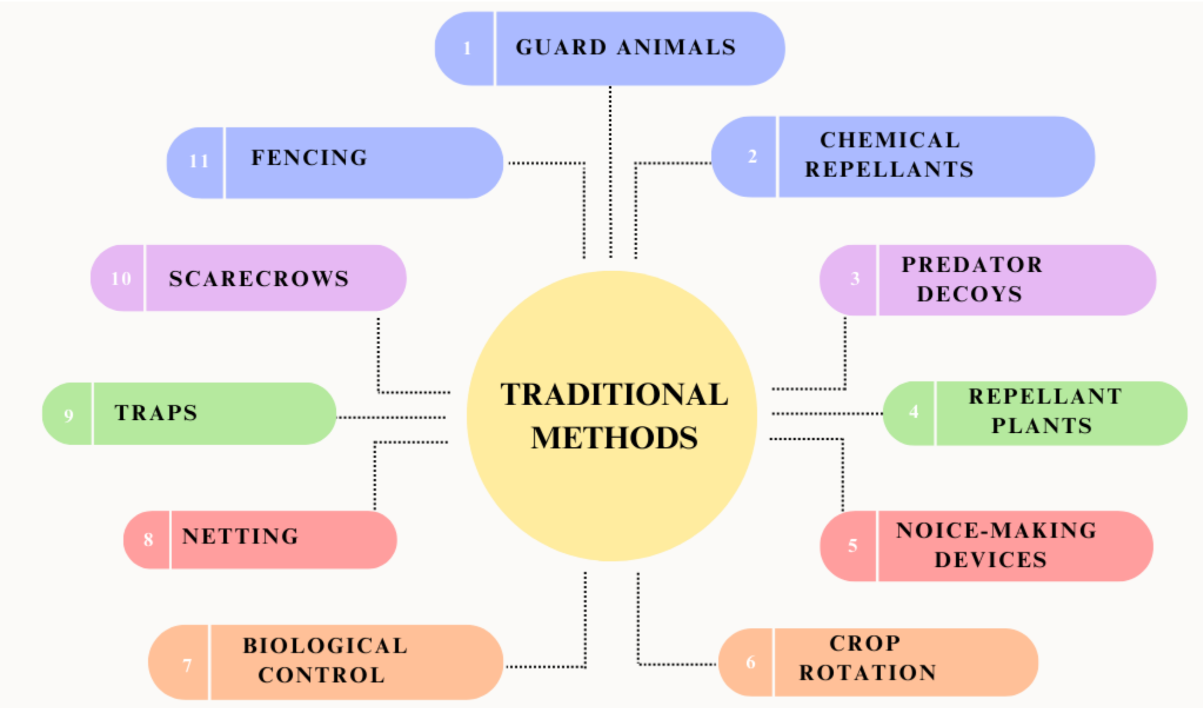

Fig 1.1.2.1. The traditional techniques used by farmers to prevent animals from entering crop fields

- **Fencing:** One of the most common methods of animal control in crop fields is fencing. There are several types of fences that can be used, including physical barriers such as barbed wire or electric fences, as well as sensory fences that use motion detectors or other sensors to detect the presence of animals. Fences can be expensive to install and maintain, and they may not be practical in all situations. In addition, animals can sometimes find ways to breach fences, such as by digging under them or climbing over them. Electric fences can start fires when the energizer is combined with the vegetation.

- **Scarecrows:** Scarecrows are commonly used to scare birds away from crops. They are usually made of straw and placed in fields to resemble human figures. However, scarecrows may not be effective against larger animals such as deer, and they require frequent maintenance.
- **Noise-making devices**: Another common technique for animal control in crop fields is the use of noise-making devices. These devices emit loud noises or flashing lights when triggered by the presence of animals, which can be effective in deterring them from entering the crop field. Some animals may become accustomed to the noise made by these devices and no longer be deterred by them. In addition, these devices may be annoying to humans who live or work near the crop field
- **Chemical repellents:** Some crop farmers use chemical repellents to deter animals from entering their fields. These repellents can be sprayed on the crops or applied to the perimeter of the field to create an unpleasant or toxic environment for the animals. Chemical repellents can be harmful to the environment and to non-target species, and they may not be effective against all types of animals. In addition, animals may become accustomed to the repellents over time and no longer be deterred by them.
- **Predator decoys:** Predator decoys, such as fake owls or coyotes, can be used to deter animals from entering crop fields. These decoys mimic the presence of natural predators, which can cause animals to avoid the area out of fear. Some animals may not be deterred by predator decoys, particularly if they are not familiar with the species being simulated. In addition, these decoys may need to be replaced or repaired over time, which can be costly.
- **Biological control:** Some farmers use biological control methods to control animal populations in their crop fields. This can involve releasing predators that prey on the animals that are causing damage, or introducing a disease or parasite that affects the target animal species. The use of biological control methods can be controversial, as they can have unintended consequences on the ecosystem. In addition, these methods may not be effective against all types of animals.
- **Netting:** Crop netting or bird netting can be used to physically exclude animals from entering a crop field. This method is often used to protect small fruit trees or berry bushes from birds and other small animals. Netting can be expensive to install and maintain, and it may not be practical in all situations. In addition, animals may be able to find ways to breach the netting, such as by climbing over it or chewing through it.

- **Traps:** Traps can be used to capture animals that are causing damage in crop fields. These can be live traps that allow the animal to be released elsewhere, or lethal traps that kill the animal. Traps can be humane if they are used properly, but they may not be effective against all types of animals. In addition, traps can be expensive to purchase and maintain, and they may be illegal in some areas.
- **Repellent plants:** Some plants produce chemicals that can repel animals. These plants can be used as a natural deterrent around the perimeter of a crop field. The effectiveness of repellent plants can vary depending on the specific plant and animal species. In addition, these plants may need to be replaced over time, and they may not be effective against all types of animals.
- **Guard animals:** Some farmers use guard animals, such as guard dogs or llamas, to protect their crops from animal damage. These animals can deter animals from entering the field by barking or chasing them away. Guard animals can be effective in deterring certain types of animals, but they may not be practical or cost-effective in all situations. In addition, guard animals may need to be trained and cared for, which can be time-consuming and costly.
- **Crop rotation:** Crop rotation can be used to deter certain animals from entering a crop field. For example, planting a field with a crop that is less attractive to a particular animal species can help to reduce the likelihood of that animal causing damage. Crop rotation can be an effective method of deterring certain animals, but it may not be practical in all situations. In addition, crop rotation can be time-consuming and may require the use of additional resources, such as irrigation or fertilization.

1.1.3. Modern Techniques

Modern techniques have been developed to address the limitations of traditional methods. These techniques and their limitations are given below:

- **Ultrasonic devices:** Ultrasonic devices emit high-frequency sounds that are supposed to repel animals. These devices can be effective in deterring certain animals, but they may not work on all species. They are also expensive and may require frequent battery replacement.
- **Motion-activated sprinklers:** Motion-activated sprinklers are designed to spray water when an animal enters a designated area. This can be effective in deterring animals, but it can also waste water and may not work in all weather conditions.
- **Thermal imaging:** Thermal imaging cameras can detect the presence of animals in crop fields. This can be useful for identifying which animals are causing damage and where they

are entering the fields. However, thermal imaging cameras can be expensive and require trained personnel to operate.

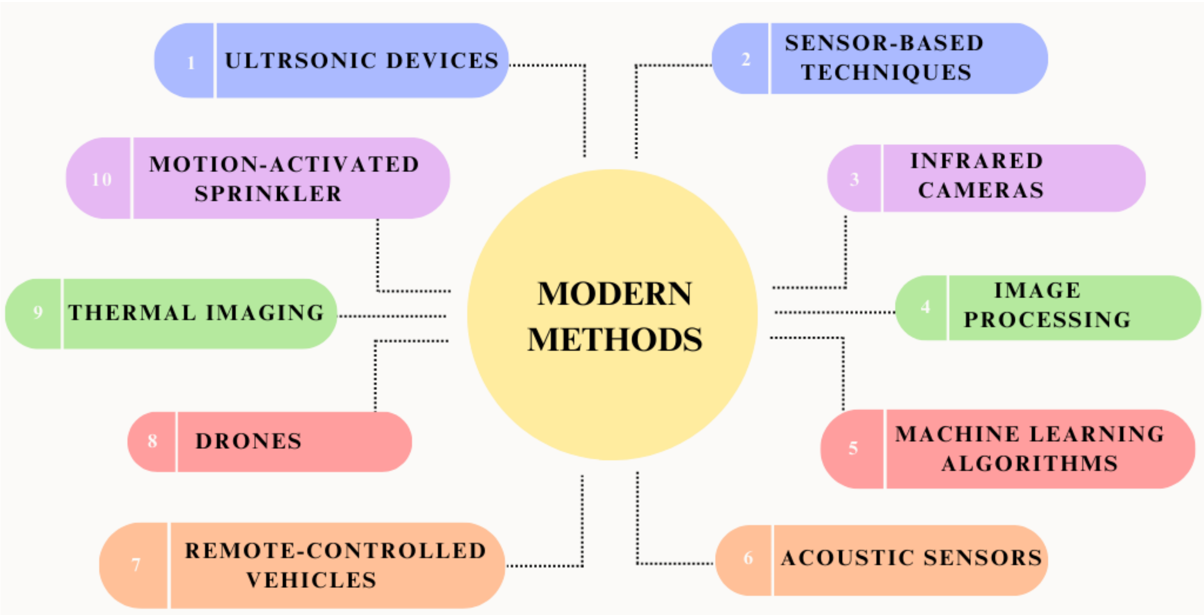

Fig 1.1.3.1. The modern techniques used by farmers to prevent animals from entering crop fields

- **Drones:** Drones equipped with cameras can be used to monitor crop fields and detect the presence of animals. This can be useful for identifying areas of the field that require additional protection. However, drones can be expensive and may require a trained operator.
- **Remote-controlled vehicles:** Remote-controlled vehicles equipped with cameras and sensors can be used to monitor crop fields and detect any animal intrusion. The advantage of this method is that it allows for real-time monitoring and can be effective in detecting animals that are difficult to spot. However, remote-controlled vehicles can be expensive and require skilled operators to use effectively.
- **Sensor-based techniques:** These techniques use sensors to detect the presence of animals in crop fields. The sensors can be placed on the ground or on fences and walls. They can detect the movement and vibration caused by animals and trigger an alarm to scare them away. However, these sensors may not be effective in detecting small animals or animals that move slowly.
- **Infrared cameras:** Infrared cameras can detect the heat signatures of animals and are useful for detecting animals at night or in low light conditions. However, these cameras

may not be effective in detecting animals that blend in with their surroundings or animals that move too quickly.

- **Image processing:** Image processing techniques can be used to analyze images captured by cameras and detect the presence of animals. This technique involves using algorithms to identify the animals in the images and trigger an alarm. However, image processing techniques may not be effective in detecting animals that are partially obscured or hidden by foliage.
- **Machine learning algorithms:** Machine learning algorithms can be used to analyze images and identify animals in crop fields. These algorithms can be trained on a large dataset of images to improve their accuracy. However, these algorithms require a large amount of data and computational resources to train, and may not be effective in detecting new or unknown animal species.
- **Acoustic sensors:** Acoustic sensors can be used to detect the sound of animal movement and trigger an alarm. However, these sensors may be triggered by other sounds, such as wind or rain, and may not be effective in detecting small animals or animals that move slowly.

In summary, animal intrusion into crop fields is a significant challenge facing farmers worldwide. Traditional methods used to address this problem have limitations, and modern techniques have been developed to overcome them. However, these techniques also have some limitations, and there is a need for further research to develop more effective and affordable solutions.

1.2. Machine Learning Approach

As the world becomes increasingly complex, people are faced with new challenges that require innovative solutions. Luckily, advances in machine learning help to tackle these problems in ways were never thought possible. From predicting stock prices to diagnosing diseases, machine learning has revolutionized the way problem-solving is approached. Now, with the power of synthetic image generation and advanced object detection algorithms, farmers can even protect the crops from animal intrusion.

1.2.1. LSGAN

LSGAN, or Least Square Generative Adversarial Network, is a type of generative adversarial network (GAN) that is used in image synthesis tasks. In traditional GANs, the generator is trained to create fake images that are then evaluated by the discriminator to determine their authenticity. However, GANs can be difficult to train and may suffer from problems like mode collapse, where the generator produces a limited range of outputs. To address these issues, LSGAN was introduced, which uses the least squares objective function instead of the traditional binary cross-entropy loss function. This change results in a more stable training process and can generate higher-quality images with greater diversity.

It is an advanced deep learning architecture that can learn the statistical distribution of data and generate new data similar to the training data. LSGAN is a recent development in the field of GANs that uses a least squares loss function to stabilize the training process and generate high-quality synthetic images.

The main idea behind LSGAN is to improve the training stability of GANs by using a least squares loss function. In traditional GANs, the generator network generates images to deceive the discriminator network, which tries to distinguish between real and fake images. However, this approach can lead to instability during training and produce low-quality images. LSGAN overcomes this issue by using a least squares loss function instead of a binary cross-entropy loss function. The generator network is trained to minimize the distance between the generated and real images, while the discriminator network is trained to assign high scores to real images and low scores to generated images. This approach results in a more stable training process and generates high-quality synthetic images.

LSGAN has many applications in various fields, including computer vision, image synthesis, and medical image analysis. It can be used for image editing, style transfer, and image-to-image translation. It can also be used for data augmentation, where it can generate synthetic images to increase the size of the training dataset.

LSGAN is being utilized in the proposed project to generate synthetic images of animals in crop fields, which can be used to improve the training dataset for the object detection model. These synthetic images are combined with the original dataset, allowing for a larger and more diverse dataset, ultimately resulting in a more accurate object detection model. The use of LSGAN facilitates the generation of high-quality synthetic images that are realistic and varied, resulting in an improved ability to detect animals accurately in different lighting and situational conditions.

1.2.2. YOLOv7

You Only Look Once (YOLO) is a state-of-the-art object detection algorithm that is widely used in computer vision applications. YOLOv7 is the latest version of this algorithm, which is known for its high accuracy and real-time performance. It was developed by Joseph Redmon and Ali Farhadi in 2016 and has since undergone several updates and improvements. The YOLO algorithm works by dividing the input image into a grid of cells and predicting bounding boxes and class probabilities for each cell. The algorithm also uses a single neural network to make these predictions, which makes it much faster than traditional object detection algorithms that use multiple networks for different tasks.

