INTELLIGENT SURVEILLANCE AND SPECIES DETECTION SYSTEM FOR CROP PROTECTION FROM WILD ANIMALS

PROJECT REPORT

SUBMITTED IN FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

by

AGHIL V S (VAS21EC011)
AKSHAYA T S (VAS21EC015)
AKSHAY KRISHNA K (VAS21EC017)
ATHUL T S (LVAS21EC111)

(AN ISO 9001:2015 CERTIFIED INSTITUTION)

Department of Electronics and Communication Engineering

Vidya Academy of Science & Technology
Thalakkottukara, Thrissur - 680 501
(http://www.vidyaacademy.ac.in)

April 2025

Department of Electronics and Communication Engineering
Vidya Academy of Science & Technology
Thalakkottukara, Thrissur - 680 501
(http://www.vidyaacademy.ac.in)

(AN ISO 9001:2015 CERTIFIED INSTITUTION)

Certificate

This is to certify that the project report titled **"INTELLIGENT SURVEILLANCE AND SPECIES DETECTION SYSTEM FOR CROP PROTECTION FROM WILD ANIMALS"** is a bonafide record of the work carried out by **AKSHAYA T S (VAS21EC015)** of Vidya Academy of Science & Technology, Thalakkottukara, Thrissur - 680 501 in partial fulfillment of the requirements for the award of **Degree of Bachelor of Technology** in **Electronics and Communication Engineering** of **APJ Abdul Kalam Technological University**, during the academic year 2024-2025. The Project report has been approved as it satisfies the academic requirements in the respect of project work prescribed for the said degree.

Project Guide/Supervisor	**Project Coordinator**
Ms. Aswathy E R	Mr. Anil M
Asst. Prof., Dept. of ECE	Asst. Prof., Dept. of ECE

Head of Department

Dr. S. Swapna Kumar

Prof.and Head, Dept. of ECE

Undertaking

We hereby declare that the project report **INTELLIGENT SURVEILLANCE AND SPECIES DETECTION SYSTEM FOR CROP PROTECTION FROM WILD ANIMALS** submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of **Ms.Aswathy E R** and **Mr. Anil M.**

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University (2024-2025).

AGHIL V S (VAS21EC011)

AKSHAYA T S (VAS21EC015)

Thrissur-680501

AKSHAY KRISHNA K (VAS21EC017)

April 2025

ATHUL T S (LVAS21EC111)

Acknowledgement

We would like to extend our gratitude to everyone who supported our project. Their guidance and encouragement were invaluable in completing this work.

First, we express sincere thanks to our project guide, **Ms.Aswathy E R**, Asst. Prof., Dept. of ECE, for their insightful advice and guidance.

We also wish to thank the project coordinators, **Mr. Anil M**, Asst. Prof., Dept. of ECE, and **Ms. Anju Vincent**, Asst. Prof., Dept. of ECE, for their valuable feedback.

Our heartfelt appreciation goes to the faculty members of our department for their support, and to our friends for their encouragement.

We are thankful to **Dr. S. Swapna Kumar**, Head of the Electronics and Communication Department, and our Principal, **Dr Sunitha C.**, for their cooperation throughout this work.

Finally, we thank God Almighty for His blessings in this endeavor.

AGHIL V S (VAS21EC011)
AKSHAYA T S (VAS21EC015)
AKSHAY KRISHNA K (VAS21EC017)
ATHUL T S (LVAS21EC111)
Eighth Semester B.Tech (2021 Admissions)
Vidya Academy of Science & Technology
April 2025 Thrissur - 680 501.

Abstract

The project ”Protection of Crops from Wild Animals Using Intelligent Surveillance System” focuses on developing a system that mitigates crop damage caused by wildlife. By utilizing advanced surveillance technologies such as motion sensors, infrared cameras, and artificial intelligence, the system detects the presence of animals near farmland in real-time. A machine learning model is employed to identify the species and determine if it poses a threat to the crops. If a harmful species is detected, protective measures are triggered, including electronic firecrackers for large animals like elephants .Strategically placing sound deterrents around the crop field can keep animals away effectively. For elephants, loud noises like firecrackers or drums work well. To deter sheep, use sounds of barking dogs or loud horns. Pigs can be deterred by clanging metal objects or recordings of predators. High-pitched noises or predator calls can help keep deer away from the crops. Farmers are provided with realtime image and video feeds via smart devices, along with environmental data such as humidity and temperature. Additionally, the system uses face recognisation technology to recognize authorized personnel, automatically recording the attendance of farm workers if detected. This intelligent and automated approach ensures efficient crop protection by targeting harmful wildlife while minimizing unnecessary interventions .