YOLOv7 is an improved version of the earlier YOLO models, with several enhancements that make it more accurate and efficient. One of the main advantages of YOLOv7 is its real-time performance, which makes it suitable for use in applications where speed is critical, such as autonomous vehicles, robotics, and surveillance systems. It is also highly accurate, with state-of-the-art performance on several benchmark datasets.

YOLOv7 uses a single CNN that simultaneously predicts the class probabilities and bounding boxes for all the objects in the image in a single forward pass. This makes YOLOv7 very fast and efficient, as it can process images in real-time with high accuracy. The network architecture of YOLOv7 is based on the Darknet neural network framework, which has been optimized for speed and efficiency. The YOLOv7 network is divided into three main parts: the backbone, the neck, and the head. The backbone is a feature extraction network that extracts features from the input image. The neck is a set of convolutional layers that merge features from different scales, while the head is a set of fully connected layers that predict the class probabilities and bounding boxes for each object.

It uses anchor boxes, which are predefined shapes and sizes that are used to predict the bounding boxes for each object. The network predicts the offsets from the anchor boxes to the actual bounding boxes for each object. This allows the network to detect objects of different sizes and aspect ratios. YOLOv7 is trained on large datasets of labelled images using backpropagation and gradient descent. During training, the network adjusts its weights to minimize the difference between its predictions and the ground truth labels. It uses various optimization techniques, such as data augmentation, dropout, and learning rate scheduling, to improve its accuracy and prevent overfitting.

One of the most significant applications of YOLOv7 is in the field of computer vision for object detection and tracking. The algorithm can detect and track objects in real-time video streams, making it useful for applications such as surveillance, autonomous driving, and robotics. It is also used in fields such as sports analysis, where it can be used to track the movement of players and balls in real-time.

In the agricultural sector, YOLOv7 can be used for detecting and tracking animals in crop fields, which is the focus of the project. The algorithm can detect animals such as deer, rabbits, birds, and other wildlife that can cause damage to crops. By detecting the presence of these animals in real-time, farmers can take immediate action to drive them away and prevent crop damage.

It is also used in the healthcare industry for applications such as medical imaging and disease diagnosis. The algorithm can detect and classify various types of medical images such as X-rays, CT scans, and MRI images, and help doctors make accurate diagnoses. In addition, it is used in the field of facial recognition for security and surveillance purposes. Overall, YOLOv7 is a highly versatile algorithm with a wide range of applications in various fields, making it a valuable tool for many industries.

YOLOv7 is utilized for animal detection in the project because it is a state-of-the-art object detection algorithm that has proven to be highly accurate and efficient. YOLOv7 uses a single neural network to predict the bounding boxes and class probabilities for objects in an image, which makes it faster and more accurate than other object detection algorithms that use multi-stage processing. This is important in the project because quick and accurate detection of animals is required to alert farmers and prevent crop damage.

The model is trained on a dataset that includes both real and synthetic images of animals in crop fields. Synthetic images generated by LSGAN are used to increase the size and diversity of the training dataset, which improves the accuracy of the object detection model. The model is fine-tuned to detect specific types of animals that are common in crop fields, such as deer, rabbits, and birds. Overall, YOLOv7 is an ideal algorithm for the project because it is fast, accurate, and efficient, making it suitable for real-time animal detection in crop fields.

1.2.3. Complete Model

The project aims to solve the problem of animal intrusion in crop fields using machine learning and computer vision techniques. LSGAN has been used to generate synthetic images of animals in crop fields, which are combined with real images to create a larger and more diverse training dataset for the object detection model. YOLOv7 has been used as the object detection algorithm because of its high accuracy and efficiency.

The trained model is integrated with a camera, which captures real-time footage of crop fields. The frames captured by the camera are passed to the object detection model, which detects animals and provides their location and type. Based on the animal detected, the system generates a specific sound to repel the animal and prevent crop damage.

Chapter 2

LITERATURE SURVEY

1. **Title:** Ethical Crop Protection.

   Authors: Kekre, O., Paliwal, A., Mehatkar, S., & Hazare, R.

   Journal: *J. Crit. Rev*

   Year: 2019

   The authors described an ethical crop protection system which uses a PIR sensor, accelerometer sensor, thermal camera, and Raspberry Pi. PIR sensor detects object in range 5-6 meters, accelerometer detects the vibrations of the fences when an animal touches them in range of 20 meters and thermal camera measures the temperature of animals. All these readings are then fed to Raspberry Pi with KNN classifier for detecting weather the detected object by the sensors is an animal or not. The experimental results show that this system is having an accuracy of 82.7%. If animal detected the appropriate sound and light combination is used to scare the animals away.

2. **Title:** Wild Animal Intrusion Detection using Deep Learning Techniques.

   Authors: DR. R.S. Sabeenian, N. Deivanai, B. Mythili.

   Journal: International Journal of Engineering and Advanced Technology (IJEAT).

   Year: 2020

   Here machine learning is used to detect animals by using deep neural network concept, a division in computer vision. In their project the farm is monitored through camera on regular basis and with help of machine learning model, it detects entry of animals and play sound to drive them away. Their project works on limited animal dataset with very low accuracy.

3. **Title:** Animal detection using deep learning algorithm.

   Authors: Banupriya, N., Saranya, S., Swaminathan, R., Harikumar, S., & Palanisamy, S.

   Journal: *J. Crit. Rev*

   Year: 2020

   The authors developed an algorithm to detect animals in wild life. CNN algorithm is developed which consist of convolutional layer that extract features from input image, then the 2x2 pooling is done to reduce the number of parameters. Flattening is performed to convert the 2D arrays into a single long continuous linear vector. The last step is the fully connected layer which combines the features into more attributes that predict the output more accurately. The experiment conducted with dataset of elephant and cheetah 100 images each showed a result of 79.25% accuracy for elephant and 86.79% accuracy for cheetah.

4. **Title:** Machine Learning-based Acoustic Repellent System for Protecting Crops Against Wild Animal Attacks.
   Authors: Devsmit Ranparia, Gunjeet Singh, Anmol Rattan, Dr. Harpreet Singh, Dr. Nitin Auluck.
   Journal: 15th (IEEE) International Conference on Industrial and Information Systems (ICIIS)
   Year: 2020
   Here an Acoustic Repellant System has been designed which uses a conventional neural network-based ML model and an IR camera to identify target animals. A Raspberry Pi (Rpi) module has been integrated with a camera and frequency generator to recognize different animals. Their system has a problem of using IR camera which gives positive result for both domestic and non-harmful animals.

5. **Title:** Yolo v4 for Advanced Traffic Sign Recognition with Synthetic Training Data Generated by various GAN.
   Authors: Christine Dewi, Rung-Ching Chen, Yan-Ting Liu, Xiaoyi Jiang, Kristoko Dwi Hartomo.
   Journal: IEEE 15th International Conference on Industrial and Information Systems (ICIIS).
   Year: 2021
   The purpose of their research is to describe how the quality of synthetic pictures created by DCGAN, LSGAN, and WGAN is determined. Their work combines synthetic images and original images to enhance datasets and verify the effectiveness of synthetic datasets. After mixing the real images with synthesized images produces by LSGAN, the recognition performance has been improved with 84.9% on YOLO v3 and 89.33% on YOLO v4.

6. **Title:** Design, Development and Evaluation of an Intelligent Animal Repelling System for Crop Protection Based on Embedded Edge-AI.

Authors: Davide Adami, Mike O. Ojo, Stefano Giordano.

Journal: International Journal of Recent Technology and Engineering (IJRTE)

Year: 2021

The authors proposed an intelligent animal repelling system for crop protection based on embedded edge AI. Here Raspberry Pi, NVDIA jetson nano and intel NGS as edge computing device. PIR sensor is used to detect activity and send message to edge computing device, which in turn activates the camera and executes its DNN software to identify the target. 1000 images of wild boar and deer are expanded to 10,000 images using data augmentation method and used for training. YOLO v3 and tiny YOLO v3 is used as animal detectors and comparison is done. The experimental results showed that YOLO v3 had better accuracy with 82.5% mAP and 64% recall.

7. **Title:** Application of IOT and machine learning in crop protection against animal intrusion.

   Authors: Balakrishna, K., Mohammed, F., Ullas, C. R., Hema, C. M., & Sonakshi, S. K.

   Journal: Global Transitions Proceedings

   Year: 2021

   Authors combined the IoT and ML and compared the performance of two different algorithms R-CNN (Regin based Convolutional Neural Network) and SDD (Single Shot Detector). They have used Raspberry Pi 4, ESP826 Wi-Fi module, Pi camera (8 MP), LED, lead acid battery (12v). A 300 images dataset of horse, zebra, cheetah, elephant, cow with 60 images each was used to train the model. Pi camera captured the images every 5 seconds in both day and night. These images were analyzed by R-CNN and SDD. If animal detected buzzer is played, LED is turned on and message is sent via Wi-Fi module to app through IoT cloud. The experiments showed that the mAP (max Average Precision) of R-CNN approach is 83.42% and that of SDD is 89.32%. The SDD outstands R-CNN and authors described that R-CNN take 40-50 seconds more for detection than SDD.

8. **Title:** Identification of animals and recognition of their actions in wildlife videos using deep learning techniques.

   Authors: Schindler, F., & Steinhage, V.

   Journal: *Ecological Informatics*

   Year: 2021

   The authors proposed methods for animal detection and animal recognition. In animal detection, an infrared camera (8 frames per second) is used to take videos at night. Dataset used here consists of 10 seconds video clips of 8 fps with a total of 528 clips (deer – 477, boar – 20, fox – 8, hare – 23). Authors compared the results of Mask R-CNN and FGFA (Flow Guided Feature Aggregation) with given dataset on 4 GHz CPU, i7 Intel Processor, 16 GB RAM, Python 3 and PyTorch. Experiments showed that Mask R-CNN approach had best results than FGFA with an average precision of 63.8%. For animal recognition authors compared three different ResNet-18 variations (3D ResNet (R3D), ResNet with mixed convolutions (MC ResNet), and R(2+1)D ResNet) and SlowFast architecture on same dataset. Here the data was divided into 3 categories, eating – 257, moving – 39, watching – 50. The experiments showed that the (2+1)D ResNet reached highest accuracy of 94%, with other two ResNet of 93% and 92%. SlowFast architecture had least accuracy with 66%.

9. **Title:** SAR Target Recognition Based on Transfer Learning and Data Augmentation with LSGANs.

   Authors: Yu Ma, Yan Liang, Wan Ying Zhang, Shi Yan.

   Journal: National Science Foundation of China under grant 61873205

   Year: 2022

   Their paper presents a method combining transfer learning and data augmentation based on LSGAN. It used to effectively solve the problem of lacking training samples in SAR target recognition algorithm. SAR is kind of high-resolution imaging radar which has ability to operate in all-weather condition in many fields such as military, agriculture and fisheries.

Chapter 3

SYSTEM ANALYSIS

3.1. Existing System

Animal attacks are typical occurrence in India these days, the crops are destroyed by them due to lack of any detection system. Traditional techniques used in crop fields for animal protection include fencing, scarecrows, noise-making devices, chemical repellents, predator decoys, biological control, netting, traps, repellent plants, guard animals, and crop rotation.

Fencing is a common method that involves physical or sensory barriers to prevent animals from entering the crop fields. Scarecrows are effective against birds but may not deter larger animals like deer. Noise-making devices emit sounds or lights to deter animals. Chemical repellents create an unpleasant or toxic environment for animals. Predator decoys mimic natural predators to scare animals away. Biological control involves releasing predators or introducing diseases or parasites to control animal populations. Netting physically excludes. Traps capture animals and can be humane or lethal. Repellent plants produce chemicals that repel animals. Guard animals, such as dogs or llamas, deter animals through barking or chasing. Crop rotation involves planting less attractive crops to deter specific animal species.

These traditional techniques have their limitations and may not provide a comprehensive solution to the problem of animal intrusion in crop fields. Therefore, our project aims to leverage advanced technologies like machine learning and computer vision to develop a more effective and efficient solution.

Modern techniques used by farmers to prevent animals from entering crop fields include ultrasonic devices, motion-activated sprinklers, thermal imaging, drones, remote-controlled vehicles, sensor-based techniques, infrared cameras, image processing, machine learning algorithms, and acoustic sensors.

Ultrasonic devices emit high-frequency sounds to repel animals. Motion-activated sprinklers spray water when an animal enters a designated area, deterring them. Thermal imaging cameras detect animal presence, aiding in identifying areas of damage and entry points. Drones equipped with cameras monitor crop fields and detect animal presence, particularly useful for targeting specific areas. Remote-controlled vehicles with cameras and sensors provide real-time monitoring, effectively detecting elusive animals. Sensor-based techniques use ground or fence sensors to detect animal movement and trigger alarms. Infrared cameras detect animal heat signatures, particularly useful in low light or night conditions.

Image processing techniques analyse camera-captured images to identify animals and trigger alarms. Machine learning algorithms analyse images and improve accuracy through training on large datasets. Acoustic sensors detect animal sounds and trigger alarms.

These modern techniques offer innovative approaches to animal intrusion prevention in crop fields. However, each technique has its limitations, and careful consideration is needed to select the most suitable method based on the specific requirements and challenges of the farming environment.

Disadvantages of the existing system:

- Fencing, while commonly used, can be expensive to install and maintain. Animals can find ways to breach fences, such as by digging under or climbing over them, making them less effective. Electric fences can also pose a fire risk when combined with vegetation.
- Scarecrows are limited in their effectiveness against larger animals and require frequent maintenance.
- Noise-making devices may become less effective over time as animals become accustomed to the noise. They can also be disruptive to humans living or working near the crop fields.
- Chemical repellents, while used by some farmers, can be harmful to the environment and non-target species. Animals may become accustomed to the repellents, reducing their effectiveness.
- Predator decoys may not deter all animals, especially if they are unfamiliar with the simulated species. These decoys may also require regular replacement or repair, adding to the cost.
- Biological control methods, although employed by some farmers, can have unintended consequences on the ecosystem and may not be effective against all animal species.
- Netting, while effective for small fruit trees or berry bushes, can be expensive and may not prevent animals from breaching it.
- Traps, while humane if used properly, may not be effective against all animals and can be expensive to purchase and maintain. Their use may also be restricted or prohibited in certain areas.
- Repellent plants, while providing a natural deterrent, may need to be replaced over time and may not be effective against all animals.

- Guard animals, such as dogs or llamas, can be effective in deterring certain animals but may not be practical or cost-effective in all situations. They also require training and care, which can be time-consuming and costly.
- Crop rotation, although effective in deterring certain animals, can be time-consuming and may require additional resources like irrigation or fertilization.
- Ultrasonic devices may not work on all animal species and can be expensive to purchase and maintain.
- Motion-activated sprinklers can waste water and may not function optimally in all weather conditions.
- Thermal imaging cameras can be costly and require trained personnel to operate.
- Drones and remote-controlled vehicles, while providing real-time monitoring, can be expensive and require skilled operators.
- Sensor-based techniques may struggle to detect small or slow-moving animals. Infrared cameras may not detect well-camouflaged or fast-moving animals.
- Image processing techniques may face challenges in identifying partially obscured or hidden animals.
- Machine learning algorithms require significant data and computational resources to train and may not be effective in detecting new or unknown animal species.
- Acoustic sensors can be triggered by other sounds and may not effectively detect small or slow-moving animals.

3.2. Proposed System

The proposed solution for identifying or locating the wild animals that trespass the crop fields is using a Deep Learning approach that involves reading data from live-video from a camera installed in the farm. The images from training dataset are improved for better image quality and then synthetic images of these images are generated which is added with original dataset for better performance. The live data is then fed to the animal detector for detection of animals and set an alarm to drive them away.

LSGAN, or Least Square Generative Adversarial Network, is a type of generative adversarial network (GAN) that is used in image synthesis tasks. LSGAN is being utilized in the proposed project to generate synthetic images of animals in crop fields, which can be used to improve the training dataset for the object detection model. These synthetic images are combined with the original dataset, allowing for a larger and more diverse dataset, ultimately resulting in a more accurate object detection model. The use of LSGAN facilitates the generation of high-quality synthetic images that are realistic and varied, resulting in an improved ability to detect animals accurately in different lighting and situational conditions.

YOLOv7 is an improved version of the earlier YOLO models, with several enhancements that make it more accurate and efficient. YOLOv7 is utilized for animal detection in the project because it is a state-of-the-art object detection algorithm that has proven to be highly accurate and efficient. YOLOv7 uses a single neural network to predict the bounding boxes and class probabilities for objects in an image, which makes it faster and more accurate than other object detection algorithms that use multi-stage processing. This is important in the project because quick and accurate detection of animals is required to alert farmers and prevent crop damage.

The model is trained on a dataset that includes both real and synthetic images of animals in crop fields. Synthetic images generated by LSGAN are used to increase the size and diversity of the training dataset, which improves the accuracy of the object detection model. The model is fine-tuned to detect specific types of animals that are common in crop fields, such as deer, rabbits, and birds. Overall, YOLOv7 is an ideal algorithm for the project because it is fast, accurate, and efficient, making it suitable for real-time animal detection in crop fields.

The project aims to solve the problem of animal intrusion in crop fields using machine learning and computer vision techniques. LSGAN has been used to generate synthetic images of animals in crop fields, which are combined with real images to create a larger and more diverse training dataset for the object detection model. YOLOv7 has been used as the object detection algorithm because of its high accuracy and efficiency.

The trained model is integrated with a camera, which captures real-time footage of crop fields. The frames captured by the camera are passed to the object detection model, which detects animals and provides their location and type. Based on the animal detected, the system generates a specific sound to repel the animal and prevent crop damage.

Advantages of proposed system:

- User Friendly: The proposed system is very user friendly. The graphical user interface is provided in the proposed system, which provides user to deal with the system very comfortably.
- Real-time monitoring: - The system employs cameras to monitor the entire farm in real-time. This continuous surveillance enables farmers to have a comprehensive view of their

fields and immediate awareness of any animal intrusion. Real-time monitoring ensures quick respond to potential threats and take necessary actions in a timely manner.

- Cost-effective solution: - By utilizing cameras and alarm systems, it eliminates the need for manual patrolling or hiring additional personnel for constant monitoring. This automation reduces labour costs and provides round-the-clock surveillance, making it an efficient and economical option for farmers.
- Improved accuracy and efficiency: - The use of image enhancement techniques based on LSGAN and object detection algorithm YOLOv7 enhances the accuracy and efficiency of animal detection. This improved accuracy ensures reliable animal detection, reducing false alarms and minimizing disruption for farmers.
- Animal-friendly approach: - The system aims to ensure the safety of both humans and animals. Instead of causing harm to the animals, the system plays appropriate sounds to drive them away from the fields. It allows for the coexistence of wildlife and agriculture, contributing to sustainable farming practices.
- Time-saving solution: - By automating the process of animal detection and alerting, the proposed system saves farmer’s valuable time. Farmers no longer need to physically patrol their fields or rely solely on manual observation. The system provides instant notifications and alerts, enabling farmers to focus their time and efforts on other essential tasks, such as crop management and productivity enhancement.
- Support for farmers: - The project aims to support farmers by providing them with a reliable tool to protect their crops from animal damage. The proposed system offers an accessible and user-friendly solution that can be easily implemented on farms of varying sizes. By minimizing crop losses due to animal attacks, the system contributes to the overall well-being and profitability of farmers.

Chapter 4

REQUIREMENT SPECIFICATIONS

4.1. SOFTWARE REQUIREMENTS

- Any 64-bit Operating System with python installed.
- Tool: Python, OpenCV, Flask, SQLite
- Technology and Languages: Python, JavaScript
- Any Browser (preferably chrome)

4.2. HARDWARE REQUIREMENTS

- Intel i3 or Ryzen 3 and above
- Speed 1.1 GHz
- RAM 4GB and Above
- 20 GB Free Hard Disk Space

Chapter 5

SYSTEM DESIGN

5.1. Architecture Diagram

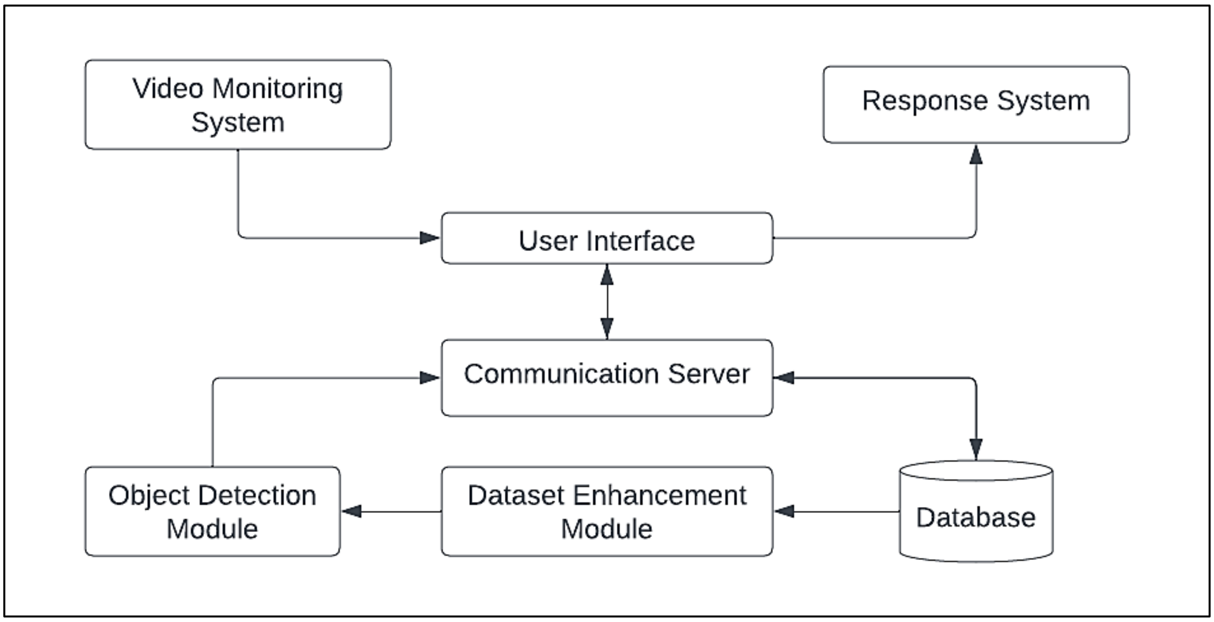

Fig 5.1.1. Architecture Diagram

The architecture diagram is a visual representation of the system components and their interactions. The video monitoring system captures the footage of the crop fields and sends it to the object detection module for animal detection. The dataset enhancement module is responsible for generating more realistic and varied training pictures using LSGAN. The object detection module processes the images and detects animals using YOLOv7 algorithm.

The user interface is designed for the farmer to interact with the system and monitor the farm in real-time. The response system generates appropriate sounds to drive away the detected animals. The communication server handles the communication between different system components. The database stores the information about the detected animals, and the system events. Together, these components form a reliable and efficient system for detecting animals in crop fields.

5.2. Data Flow Diagram

A data flow diagram is a visual representation of how data flows through a system.

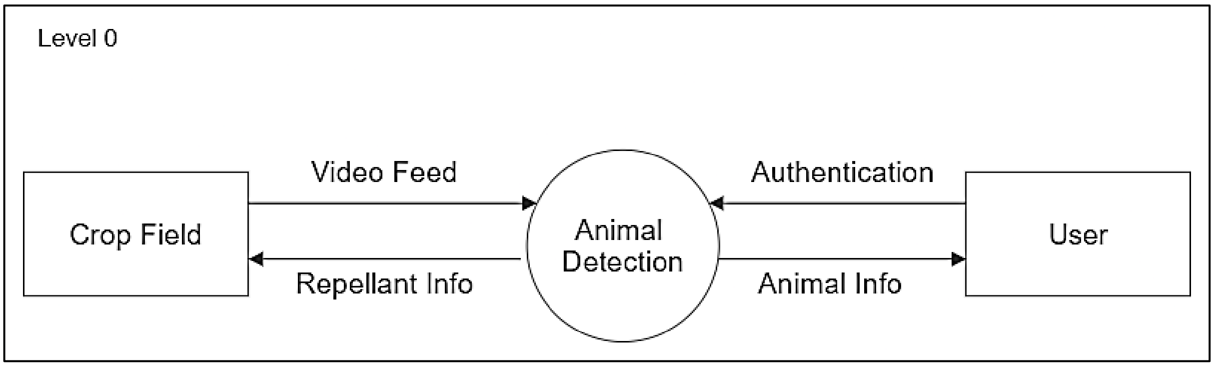

Fig 5.2.1. Level 0 of Data Flow Diagram

Level 0 of the data flow diagram represents the main components of the system which includes Crop Field, Animal Detection and Response System. Crop Field sends the video feed to the Animal Detection module which analyses it to detect animals. If any animal is detected, the information is passed on to the User and Crop Field which receives this repellent information and takes necessary action to safeguard the crops.

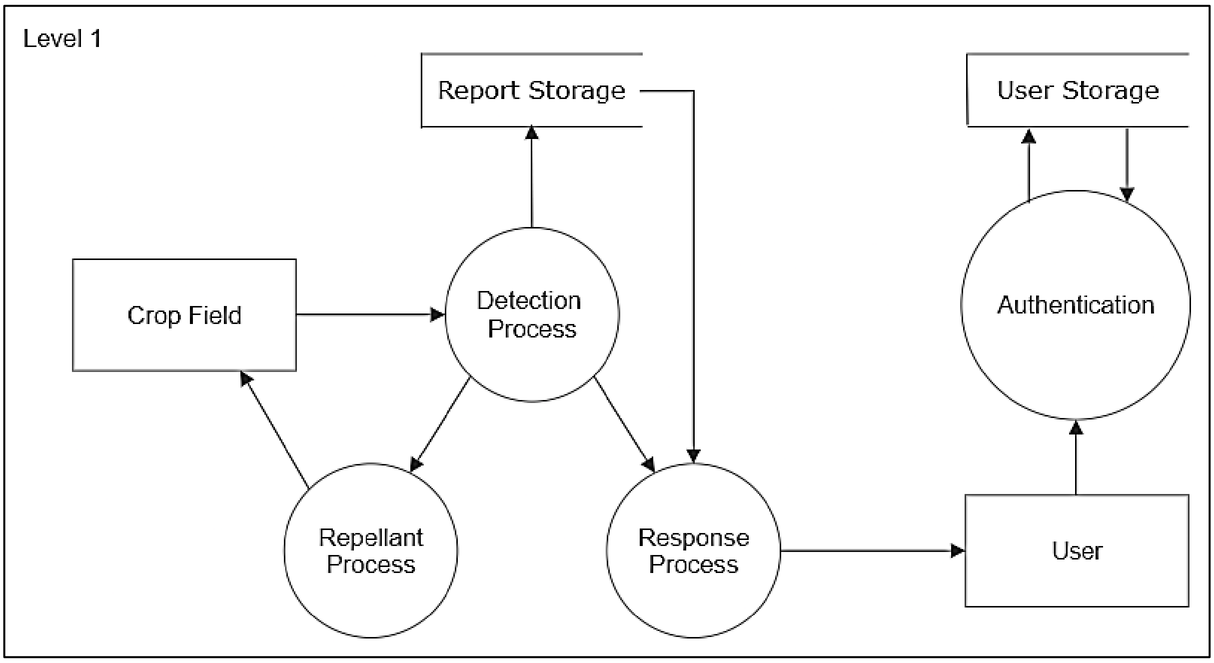

Fig 5.2.2. Level 1 of Data Flow Diagram

Level 1 of the data flow diagram shows the detailed flow of data between various components of the system. The Crop Field captures the video feed and sends it to the Detection Process for analysing the images. Once the detection process detects any animal, it sends the information to the Repellent Process, which starts the appropriate alarm sound to drive the animals away. The Response Process sends the message to user. The User accesses the system through the Authentication Process, and the User Storage stores the user data. The Report Storage stores the reports generated by the system.