Contents

Acknowledgement **i**

Abstract **i**

List of Figures **vi**

Symbols and Abbreviations **vii**

1 INTRODUCTION **1**

1.1 INTRODUCTION 1

1.2 PROJECT BACKGROUND AND MOTIVATION 1

1.3 SUMMARY 2

2 LITERATURE REVIEW **4**

2.1 INTRODUCTION 4

2.2 WILDLIFE DAMAGE IN AGRICULTURE 4

2.3 SURVEILLANCE TECHNOLOGIES 5

2.4 OBJECT DETECTION AND RECOGNITION 6

2.5 REAL-TIME DATA PROCESSING 7

2.6 RESPONSE MECHANISMS 7

2.7 ENVIRONMENTAL MONITORING 8

2.8 PERSONNEL AUTHORIZATION AND SAFETY 8

2.9 SUMMARY 8

3 ETHICAL AND ENVIRONMENTAL CONSIDERATIONS **9**

3.1 INTRODUCTION 9

3.2 ETHICAL CONCERNS IN AI-BASED SURVEILLANCE . . . 9
3.3 WILDLIFE AND ENVIRONMENTAL IMPACT . . . 10
3.4 SUSTAINABILITY IN AI-BASED SURVEILLANCE . . . 12
3.5 LEGAL AND REGULATORY FRAMEWORKS . . . 12
3.6 CONCLUSION . . . 13

4 DESIGN METHODOLOGY 14
4.1 INTRODUCTION . . . 14
4.2 IMAGE PROCESSING . . . 14
4.2.1 Data collection . . . 15
4.2.2 Image Preprocessing . . . 16
4.2.3 Importing Module . . . 17
4.2.4 Capturing The Images Of Animals . . . 18
4.2.5 Camera Interfacing . . . 19
4.3 CHALLENGES IN CAMERA INTERFACING . . . 20
4.4 BLOCK DIAGRAM . . . 20
4.5 CIRCUIT DIAGRAM . . . 21

5 HARDWARE COMPONENTS 23
5.1 ESP32-WROOM . . . 23
5.2 RASPBERRY PI 4B . . . 25
5.3 DHT11 TEMPERATURE AND HUMIDITY SENSOR . . . 26
5.4 SOIL PH METER . . . 27
5.5 2-CHANNEL RELAY MODULES . . . 28
5.6 SERVO MOTOR (MG995) . . . 29
5.7 ELECTRIC CRACKER . . . 30

6 SOFTWARE AND ALGORITHMS 31
6.1 FACE RECOGNITION . . . 31
6.1.1 OpenCV . . . 31
6.1.2 Haar Cascade Classifier . . . 32

6.1.3 LBPH Face Recognition 32
6.2 OBJECT DETECTION 33
6.2.1 TensorFlow Lite for Object Detection 33
6.2.2 Pre-Trained Model and Dataset 34
6.2.3 Object Detection Applications 34

7 HAAR CASCADE CLASSIFIER 35

7.1 HAAR-LIKE FEATURES AND INTEGRAL IMAGES 35
7.2 CASCADE OF CLASSIFIERS 36
7.3 IMPLEMENTATION AND FUTURE ENHANCEMENTS 37

8 LOCAL BINARY PATTERNS HISTOGRAM(LBPH) 38

8.1 OVERVIEW OF LBPH ALGORITHM 38
8.1.1 Steps of the LBPH Algorithm 39
8.2 ADVANTAGES OF LBPH 39
8.3 IMPLEMENTATION IN THE SURVEILLANCE SYSTEM 40
8.3.1 Training the Model 40
8.3.2 Recognition Process 40
8.4 LIMITATION OF LBPH 41
8.5 FUTURE ENHANCEMENTS 41

9 REAL-TIME PROCESSING AND NOTIFICATION 43

9.1 REAL-TIME PROCESSING 43
9.1.1 Video Capture and Preprocessing 43
9.1.2 Face and Object Detection 44
9.1.3 Decision Making and Control Logic 44
9.2 NOTIFICATION SYSTEM 45
9.2.1 Blink App Integration 45
9.2.2 Types of Notifications 45
9.2.3 User Interaction with the Notification System 46
9.3 CHALLENGES AND OPTIMIZATIONS 47