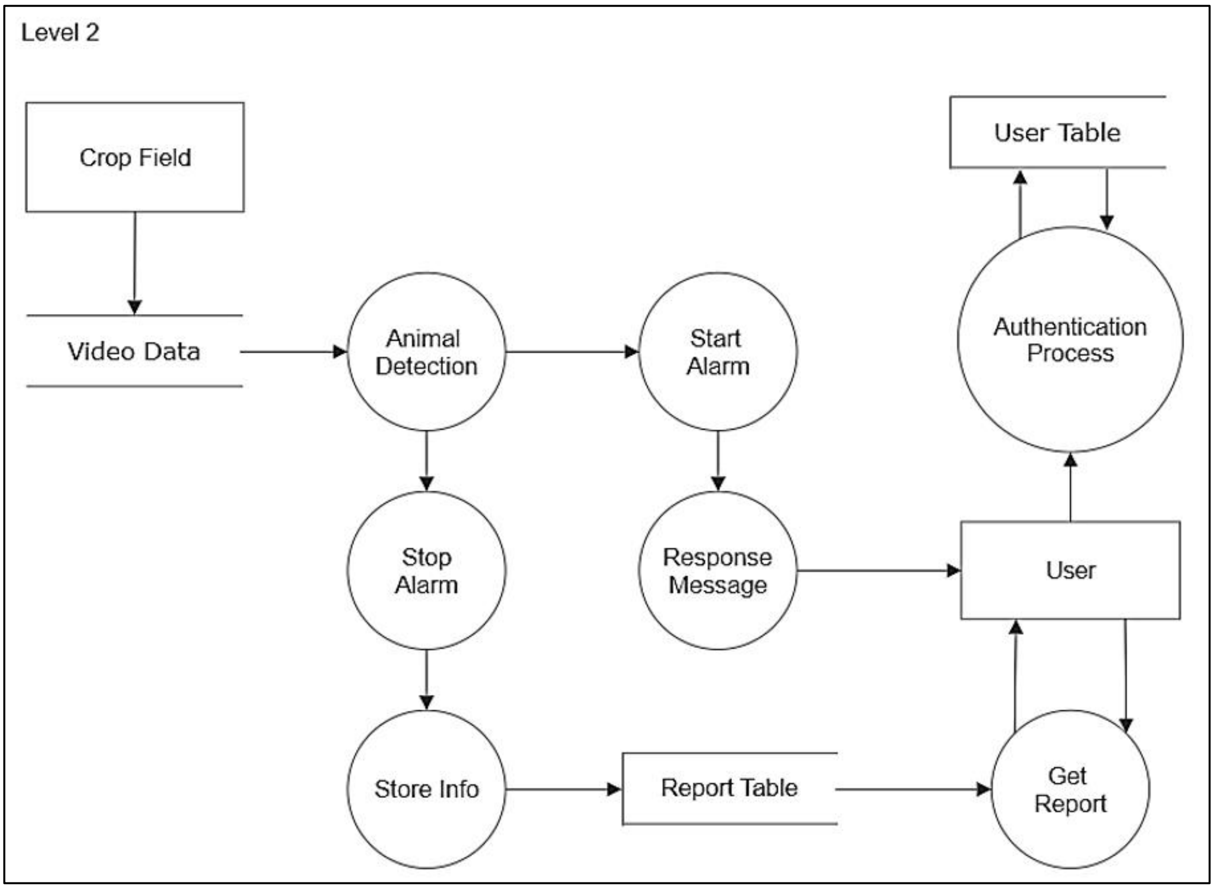

Fig 5.2.3. Level 2 of Data Flow Diagram

Level 2 of the data flow diagram shows a more detailed process flow of the Response Process and the Repellent Process. These two processes are divided into three sub-processes: Start Alarm, Stop Alarm and Send Message. The Start Alarm process triggers the alarm when an animal is detected, and the Stop Alarm process stops the alarm once the animal is driven away. The Send Message process sends a message to the user about the animal detection and the action taken by the system.

5.3. Use Case Diagram

The Use Case diagram depicts the interaction between two agents - the User and the Detection module. The User can perform actions such as signup, login, start model, view response, download report, and logout. On the other hand, the Detection module can capture video, detect animal, sound alarm, send response, and generate reports. This diagram gives a clear understanding of the different functionalities of the system and how each agent interacts with the system. It helps in identifying the user's goals and the system's responses to those goals.

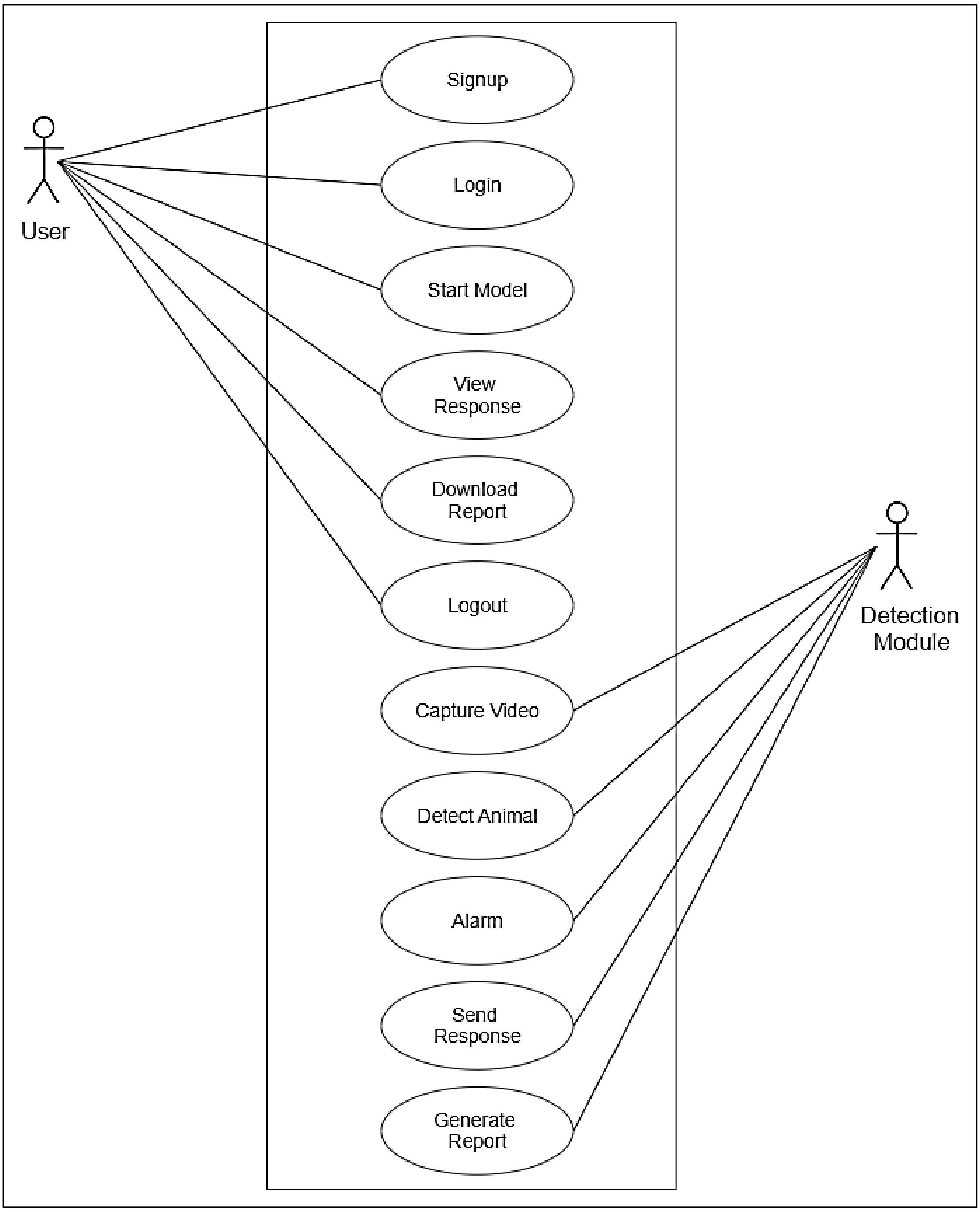

Fig 5.3.1. Use Case Diagram

5.4. Sequence Diagram

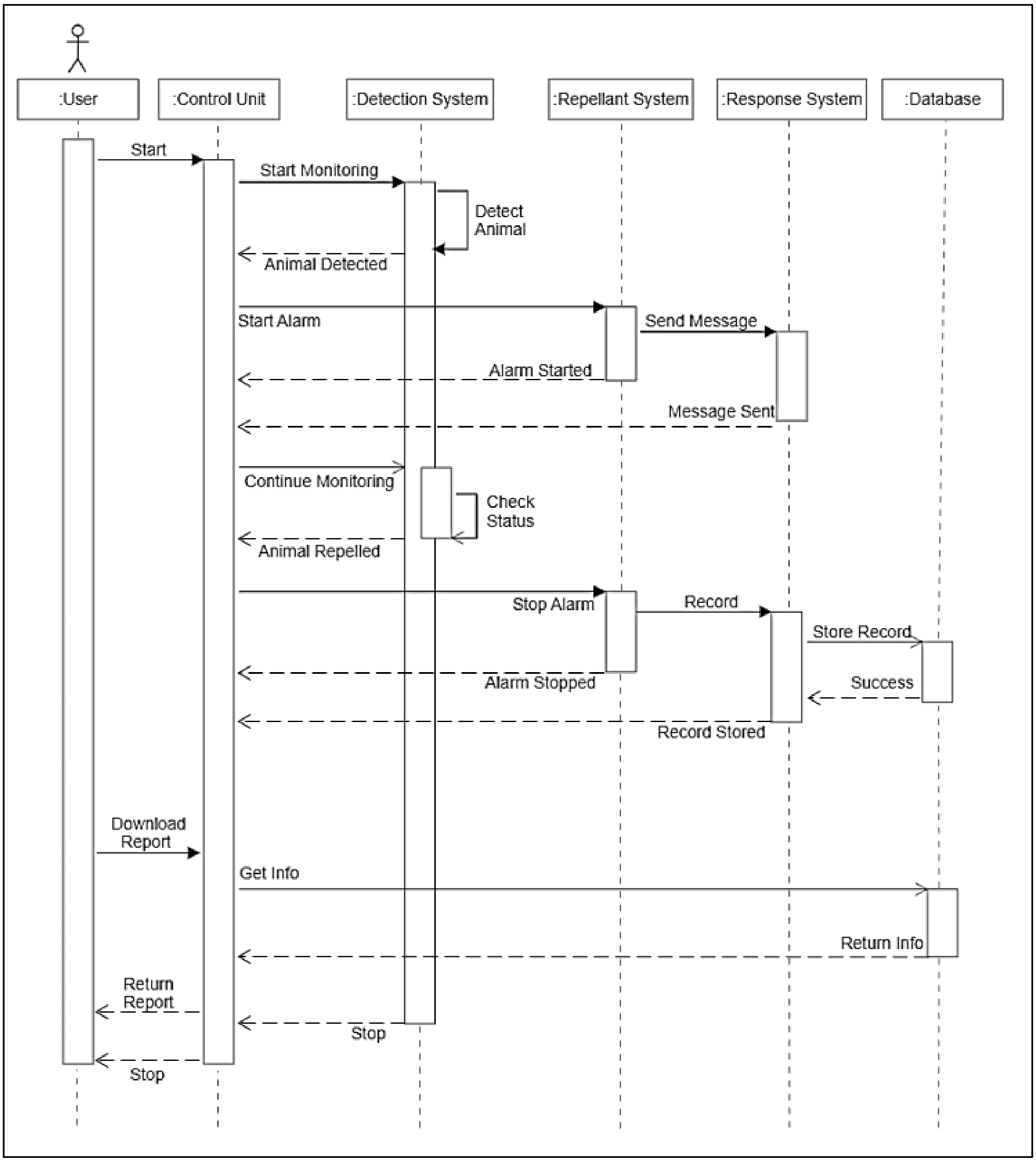

Fig 5.4.1. Sequence Diagram

The Sequence diagram illustrates the interaction between various objects in the system. It shows how the user interacts with the control unit, which in turn interacts with the detection system until an animal is detected. Once an animal is detected, the control unit receives a response and starts the alarm by interacting with the repellent and response system. If the animal is repelled, the alarm is stopped. The user can interact with the control unit and the control unit interacts with database for downloading the report. The sequence diagram provides a visual representation of how the objects in the system interact with each other, making it easy to understand the flow of the system. It helps in identifying the potential issues and helps to optimize the system for better performance.

5.5. Activity Diagram

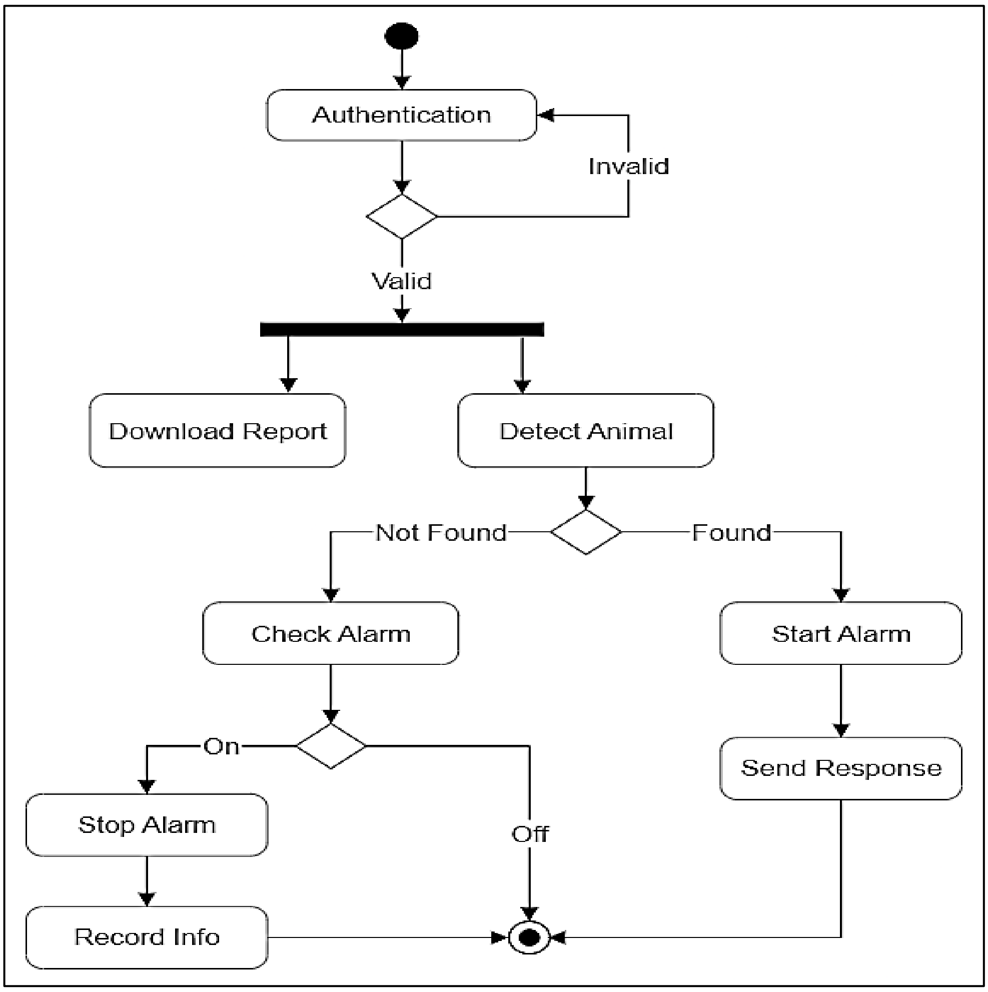

Fig 5.5.1. Activity Diagram

The activity diagram illustrates the sequence of activities involved in the video monitoring system. The diagram starts with the authentication process, where the user logs in and verifies the credentials. In case of invalid credentials, the system loops back to the authentication step until the user enters valid credentials. Once the user is authenticated, the system forks into two paths - download report or detect animal. In case the user wants to download the report, the system retrieves the report from the database and sends it to the user. If the user wants to detect animals, the system starts the animal detection process. If an animal is detected, the system starts the alarm and sends a response to the user. In case no animal is detected, the system checks if the alarm is on and turns it off if required. The activity diagram provides a clear understanding of the flow of activities involved in the system and helps in identifying any potential issues or bottlenecks in the process.