10 IMPLEMENTATION CHALLENGES AND SOLUTIONS **48**

10.1 INTRODUCTION 48

10.2 HARDWARE CHALLENGES AND SOLUTIONS 48

10.3 SOFTWARE AND ALGORITHM CHALLENGES 49

10.4 DEPLOYMENT AND ENVIRONMENTAL CHALLENGES 50

11 RESULT AND DISCUSSION **51**

11.1 INTRODUCTION 51

11.2 EXPERIMENTAL SETUP 51

11.3 DETECTION ACCURACY ANALYSIS 52

11.4 RESPONSE TIME ANALYSIS 52

11.5 DETECTION PROCESS AND LABELING 53

11.6 ENVIRONMENTAL IMPACT ON DETECTION PERFORMANCE . 55

11.7 LIMITATIONS AND FUTURE ENHANCEMENTS 55

12 CONCLUSION **56**

13 FUTURE WORK **57**

REFERENCES **58**

APPENDIX **60**

List of Figures

4.1 Block diagram of animal detection 15

4.2 Block diagram of entire system 21

5.1 ESP32-WROOM 24

5.2 Raspberry pi 4B 26

5.3 DTH11 27

5.4 Soil ph three wave meter 28

5.5 2-Channel relay module 29

5.6 SERVO MOTOR (MG995) 30

7.1 Haar-Cascade Classifie 36

9.1 Screen shot of blink app 46

11.1 Example of Animal Detection and Labeling in Real-Time Processing . 54

11.2 Example of Animal Detection and Labeling in Real-Time Processing . 54

Symbols and Abbreviations

Abbreviations

CNN	Convolutional Neural Network
Open C V	Open Source Computer Vision Library
IDE	integrated development environment
Numpy	Numerical python
RGB	Red, Green, Blue color
AI	Artificial Intelligence

Chapter 1

INTRODUCTION

1.1 INTRODUCTION

Agriculture is a backbone of many economies, particularly in rural areas, where it serves as a primary source of income and food security. However, one of the major challenges faced by farmers is the destruction of crops by wild animals. This conflict not only leads to significant economic losses but also creates tension between wildlife conservation and agricultural productivity. Traditional methods of deterring wild animals, such as manual patrolling or physical barriers, are often ineffective or unsustainable, prompting the need for more advanced solutions.

1.2 PROJECT BACKGROUND AND MOTIVATION

Agriculture is a crucial sector for food production and rural livelihoods, yet it faces persistent challenges, one of the most significant being crop damage caused by wild animals. Farmers, especially those near forests or wildlife habitats, frequently experience substantial losses when animals like elephants, wild boars, and deer encroach on their fields. This not only threatens food security but also leads to human-wildlife conflicts, sometimes resulting in dangerous confrontations. Traditional methods like physical barriers, manual patrolling, and scarecrows have been widely used to deter animals, but these solutions are often temporary, labor-intensive, and become less effective over time as animals adapt to them.

With recent advancements in technology, particularly in the fields of artificial intelligence (AI), machine learning (ML), and real-time surveillance systems, there is now the potential to develop more effective and sustainable methods for protecting crops. These technologies can automate the detection of wildlife near farmlands, reduce the need for constant human intervention, and provide precise responses to potential threats. This has laid the foundation for the *"Protection of Crops from Wild Animals Using Intelligent Surveillance System"* project.

The motivation behind this project arises from the pressing need to address the growing problem of wildlife-induced crop damage in a way that benefits both farmers and the environment. Farmers experience severe economic losses due to crop destruction, and existing methods of crop protection are either inefficient or harmful to wildlife. This project aims to leverage AI and ML to create a system that can intelligently monitor fields, identify animal species, and deploy appropriate deterrents based on the threat level. For example, large animals like elephants can be scared off using loud noises such as electronic firecrackers, while smaller animals like wild pigs and deer can be deterred with less invasive methods such as sound-based deterrents or harmless sprays.

This system not only protects crops effectively but also provides real-time data to farmers via smart devices, allowing them to monitor their fields remotely. Additionally, the system includes environmental sensors and worker recognition technology, making farm management more streamlined. The broader motivation is to create a solution that balances agricultural needs with wildlife conservation, ensuring that crops are protected in a way that minimizes harm to the ecosystem. Through this innovative approach, the project aims to promote sustainable farming practices and reduce the human-wildlife conflict that affects rural communities worldwide.