5.6. Process Flow Diagram

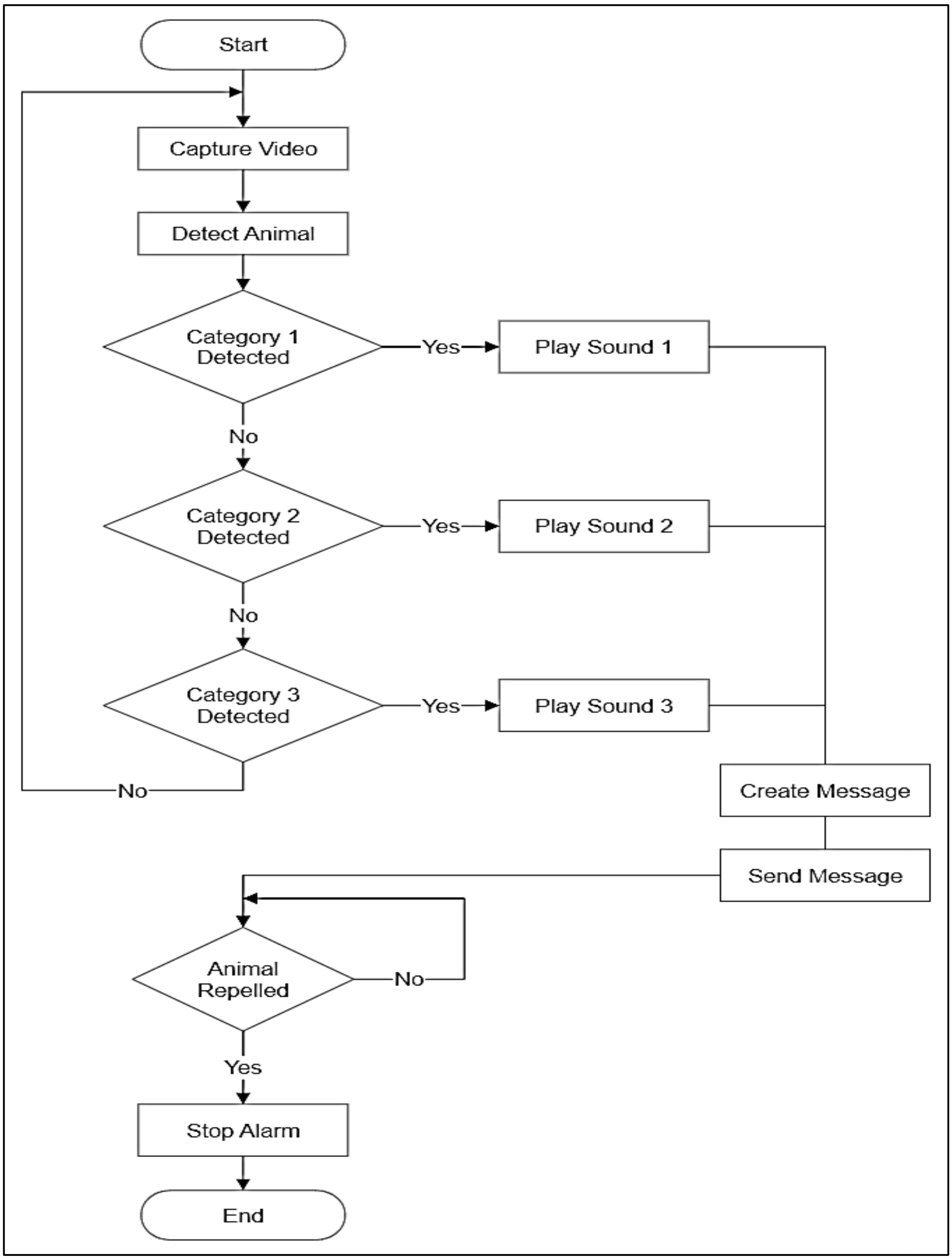

Fig 5.6.1. Process Flow Diagram

The process flow diagram depicts the sequence of steps involved in the functioning of our Animal Detection system. Starting with capturing video, the system detects any animals in the field using object detection. If an animal is detected, the system triggers an alarm and sends a message to the user. If the animal is repelled, the alarm is stopped else if the animal is still present, the process loops back until the animal is repelled and the alarm stops. This process ensures timely detection and response to any animal intrusion in the crop field, thereby safeguarding the crops and reducing the losses incurred due to animal damage.

Chapter 6

SYSTEM STUDY

6.1. Feasibility Study

A feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present objectively and rationally in the natural environment, the resources required to carry through and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. Generally, feasibility studies precede technical development and project implementation. The acronym TELOS refers to the five areas of feasibility - Technical, Economic, Legal, Operational and Scheduling.

Why we need Feasibility Study?

- When we have new project with new concept.
- When we are not confirming about the resources and time.
- To suggest possible alternative solution.
- To provide management with enough information to know.
- Whether the project can be done.
- Whether the final product will benefit its intended users.

When we do not need Feasibility Study?

- We know it is feasible. An existing business is already doing it
- Why to do another feasibility study when there is an existing one which was done just a few months ago?

6.2. Types of Feasibility Study

6.2.1. Economic Feasibility

Economic feasibility analysis is the most used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study. Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.

6.2.2. Market Feasibility

Market Feasibility Study determines the depth and condition of a particular market and its ability to support a particular development. The main objective of a market feasibility study is to understand the market to determine if enough demand exists to make the venture successful. It provides a more in-depth and thorough analysis than any other type of market research.

6.2.3. Technical Feasibility

The technical possibility of developing a product per client expectation is called technical feasibility. It is a study carried out manually or using tools to understand the likelihood of completing a work based on the availability of materials, people, infrastructure.

Whether the work is technically feasible? This means if it can be done from a technical point of view. There may be financial feasibility that will relate to the cash flow or funding allocated to complete the work. Likewise, operational feasibility deals with the larger picture including cost, people, and facilities available to complete a project.

On the other hand, technical feasibility considers only the technical chances of completing a given work. Say, for instance, there are enough funding, great technical equipment, sufficient time, ample human resources available. But, if the resources are not technically proficient, then the whole purpose gets defeated. Yes, people should know how to use technology in the development process to greater levels to ship the product to the customer.

6.2.4. Legal Feasibility

Legal feasibility study is to know if the proposed project confirms the legal and ethical requirement. It is important that the project or business is following the requirements needed to start a business or a project including business licenses, certificates, copyrights, business insurance, tax number, health and safety measures, and many more. It determines whether the proposed system conflicts with legal requirements, e.g., a data processing system must comply with the local data protection regulations and if the proposed venture is acceptable in accordance to the laws of the land.

6.2.5. Schedule Feasibility

It is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. If a project has a high probability to be completed on-time, then its schedule feasibility is appraised as high. In many cases a project will be unsuccessful if it takes longer than it was estimated: some external environmental conditions may change, hence a project can lose its benefits, expediency, and profitability.

6.2.6. Operational Feasibility

Operational feasibility is the ability to utilize, support and perform the necessary tasks of a system or program. It includes everyone who creates, operates, or uses the system. To be operationally feasible, the system must fulfil a need required by the business. Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase.

Chapter 7

SOFTWARE ENVIRONMENT

7.1. Front End

7.1.1. HTML

HTML (HyperText Markup Language) is the backbone of the web. It provides the structure and content of a web page, and it works together with other technologies such as CSS and JavaScript to make web pages interactive and visually appealing. HTML uses tags to define different parts of a web page, such as headings, paragraphs, images, links, and forms. It is a fundamental technology for web development and is widely used in creating websites, web applications, and mobile apps.

7.1.2. CSS

CSS (Cascading Style Sheets) is a styling language used for describing the presentation of a document written in HTML or XML. It allows developers to separate the presentation of a web page from its content, which makes it easier to manage and modify the visual appearance of a website. With CSS, developers can define the layout, colours, fonts, and other visual elements of a web page. CSS can be used in conjunction with HTML and JavaScript to create dynamic and responsive web pages.

7.1.3. JavaScript

JavaScript is a high-level, interpreted programming language that is used to make web pages interactive and dynamic. It was originally designed to be used in web browsers, but today it can be used on both client and server sides with the help of various frameworks and libraries.

Some of the key features of JavaScript include its dynamic typing, object-oriented programming support, and functional programming capabilities. It is also easy to learn and has a vast community of developers who contribute to various open-source projects. JavaScript allows for the creation of rich and interactive user interfaces, making web pages more engaging and responsive. It can be used to create animations, validate forms, and interact with server-side APIs to update the page content dynamically without refreshing the entire page.

In our project, we have used JavaScript to perform various tasks such as starting and stopping the web camera, sending frames to the backend for processing, receiving processed images via fetch, filtering the report table, downloading the report, and validating the authentication. These features have made our project more dynamic and user-friendly.

7.1.4. Bootstrap

Bootstrap is a popular front-end web development framework that provides a set of pre-built HTML, CSS, and JavaScript components and tools for creating responsive and mobile-first websites. It is designed to simplify web development by providing a consistent and reusable set of components that can be easily customized and extended. Bootstrap includes a grid system for layout, typography, forms, buttons, and navigation, as well as JavaScript plugins for sliders, modals, carousels, and other interactive elements. Bootstrap is widely used by web developers and designers for creating modern and responsive websites that work across different devices and screen sizes.

7.2. Back End

7.2.1. Python

Python is a high-level, general-purpose programming language that is used in a wide range of applications, including web development, data science, artificial intelligence, and machine learning. It is known for its simplicity, readability, and ease of use, making it an excellent language for beginners and experienced developers alike. Python is used in web development for building web applications, web frameworks, and web scraping tools. It is also widely used in data science for analysing and visualizing data, building machine learning models, and automating data workflows.

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

7.3. Packages Used

7.3.1. NumPy

NumPy is a popular library for numerical computing in Python. It is widely used for tasks such as scientific computing, data analysis, and machine learning. In this project, NumPy was used for image processing and data manipulation tasks. NumPy provides a fast and efficient way to work with arrays and matrices in Python, making it ideal for tasks such as image processing. In the animal detection system, NumPy was used to process the image frames captured by the webcam and convert them to the format required by the YOLOv7 object detection model. NumPy was also used to manipulate and process the output of the YOLOv7 model, such as resizing and converting the detected object coordinates to the appropriate format.

7.3.2. Flask

Flask is a popular web development framework for Python that was used in this project to create a web application for the animal detection system. Flask is known for its simplicity, flexibility, and ease of use. It provides a wide range of functionalities and extensions that allow developers to build web applications efficiently. In this project, Flask was used to create routes, render templates, and process HTTP requests. With Flask, it was easy to create a web interface for the animal detection system and deploy it to a web server. Flask enables you to create web applications very easily, hence enabling you to focus your energy more on other important parts of a ML lifecycle like EDA, feature engineering, etc.

7.3.3. Cv2

The cv2 module in Python is a computer vision library that is used for image and video processing. It is built on the OpenCV library, which is a popular open-source computer vision and machine learning software library. The cv2 module provides a variety of functions and algorithms for performing image processing tasks such as image manipulation, feature detection, object recognition, and tracking. It supports a wide range of image and video file formats, and can be used for both real-time and offline processing. The cv2 module is widely used in the field of computer vision, and is particularly useful for applications such as robotics, surveillance, and autonomous vehicles. In our project, we used cv2 to capture frames from the webcam for object detection using the YOLOv7 model, and to display the processed frames in real-time on the web page using Flask.

7.3.4. Datetime

Date and time objects may be categorized as “aware” or “naive” depending on whether they include time zone information. The datetime module in Python is a powerful tool used for working with dates and times. It provides a range of classes to work with dates, times, and time intervals, including the date, time, datetime, timedelta, and tzinfo classes. With the datetime module, it is easy to perform common operations such as date arithmetic, formatting, and parsing. Additionally, it can handle various time zones and daylight-saving time (DST) adjustments. The datetime module is very useful in many applications, including data analysis, financial analysis, and scientific computing. In our project, we have used the datetime module to timestamp the reports and to set the date and time of the animal detections.

7.3.5. Torch

Torch is an open-source machine learning library based on the Lua programming language. It is primarily used for deep learning and provides a flexible platform for building and training neural networks. Torch offers a wide range of tools and modules for optimization, regularization, parallel processing, and data loading. One of the key features of Torch is its ease of use and speed, which makes it a popular choice among researchers and developers. Torch also has a large community of contributors and users who actively develop and share new tools and models, making it a vibrant and evolving ecosystem for deep learning. It provides a high-level interface for creating and manipulating tensors, which are multidimensional arrays used to store and manipulate data in deep learning models. It also provides a suite of tools for building and training neural networks, including optimization algorithms, loss functions, and data loaders.