1.3 SUMMARY

The system works by continuously monitoring farm areas using real-time surveillance. It detects the presence of animals near the crops using motion sensors and cameras. A machine learning model is employed to identify the species, and if the species poses a threat, appropriate protective measures are triggered. These include electronic firecrackers for large animals like elephants, and non-harmful deterrents such as sound systems or sprays for smaller animals like wild pigs and deer. The system's core functionality relies on machine learning-based image processing to recognize different animal species. When motion sensors or cameras detect movement near the farmland, the system captures the visual data and processes it using AI models trained to identify various animals such as elephants, wild boars, and deer. Based on the identified species, the system determines the level of threat and triggers an appropriate response. For example, large animals like elephants can be deterred using electronic firecrackers or ultrasonic sounds, while smaller animals like deer or wild boars can be repelled using sound alarms or harmless chemical sprays.

To ensure that farmers remain well-informed, the system integrates with smart devices, allowing real-time monitoring and control. Farmers receive instant alerts along with image and video feeds of detected animals, helping them take additional preventive measures if needed. The system also includes environmental sensors to provide critical data such as temperature, humidity, and weather conditions, ensuring comprehensive farm management. Additionally, a face recognition feature is implemented to log and verify the presence of authorized personnel and farm workers, enhancing security.

This project represents a technologically advanced, automated, and sustainable approach to wildlife deterrence. By leveraging AI, IoT, and machine learning, the system reduces human-wildlife conflicts, minimizes crop losses, and promotes eco-friendly farming practices, ultimately benefiting both farmers and wildlife conservation efforts. Farmers are kept informed through smart devices, where they receive real-time image and video feeds of the field along with environmental data such as humidity and temperature. Additionally, the system includes a face recognition feature to log the presence of authorized personnel and farm workers, providing a comprehensive and automated solution for both crop protection and farm management.

This project represents an intelligent, sustainable, and technologically advanced approach to safeguarding agricultural land from wildlife interference, benefiting both farmers and ecosystems.

Chapter 2

LITERATURE REVIEW

2.1 INTRODUCTION

Agriculture plays a crucial role in food production and the economy, but one of the most persistent challenges faced by farmers is crop damage caused by wild animals. This human-wildlife conflict results in significant financial losses, food shortages, and sometimes even dangerous encounters between farmers and animals. Traditional crop protection methods such as fencing, scarecrows, and manual patrolling are often ineffective in the long run. The integration of advanced surveillance technologies, artificial intelligence (AI), and machine learning (ML) offers a promising solution to minimize crop damage while ensuring ecological balance.

This chapter reviews existing studies on wildlife damage in agriculture, advancements in surveillance technology, real-time data processing, object detection techniques, response mechanisms, environmental monitoring, and security aspects related to personnel authorization. These aspects collectively form the foundation of this research and contribute to the development of an effective Intelligent Surveillance and Species Detection System for Crop Protection from Wild Animals.

2.2 WILDLIFE DAMAGE IN AGRICULTURE

Wildlife interference in agricultural lands has been a persistent issue worldwide. In many regions, animals such as elephants, wild boars, deer, monkeys, and birds invade farms, feeding on crops and causing extensive damage. A study by Barua et al. (2013) found that elephants in India cause substantial crop losses, affecting not only the livelihoods of farmers but also increasing hostility towards wildlife conservation efforts.

Similarly, Conover (2002) explored the effects of wildlife on agricultural productivity across different regions, noting that the economic impact of crop raiding by wild animals can account for over 10 percentage of total crop losses annually. The study also identified that smaller animals such as rodents and birds can contribute to significant losses, particularly in grain and fruit farming.

A research paper by Karanth et al. (2017) highlighted that human-wildlife conflicts are becoming more frequent due to habitat loss and climate change, pushing animals to venture into farmlands in search of food. The study emphasized that technological solutions, such as automated surveillance and AI-based deterrent systems, can help mitigate these conflicts in a more sustainable manner.

These studies highlight the need for intelligent and automated systems that can detect and prevent wildlife intrusion without harming the animals or disrupting the natural ecosystem.

2.3 SURVEILLANCE TECHNOLOGIES

Surveillance plays a vital role in monitoring and preventing animal intrusions into agricultural fields. Traditional surveillance techniques involved manual patrolling and human observation, which are labor-intensive and ineffective over large areas. Modern technology-driven solutions, such as motion-activated cameras, infrared sensors, and AI-based monitoring, have significantly improved farm security.

A study by Lescureux and Linnell (2010) demonstrated how motion sensors and infrared cameras can be used to monitor the movements of wild animals, allowing farmers to take timely preventive measures. These technologies are particularly effective in remote farms where constant human supervision is not feasible.