Chapter 8

IMPLEMENTATION

8.1. System Module

1. User Authentication
2. Animal Detection
3. Report Generation

8.2. Sample Code

Detect.html

```
{% extends 'base.html' %} {% block navcontent %}
<a class="nav-link me-5 mx-2" href="/home">Home</a>
{% endblock navcontent %} {% block content %}
<section style="height: 90vh">
  <div class="container-fluid" style="height: 100%">
    <div class="row" style="height: 100%">
      <div class="col-md-2 d-flex justify-content-center align-items-center">
        <div class="text-center ms-4">
          <h1 class="lh-lg">Animal Detection</h1>
        </div>
      </div>
      <div class="col-md-8 d-flex justify-content-center align-items-center">
        <div class="container">
          <div class="card mx-3">
            <div class="card-body bg-secondary position-relative text-center">
              <video autoplay id="videoElement" style="display: none"></video>
              <canvas id="canvas" style="display: none"></canvas>
              <img id="photo" />
              <div
                class="pressStart position-absolute top-50 start-50 translate-middle text-light"
              >
                Press Start to Start Detecting Animal
              </div>
```

```
        <div
          id="elem"
          style="display: none"
          width="400"
          height="400"
        ></div>
      </div>
    </div>
  </div>
</div>
<div
  class="col-md-2 d-flex justify-content-center align-items-center my-4"
>
  <div class="d-grid gap-5 col-10 mx-auto me-md-5">
    <button
      type="button"
      class="btn btn-outline-dark btn-lg rounded-pill"
      id="startVid"
    >
      START
    </button>
    <button
      type="button"
      class="btn btn-outline-dark btn-lg rounded-pill"
      id="stopVid"
    >
      STOP
    </button>
    <a
      href="/home"
      class="btn btn-outline-dark btn-lg rounded-pill"
      role="button"
    >
      HOME
```

```
          </a>
        </div>
      </div>
    </div>
  </div>
</section>
{% endblock content %}
```

Script.js

```
let audio1 = new Audio("./static/assets/audio/monkey_repel.mp3");
let audio2 = new Audio("./static/assets/audio/cat_dog_repel.mp3");
let audio3 = new Audio("./static/assets/audio/elephant_repel.mp3");
let audio4 = new Audio("./static/assets/audio/lion_leopard_repel.mp3");
let audio5 = new Audio("./static/assets/audio/sheep_cattle_repel.mp3");
let audio = audio1;

function loopAudio() {
    this.currentTime = 0;
    this.play();
}

let noAnimal = -1
async function checkAnimal(animalList) {
    if (animalList.length == 0 || animalList[0] == "person") {
        if (noAnimal == -1 || ++noAnimal > 3) {
            if (!audio.paused) {
                noAnimal = -1;
                audio.removeEventListener("ended", loopAudio);
                audio.pause();
            }
            return;
        }
    }
```

```
    if (audio.paused) {
        if (animalList[0] == 'cat' || animalList[0] == 'dog') {
            audio = audio2;
        } else if (animalList[0] == 'elephant' || animalList[0] == 'bear') {
            audio = audio3;
        } else if (animalList[0] == 'horse' || animalList[0] == 'zebra') {
            audio = audio4;
        } else if (animalList[0] == 'sheep' || animalList[0] == 'cow') {
            audio = audio5;
        } else {
            audio = audio1;
        }
        noAnimal = 0;
        audio.addEventListener('ended', loopAudio);
        audio.currentTime = 0;
        audio.play();
    }
}

let startBtn = document.getElementById('startVid');
let stopBtn = document.getElementById('stopVid');
let canvas = document.getElementById("canvas");
let context = null;
const video = document.querySelector("#videoElement");
let elem = document.getElementById("elem");
let timeout = null;
let pressStart = document.getElementsByClassName("pressStart")[0];

if (startBtn) {
    startBtn.addEventListener('click', startVideo);
    stopBtn.addEventListener('click', stopVideo);
    video.addEventListener("playing", videoStarted);
}
```

```
let videoPlaying = false;
function startVideo(e) {
    context = canvas.getContext("2d");
    video.width = elem.getAttribute("width");
    video.height = elem.getAttribute("height");
    if (navigator.mediaDevices.getUserMedia) {
        navigator.mediaDevices.getUserMedia({ video: true })
            .then(function (stream) {
                video.srcObject = stream;
            })
            .catch(function (err0r) {
                alert("Something went wrong! Cannot open Webcam");
            });
    }
    pressStart.classList.add("z-n1");
    videoPlaying = true;
}

function stopVideo(e) {
    videoPlaying = false;
    let stream = video.srcObject;
    if (stream != null) {

        let tracks = stream.getTracks();

        for (let i = 0; i < tracks.length; i++) {
            let track = tracks[i];
            track.stop();
        }

    }
    video.srcObject = null;
```

```
    clearTimeout(timeout);

    context.fillStyle = "#666";
    context.fillRect(0, 0, canvas.width, canvas.height);
    const data = canvas.toDataURL("image/jpeg");
    photo.setAttribute("src", data);

    audio.removeEventListener("ended", loopAudio);
    audio.pause();
    setTimeout(() => {
        pressStart.classList.remove("z-n1")
    }, 500);
}

function videoStarted() {
    const FPS = 3.2;
    timeout = setInterval(() => {
        width = video.width;
        height = video.height;
        canvas.setAttribute("width", width);
        canvas.setAttribute("height", width);
        context.drawImage(video, 0, 0, width, height);
        let data = canvas.toDataURL("image/jpeg", 0.4);
        let mydata = { image: data };
        //console.log(data);
        context.clearRect(0, 0, width, height);
        fetch("/detectImg", {
            method: "POST",
            body: JSON.stringify(mydata),
        })
            .then((response) => response.json())
            .then((json) => {
                if (videoPlaying) {
                    photo.setAttribute("src", json.image);
```

```
                checkAnimal(json.detectedList);
            }
        })
        .catch((error) => {
            console.log(error);
        });
    }, 1000 / FPS);
}
```

App.py

```
import json, os, pandas as pd
from flask import Flask, request, render_template, redirect, session
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
from yolov7model import get_yolov7, predict, base64_to_image, image_to_base64
import smtplib, ssl
from email.message import EmailMessage

app = Flask(__name__)
model = get_yolov7()
db_path = os.path.join(os.path.dirname(__file__), 'detectAnimal.db')
db_uri = 'sqlite:///{}'.format(db_path)
app.config['SQLALCHEMY_DATABASE_URI'] = db_uri
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
app.config['SECRET_KEY'] = "phoenixshri1234"
db = SQLAlchemy(app)

class DetectAnimals(db.Model):
    sno = db.Column(db.Integer, primary_key=True)
    animal_name = db.Column(db.String(20), nullable=False)
    date_of_detection = db.Column(db.String(20), nullable=False,
                                  default=datetime.now().strftime('%d-%m-%Y'))
    time_of_detection = db.Column(db.String(20), nullable=False,
```

```
                                        default=datetime.now().strftime('%H:%M:%S'))
    def __repr__(self) -> str:
        return f"{self.sno} - {self.animal_name}"

class User(db.Model):
    sno = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(30), nullable=False)
    password = db.Column(db.String(20), nullable=False)
    def __repr__(self) -> str:
        return f"{self.sno} - {self.name}"

with app.app_context():
    db.create_all()

def addtodb(animalList):
    if len(animalList) == 0 or animalList[0] == "person":
        return
    lastRec = DetectAnimals.query.order_by(DetectAnimals.sno.desc()).first()
    if lastRec != None and lastRec.animal_name == animalList[0]:
        currDateTime = datetime.now().strftime('%d-%m-%Y %H:%M:%S')
        prevDateTime = lastRec.date_of_detection + " " + lastRec.time_of_detection
        myformat = "%d-%m-%Y %H:%M:%S"
        curr = datetime.strptime(currDateTime, myformat)
        prev = datetime.strptime(prevDateTime, myformat)
        diff = curr - prev
        diffDays = diff.days
        diffmins = diff.seconds // 60
        if not (diffDays > 0 or diffmins > 1):
            return
    date_of_detection = datetime.now().strftime('%d-%m-%Y')
    time_of_detection = datetime.now().strftime('%H:%M:%S')
    animal = DetectAnimals(animal_name = animalList[0], date_of_detection =
                                date_of_detection, time_of_detection = time_of_detection)
    db.session.add(animal)
```

```
    db.session.commit()
    message = animalList[0] + " is detected on " + date_of_detection + " at " + time_of_detection
    sendMail(message)

emailSender = "shrinathkoraj@gmail.com"
passwordSender = "tvrxrlyokoonpasd"
emailReceiver = ["shrinathkoraj@gmail.com", "rtejashwini467@gmail.com",
"shriramheb7@gmail.com", "prathamesh56@gmail.com"]
email = EmailMessage()
email['From'] = emailSender
email['To'] = ", ".join(emailReceiver)
email['Subject'] = "Animal Detected"
def sendMail(body):
    email.set_content(body)
    context = ssl._create_unverified_context()
    with smtplib.SMTP_SSL("64.233.184.108", 465, context=context) as smtpserver:
        smtpserver.login(emailSender, passwordSender)
        smtpserver.sendmail(emailSender, emailReceiver, email.as_string())

@app.route('/')
def index():
    flag = "none"
    if session.get('flag'):
        flag = session['flag']
    session.clear()
    return render_template('index.html', flag=flag)

@app.route('/home')
def home():
    return render_template('home.html')

@app.route('/detect')
def detect():
    return render_template('detect.html')
```

```
@app.route('/signIn', methods=['POST'])
def signIn():
    username = request.form['signInUsername']
    password = request.form['signInPassword']
    exists = db.session.query(User.sno).filter_by(name=username, password=password).first()
is not None
    if(exists):
        session['flag'] = "exists"
        return redirect('/')
    user = User(name=username, password=password)
    db.session.add(user)
    db.session.commit()
    session['flag'] = "signSuccess"
    return redirect('/')

@app.route('/logIn', methods=['POST'])
def logIn():
    username = request.form['loginUsername']
    password = request.form['loginPassword']
    exists = db.session.query(User.sno).filter_by(name=username, password=password).first()
is not None
    if(exists):
        user = User.query.filter_by(name=username, password=password).first()
        session['username'] = user.name
        return redirect('/home')
    session['flag'] = "loginFail"
    return redirect('/')

@app.route('/saveReport', methods=['POST'])
def saveReport():
    base = os.path.dirname(os.path.abspath(__file__))
    data = request.json
    df = pd.DataFrame(data)
```

```
    with pd.ExcelWriter(base + '/static/assets/report.xlsx', mode='w') as writer:
        df.to_excel(writer, index=False)
    return {'message': 'Data saved successfully'}, 200

@app.route('/viewreport')
def viewreport():
    allAnimals = DetectAnimals.query.all()
    return render_template('viewreport.html', report=allAnimals)

@app.route('/delete/<int:sno>')
def delete(sno):
    animal = DetectAnimals.query.filter_by(sno=sno).first()
    db.session.delete(animal)
    db.session.commit()
    return redirect("/viewreport")

@app.route('/deleteAll')
def deleteAllRecords():
    DetectAnimals.query.delete()
    db.session.commit()
    return redirect('/viewreport')

@app.route('/detectImg', methods=['POST'])
def detectImg():
    jsonObj = json.loads(request.get_data().decode('UTF-8'))
    base64Image = jsonObj['image']
    image = base64_to_image(base64Image)
    (predictedImage, detectedList) = predict(model, image)
    base64Image = image_to_base64(predictedImage)
    jsonObj['image'] = base64Image
    jsonObj['detectedList'] = detectedList
    addtodb(detectedList)
    return jsonObj
```

```
if __name__ == '__main__':
    app.run(debug=True, port=8000)
```

Yolov7Model.py

```
import torch
import base64
import numpy as np
import os
import cv2

base = os.path.dirname(os.path.abspath(__file__))

CONFIDENCE = 0.40
# Bounding Boxes color scheme
ALPHA = 0.2
CELL_FILL = (0, 0, 200)
CELL_BORDER = (0, 0, 255)
_TEXT_THICKNESS_SCALING = 700.0
_TEXT_SCALING = 520.0
normalised_scaling = 1.0

def base64_to_image(base64_string):
    base64_data = base64_string.split(",")[1]
    image_bytes = base64.b64decode(base64_data)
    image_array = np.frombuffer(image_bytes, dtype=np.uint8)
    image = cv2.imdecode(image_array, cv2.IMREAD_COLOR)
    return image

def image_to_base64(image):
    result, frame_encoded = cv2.imencode(".jpg", image)
    jpg_as_text = base64.b64encode(frame_encoded).decode()
    b64_src = "data:image/jpeg;base64,"
    base64Image = b64_src + jpg_as_text
```

```
    return base64Image

def get_yolov7():
    model = torch.hub.load(base + '/yolov7', 'custom', base + '/yolov7/yolov7.pt', source='local')

def predict(model, image_path):
    results = model(image_path)
    df = results.pandas().xyxy[0]
    boxes = []
    detectedList = []
    for _, row in df.iterrows():
        if (row['confidence'] > CONFIDENCE) and ((row['class'] == 0) or (row['class'] < 23 and
                row['class'] > 13)):
            boxes.append([int(row['xmin']), int(row['ymin']), int(row['xmax']), int(row['ymax']),
                        row['name'] + " " + str(int(float(row['confidence']) * 100)) + "%"])
            detectedList.append(row['name'])

    image = image_path
    overlay = image.copy()
    for table_bbox in boxes:
        cv2.rectangle(overlay, (table_bbox[0], table_bbox[1]),
              (table_bbox[2], table_bbox[3]), CELL_FILL, -1)
        cv2.rectangle(image, (table_bbox[0], table_bbox[1]),
              (table_bbox[2], table_bbox[3]), CELL_BORDER, 2)
        thickness = int(round((image.shape[0] * image.shape[1]) /
                (_TEXT_THICKNESS_SCALING * _TEXT_THICKNESS_SCALING)) *
                normalised_scaling)
        thickness = max(1, thickness)
        scaling = image.shape[0] / _TEXT_SCALING * normalised_scaling
        size = cv2.getTextSize(table_bbox[4], cv2.FONT_HERSHEY_SIMPLEX, scaling,
                        thickness) [0]
        cv2.putText(image, table_bbox[4], (table_bbox[0], table_bbox[1] - int(size[1] * 0.4)),
                    cv2.FONT_HERSHEY_SIMPLEX,
                    scaling,color=(255,255,255),thickness=thickness)
```

```
image_new = cv2.addWeighted(overlay, ALPHA, image, 1-ALPHA, 0)
return (image_new, detectedList)
```

Chapter 9

TESTING AND RESULTS

9.1. Testing

Testing is an important phase in the development life cycle of the product; this is phase where the error remaining from all the phases was detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software. During the testing, the program the be tested was executed with a set of test cases and the output of the program for the test cases was evaluated to determine whether the program is performing as expected. Errors were found and corrected by using the following testing steps and correction was recorded for future references. Thus, a series of testing was performed on the system before it was ready for implementation.