Further research by Zhao et al. (2019) showcased how thermal imaging cameras can detect animals even in low-light conditions, making them highly suitable for nighttime surveillance. The study found that integrating thermal sensors with AI-based species identification algorithms improved the accuracy of wildlife detection by 87

Another advancement in drone-based surveillance has been studied by Gonzalez et al. (2021), who demonstrated the use of drones equipped with high-resolution cameras and AI for real-time monitoring of large agricultural areas. Drones can provide a bird's-eye view of the farm, detect animal movements, and relay alerts to farmers instantly.

These findings indicate that a combination of motion sensors, infrared cameras, and AI-based monitoring can significantly enhance farm security and reduce the likelihood of crop damage by wildlife.

2.4 OBJECT DETECTION AND RECOGNITION

The ability to detect and recognize animals is critical in preventing wildlife intrusion into farmlands. With recent advancements in deep learning and computer vision, several object detection techniques have been developed to identify animals in real-time.

One of the most effective models is the Convolutional Neural Network (CNN), which is widely used for object recognition. Howard et al. (2017) explored the effectiveness of MobileNet, a lightweight CNN model, for real-time object detection in embedded systems. The study found that MobileNet achieved high accuracy in identifying different species of animals while consuming minimal computational power, making it suitable for IoT-based farm monitoring systems.

Another important approach is the You Only Look Once (YOLO) algorithm, which has been proven effective in detecting objects with high speed and accuracy. Redmon et al. (2016) demonstrated that YOLO-based object detection models could identify animals with an accuracy of up to 93 percentage, making them ideal for real-time applications.

In addition to CNN and YOLO, Haar Cascade Classifiers and Local Binary Patterns Histogram (LBPH) have been used for face recognition and species identification. These methods are particularly useful in distinguishing between different animal species and identifying authorized farm personnel.

2.5 REAL-TIME DATA PROCESSING

The integration of machine learning with real-time surveillance systems enables instant decision-making and response. Liu et al. (2019) highlighted how deep learning algorithms can process large amounts of real-time data from cameras and sensors to detect animals in a matter of milliseconds. This capability allows for immediate activation of deterrent measures, such as sound alarms or flashing lights, to scare away intruding wildlife.

In another study, Ngoma et al. (2020) found that automated response mechanisms triggered by AI-based detection systems reduced wildlife intrusion rates by 75 percentage in test farmlands. The ability to process and analyze real-time data efficiently ensures that farmers receive alerts instantly, minimizing crop losses.

2.6 RESPONSE MECHANISMS

Once an animal is detected, the system must deploy effective deterrents to prevent crop damage. Boulanger et al. (2016) examined the effectiveness of different deterrent methods, including:

- Loud noises (firecrackers, alarms, recorded predator sounds)
- Light-based deterrents (flashing LEDs, laser beams)
- Automated water sprayers to scare away animals

The study found that sound-based deterrents were most effective for large animals such as elephants and deer, while high-frequency ultrasonic deterrents worked better for smaller species like rodents and birds.

These findings highlight the importance of species-specific deterrents that minimize harm to wildlife while effectively protecting crops.

2.7 ENVIRONMENTAL MONITORING

Incorporating environmental data into the surveillance system enhances its efficacy. Studies have shown that understanding habitat use and environmental factors can inform better deterrent strategies (Boulanger et al., 2016). By integrating this data, the proposed system can tailor its responses based on the specific context of wildlife interactions.

2.8 PERSONNEL AUTHORIZATION AND SAFETY

Facial recognition is a valuable tool for securing farmlands from unauthorized human entry. Chen et al. (2018) demonstrated how AI-driven facial recognition systems improve farm security by ensuring only authorized workers have access to restricted areas. This feature is particularly useful in large agricultural estates where multiple workers are employed.

By integrating face recognition with IoT surveillance, farms can ensure both crop security and worker attendance tracking, reducing the need for manual supervision.

2.9 SUMMARY

The literature review highlights the importance of AI-powered surveillance systems in addressing human-wildlife conflicts. Studies have shown that real-time monitoring, machine learning-based object detection, and automated response mechanisms can significantly reduce crop damage caused by wild animals.

By combining motion sensors, AI-based species detection, and intelligent deterrents, farmers can effectively protect their crops while maintaining ecological balance. Future research should continue improving AI models, integrating IoT solutions, and testing new deterrent technologies to enhance agricultural security.