Types of Testing

In these types of testing all major activities are described below.

1. Unit testing
2. Integration testing

9.1.1. Unit testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

9.1.2. Integration testing

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units. Integration testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

9.2. Results

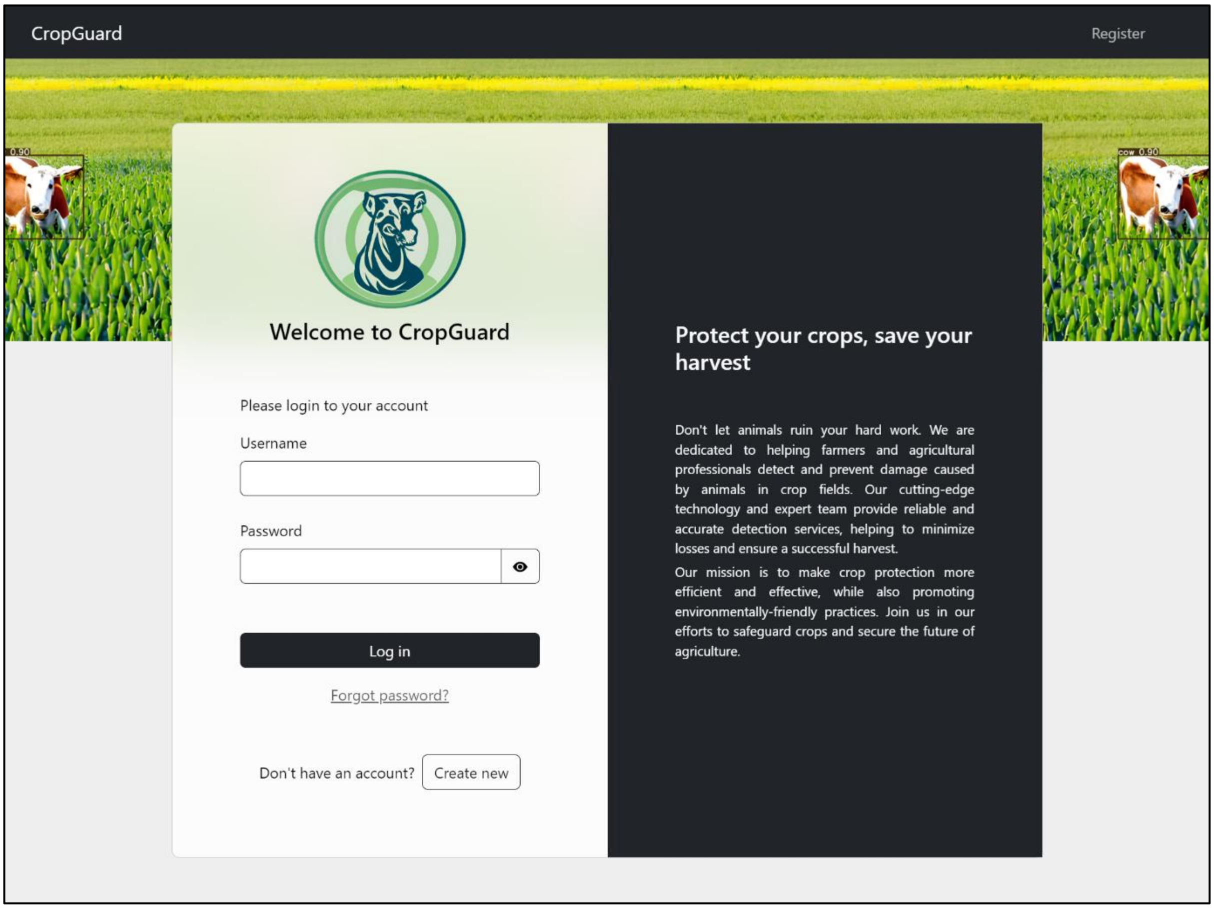

Fig 9.2.1. Index Page of Crop Guard Website

The index page is the landing page of the website. It serves as a gateway for the user to access different sections of the website. The page consists of a login form that allows users to log in to their account, and a register button that help users to create a new account. The page also contains a section that provides information about the website, including our mission and vision.

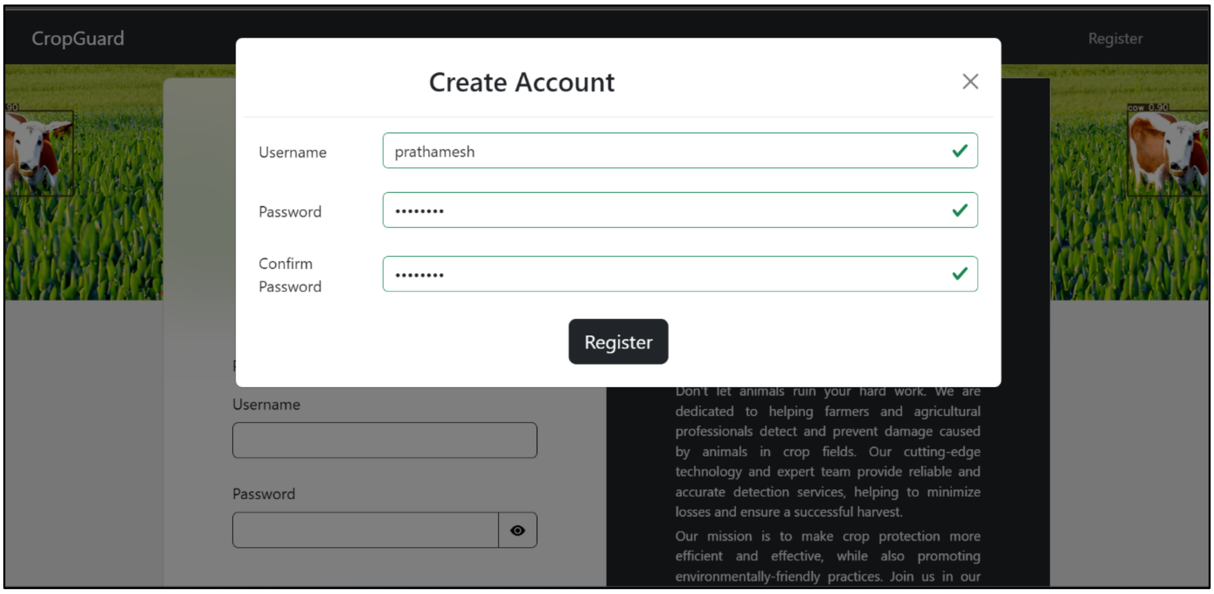

Fig 9.2.2. Register Page of Crop Guard Website

This page allows users to create a new account by submitting their personal information such as name and password. It includes validation to ensure that required fields are filled out and that the password meets certain criteria.

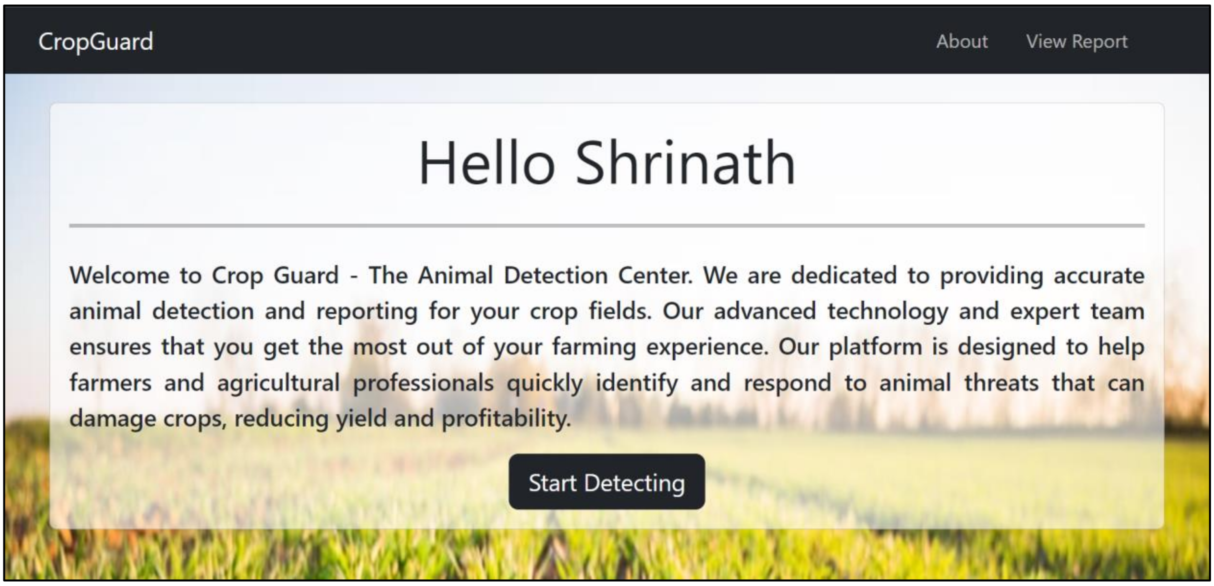

Fig 9.2.3. Home Page of Crop Guard Website

This is a homepage of our web application called Crop Guard - Animal Detection in Crop Fields. The homepage provides a clear and concise introduction to the web application and its features, making it easy for users to understand and navigate.

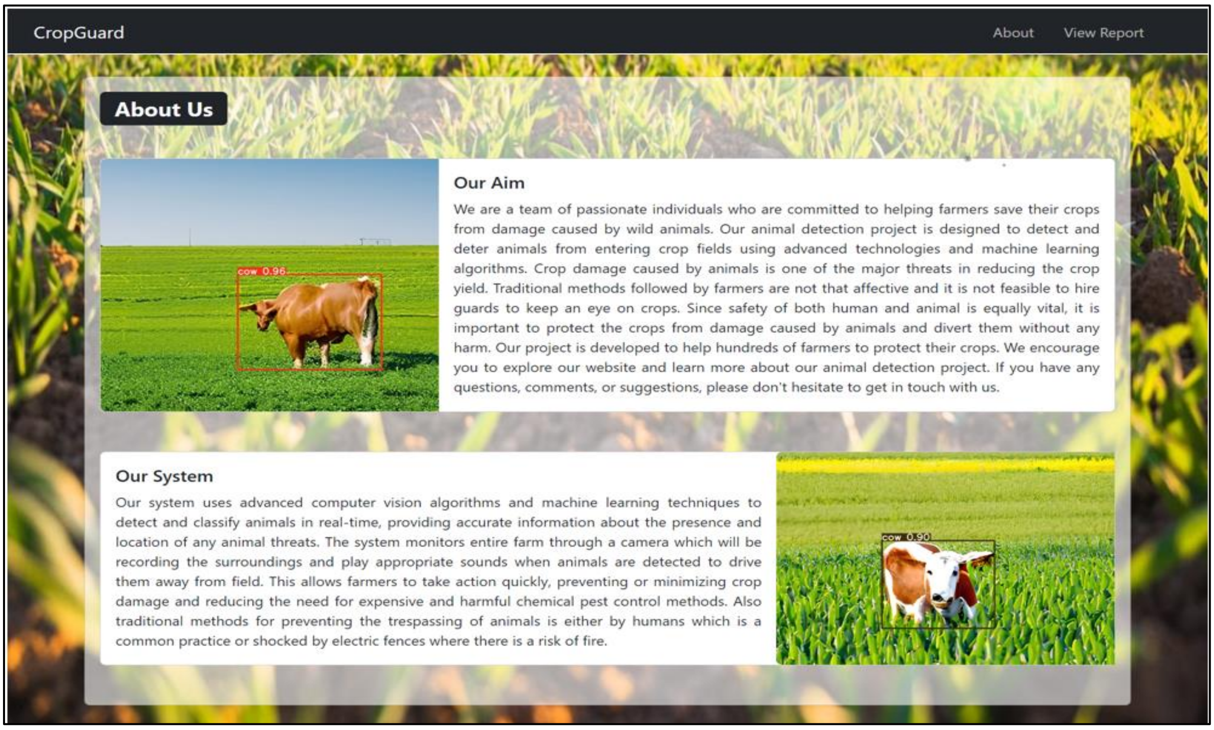

Fig 9.2.4. About Us Page of Crop Guard Website

The About Us section provides a brief overview of the project's aim and encourages users to explore the website further.

Fig 9.2.5. Animal Detection Page of Crop Guard Website

Here, users can use their webcams to capture live video, which is then analysed to detect any animals present in the frame. It also contains a series of buttons allowing users to start and stop the video feed, and to return to the home page.

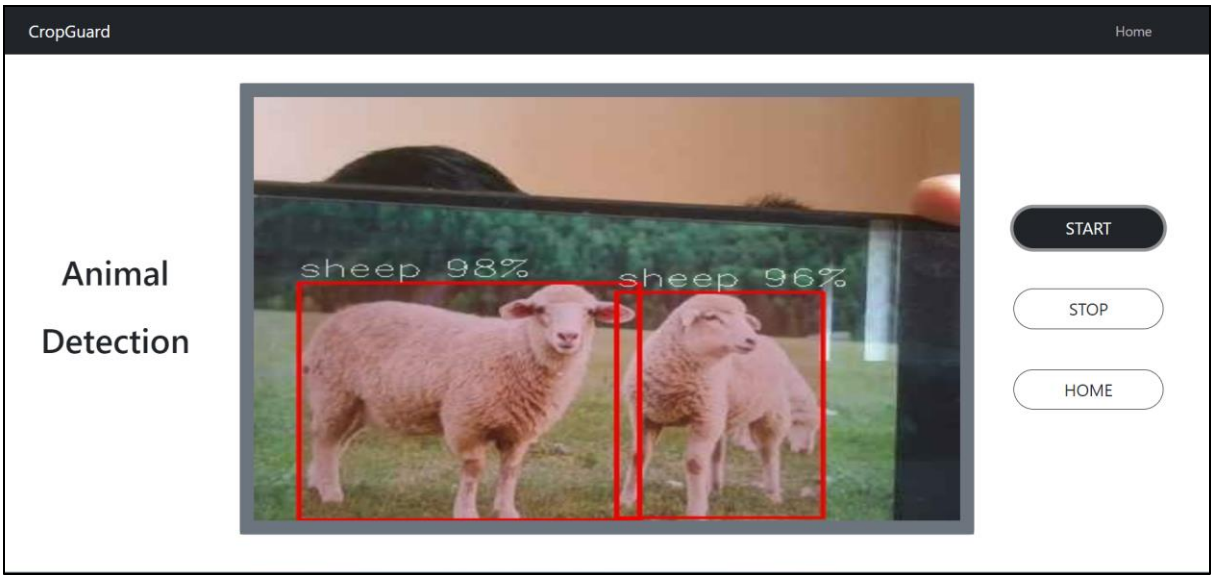

Fig 9.2.6. Live Detection Image

Observing the live detection image, we could see that the system was able to accurately identify and highlight animals present in the frame. The system was able to detect a variety of animals such as dogs, cats, elephants, bears, horses, and cows, and played an appropriate audio to repel them.

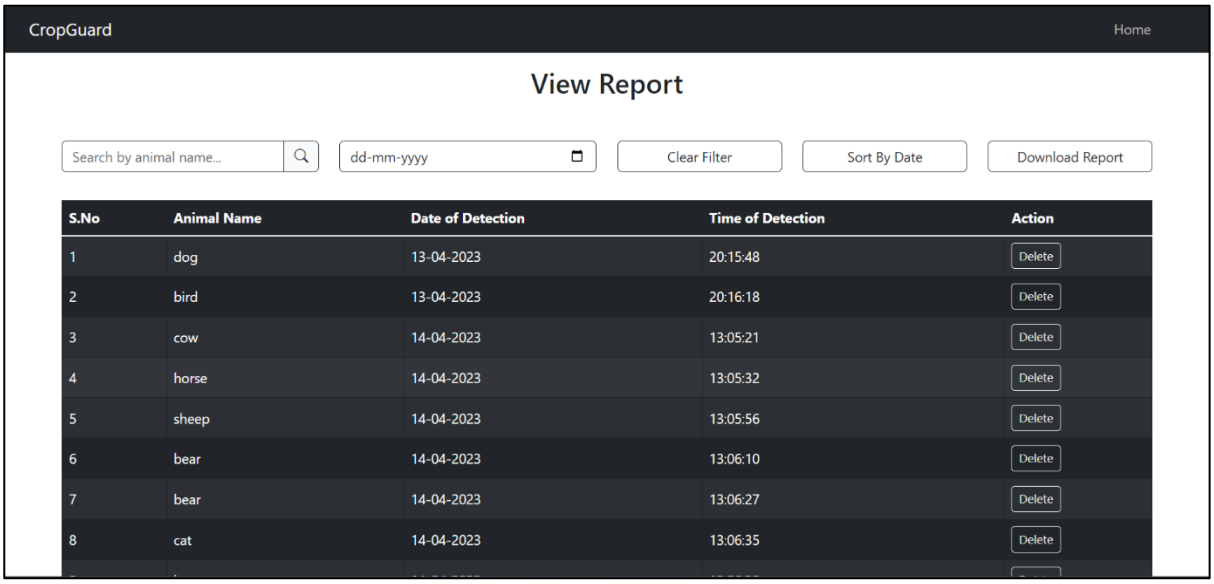

Fig 9.2.7. Report Page of Crop Guard Website

The Report page is a central hub where users can see a comprehensive table of all animal detections. The table contains a search bar that enables the user to search for a specific animal by name, and a filter that allows the user to sort the table by date.

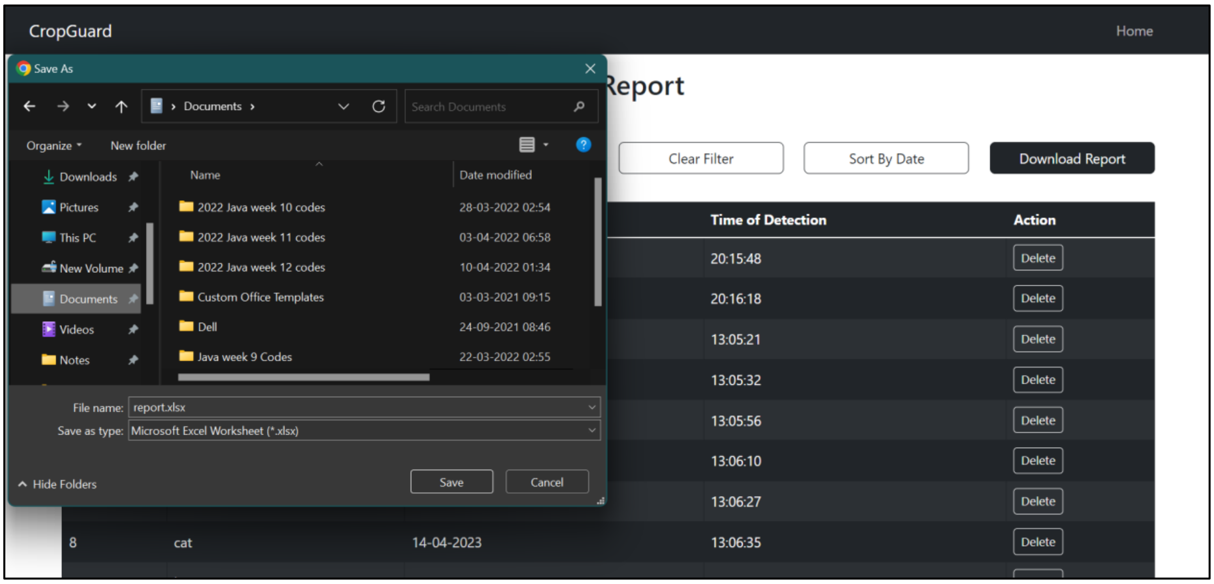

Fig 9.2.8. Download Report of Detected Animals

The table displays the animal's name, date, and time of detection, as well as a button to delete the corresponding record. The report can be downloaded as an Excel file for further analysis or record keeping.

9.3. Comparison Table

REFERENCE PAPER	APPROACH	ACCURACY
[5] Adami, D., Ojo, M. O., & Giordano, S. (2021). Design, Development and Evaluation of an Intelligent Animal Repelling System for Crop Protection Based on Embedded Edge-AI. *IEEE Access*, 9, 132125-132139.	YOLO v3, PIR sensor	82.50 %
[11] Balakrishna, K., Mohammed, F., Ullas, C. R., Hema, C. M., & Sonakshi, S. K. (2021). Application of IOT and machine learning in crop protection against animal intrusion. *Global Transitions Proceedings, 2(2)*, 169-174.	SDD	89.32 %
[16] Schindler, F., & Steinhage, V. (2021). Identification of animals and recognition of their actions in wildlife	Mask R-CNN,	63.80 %

	videos using deep learning techniques. *Ecological Informatics*, 61, 101215.	Infrared Camera,	
[18]	Banupriya, N., Saranya, S., Swaminathan, R., Harikumar, S., & Palanisamy, S. (2020). Animal detection using deep learning algorithm. *J. Crit. Rev*, 7(1), 434-439.	CNN	86.79 %
[20]	Kekre, O., Paliwal, A., Mehatkar, S., & Hazare, R. (2019). Ethical Crop Protection.	KNN, PIR sensor	82.70 %
	Proposed Model	YOLO v7, LSGAN	96.00 %

Table 9.3.1. Accuracy Comparison Table

The comparison table shows that our model outperformed previous models in terms of detecting animals in crop fields. Our model demonstrated a higher accuracy rate of 96% compared to the other models. These results suggest that our model is better suited for detecting animals in crop fields, which can be beneficial for farmers in preventing crop damage caused by animals.

Chapter 10

CONCLUSION

The "Crop Guard - Animal Detection in Crop Fields" project aimed to develop an effective and user-friendly software that can detect and repel animals from crop fields and drives them away using an alarm system. Crop damage caused by animal attacks is a significant threat to crop yield, and traditional methods are not effective or feasible. The project utilized the LSGAN algorithm to generate synthetic images of animals in crop fields, which were combined with real images to create a diverse training dataset for the object detection model. The YOLOv7 algorithm was used for animal detection due to its high accuracy and efficiency. The model was successfully trained by enhancing the training dataset and using a state-of-the-art object detection algorithm to achieve an accuracy of 96% which compared with existing methods and showed superior performance in terms of accuracy and speed. The trained model was integrated with a camera system that captures real-time footage of the crop fields. The object detection model detects animals in the captured frames and provides their location and type. Based on the type of animal detected, the system generates specific sounds to repel the animal and prevent crop damage. The proposed system offers a fast, cost-effective, and practical solution to the problem of animal intrusion in crop fields. The integrated sound generation system further enhances the system's ability to prevent crop damage caused by animal intrusion. The proposed system has the potential to support farmers in protecting their crops from animal damage while ensuring the safety of both human and animal. This project contributes to the ongoing efforts towards sustainable agriculture and can have a significant impact on crop yield and food security. To enhance the effectiveness of our animal detection system, future work will involve the integration of cameras with additional sensors to detect animals in all weather conditions. The current system relies solely on visual data captured by the camera, which may be limited during extreme weather conditions such as night, heavy fog, rain, or snow. By combining camera data with sensors that can detect animal heat signatures or motion, the system's accuracy and reliability can be improved. By incorporating these improvements, our system will continue to evolve and address the challenges of animal intrusion in crop fields.

REFERENCES

[1] Dewi, C., Chen, R. C., Liu, Y. T., Jiang, X., & Hartomo, K. D. (2021). Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various GAN. *IEEE Access, 9*, 97228-97242.

[2] Ma, Y., Liang, Y., Zhang, W., & Yan, S. (2019, November). SAR target recognition based on transfer learning and data augmentation with LSGANs. In 2019 *Chinese Automation Congress (CAC)* (pp. 2334-2337). IEEE.

[3] Sabeenian, R., Deivanai, N., & Mythili, B. (2020). Wild animals intrusion detection using deep learning techniques. *Int. J. Pharm. Res*, 12(4), 1053-1058.

[4] Ranparia, D., Singh, G., Rattan, A., Singh, H., & Auluck, N. (2020, November). Machine learning-based Acoustic Repellent System for Protecting Crops against Wild Animal Attacks. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)* (pp. 534-539). IEEE.

[5] Adami, D., Ojo, M. O., & Giordano, S. (2021). Design, Development and Evaluation of an Intelligent Animal Repelling System for Crop Protection Based on Embedded Edge-AI. *IEEE Access*, 9, 132125-132139.

[6] Iniyaa, K. K., Divya, J. K., Devdharshini, S., & Sangeethapriya, R. (2021). Crop Protection from Animals Using Deep Learning. *International Journal of Progressive Research in Science and Engineering*, 2(3), 41-44.

[7] Dev, N. G., Sreenesh, K. S., & Binu, P. K. (2019, July). Iot based automated crop protection system. In *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)* (Vol. 1, pp. 1333-1337). IEEE.

[8] Santhiya, S., Dhamodharan, Y., Priya, N. E. K., Santhosh, C. S., & Surekha, M. (2018). A smart farmland using Raspberry Pi crop prevention and animal intrusion detection system. *Int. Res. J. Eng. Technol*, 5(3), 3829-3832.

[9] Nikhil, R., Anisha, B. S., & Kumar, R. (2020, July). Real-time monitoring of agricultural land with crop prediction and animal intrusion prevention using internet of things and machine learning at edge. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)* (pp. 1-6). IEEE.

[10] Deepa, S., Alagulakshmi, G., Illakkiavathy, R., Keerthana, M., & Preethi, K. S. (2021). Implementation of Crop Protection System against Wild Animals Attack. *Annals of the Romanian Society for Cell Biology*, 3638-3646.

[11] Balakrishna, K., Mohammed, F., Ullas, C. R., Hema, C. M., & Sonakshi, S. K. (2021). Application of IOT and machine learning in crop protection against animal intrusion. *Global Transitions Proceedings*, 2(2), 169-174.

[12] Thirrunavukkarasu, R. R., Meeradevi, T., Prabhu, S. G., Arunachalam, J., & Prasath, R. (2021, March). Smart irrigation and crop protection using arduino. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 639-643). IEEE.

[13] Yadahalli, S., Parmar, A., & Deshpande, A. (2020, July). Smart intrusion detection system for crop protection by using Arduino. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 405-408). IEEE.

[14] Findlay, L. J., & Hill, R. A. (2021). Field guarding as a crop protection method: preliminary implications for improving field guarding. *Human–Wildlife Interactions*, 14(3), 20.

[15] Giordano, S., Seitanidis, I., Ojo, M., Adami, D., & Vignoli, F. (2018, March). IoT solutions for crop protection against wild animal attacks. In *2018 IEEE international conference on Environmental Engineering (EE)* (pp. 1-5). IEEE.

[16] Schindler, F., & Steinhage, V. (2021). Identification of animals and recognition of their actions in wildlife videos using deep learning techniques. *Ecological Informatics*, 61, 101215.

[17] Lee, S., Song, Y., & Kil, S. H. (2021). Feasibility analyses of real-time detection of wildlife using UAV-derived thermal and rgb images. *Remote Sensing*, 13(11), 2169.

[18] Banupriya, N., Saranya, S., Swaminathan, R., Harikumar, S., & Palanisamy, S. (2020). Animal detection using deep learning algorithm. *J. Crit. Rev*, 7(1), 434-439.

[19] Dihingia, U., Amar, P., Shyam, M. M., Thomas, V., & Chidambaram, S. (2020). Animal identification using deep learning on raspberry Pi. *Int. J. Res. Eng., Sci. Manage*., 3(3), 454-456.

[20] Kekre, O., Paliwal, A., Mehatkar, S., & Hazare, R. (2019). Ethical Crop Protection.

[21] Saravanan, K., Sharanya, V., & Lakshmi, S. V. SMART CROP PROTECTION SYSTEM FROM WILD ANIMALS USING ARM.

[22] Borah, S., Kumar, R., & Mukherjee, S. (2020, December). Study of RTPPS algorithm in UWB communication medium for a surveillance system to protect agricultural crops from wild animals. In *2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)* (pp. 121-126). IEEE.

[23] Lathesparan, R., Sharanjah, A., Thushanthi, R., Kenurshan, S., Nifras, M. N. M., & Wickramaarach, W. U. (2020). Real-time Animal Detection and Prevention System for Crop Fields.

[24] Lakshmi, K. M., Raja, C., Sreekanth, D., & Renuka, N. (2020). Security for Protecting Agricultural Crops from Wild Animals using GSM Technology. *Journal of Shanghai Jiaotong University*, 16(7).

[25] Navaneetha, P., Devi, R. R., Vennila, S., Manikandan, P., & Saravanan, D. S. (2020). IOT Based Crop Protection System against Birds and Wild Animal Attacks. *International Journal of Innovative Research In Technology*, 6(11).