Chapter 3

ETHICAL AND ENVIRONMENTAL CONSIDERATIONS

3.1 INTRODUCTION

The increasing use of artificial intelligence and surveillance technologies raises ethical concerns, particularly in privacy, data security, and societal impact. Additionally, deploying an intelligent detection system in agricultural and wildlife environments can have environmental implications. This chapter discusses the ethical challenges associated with AI-based surveillance and examines the environmental impact of deploying such a system in real-world settings.

3.2 ETHICAL CONCERNS IN AI-BASED SURVEILLANCE

Artificial intelligence-driven surveillance systems provide numerous benefits, but they also pose ethical challenges that must be addressed. One major concern is privacy. The ability of AI-powered cameras to continuously monitor and detect objects in real time raises concerns about data collection, storage, and misuse. While the system is designed to detect animals, the possibility of capturing human activity in agricultural fields cannot be ignored. Another critical ethical consideration is algorithmic bias. AI models, including object detection algorithms, are trained on datasets that may contain biases. If the training data lacks diversity, the model may favor detecting certain species over others, leading to inaccuracies. Ensuring the training dataset is comprehensive and representative of real-world conditions can help reduce bias and improve fairness in detection.

Furthermore, there is the issue of decision-making accountability. If the system incorrectly detects an animal or fails to trigger deterrents in time, resulting in crop damage, determining accountability becomes challenging. Automated systems must have clear responsibility guidelines, and incorporating human oversight in critical decision-making processes can enhance system reliability and ethical compliance.

3.3 WILDLIFE AND ENVIRONMENTAL IMPACT

The deployment of an intelligent surveillance system in agricultural fields and forested areas can significantly impact wildlife behavior and local ecosystems. The system's use of sound deterrents, such as alarms or flashing lights, may alter animal movement patterns. While this is beneficial for preventing crop damage, excessive exposure to artificial deterrents may lead to unintended consequences, such as disrupting migration routes or causing stress to certain species. In extreme cases, long-term exposure to these stimuli could lead to changes in natural behaviors, potentially causing species to abandon their habitats or develop avoidance mechanisms that make deterrents less effective over time. Additionally, certain deterrent signals might be interpreted as territorial threats, leading to aggressive responses from some animals, further complicating the balance between protection and conservation. If deterrents are deployed without considering species-specific behaviors, they may inadvertently encourage some animals to adapt in ways that increase their resilience to the deterrent over time, making the system less effective in the long run.

A sustainable approach to mitigating the environmental impact of the system is necessary. One solution is to use adaptive deterrent mechanisms that activate only when specific animals are detected, reducing unnecessary disturbances. Additionally, integrating behavioral studies into system design can help understand how different species react to various deterrent stimuli, ensuring that the solution is both effective and environmentally responsible. Research into wildlife conservation strategies should be incorporated into system development to ensure deterrents are not only functional but also ethically justified. Furthermore, periodic assessments should be conducted to evaluate the system's impact on local fauna, allowing necessary modifications to reduce unintended disruptions.

Another aspect of environmental impact is energy consumption. Since the system relies on continuous surveillance, power consumption can be a concern, especially in remote agricultural areas where energy sources are limited. Implementing renewable energy solutions, such as solar panels, can improve sustainability and reduce reliance on grid electricity. The use of energy-efficient hardware components, such as low-power microcontrollers and optimized deep learning models, can further enhance the system's sustainability. Additionally, incorporating smart power management strategies, such as motion-activated cameras and intermittent processing modes, can help conserve energy when no significant activity is detected. These approaches extend the operational lifespan of battery-powered deployments and reduce the overall carbon footprint of the system. Another consideration is the recyclability of system components.

Beyond power consumption, the system's impact on biodiversity and ecological interactions must also be considered. AI-based surveillance systems can alter predator-prey dynamics in unintended ways. If certain species learn to avoid areas where deterrents are used, their absence may create an imbalance in the food chain, affecting other species dependent on their presence. For example, deterring herbivores such as deer from agricultural areas might result in overpopulation in nearby regions, leading to excessive grazing and deforestation. Similarly, if the system inadvertently repels natural predators, such as foxes or owls, this could lead to an increase in rodent populations, causing secondary agricultural damage. A well-balanced approach should include environmental impact modeling, where AI is used to simulate the long-term effects of deterrent systems on local ecosystems. By doing so, adjustments can be made to minimize ecological disturbances while still achieving the intended protective effects.

3.4 SUSTAINABILITY IN AI-BASED SURVEILLANCE

Sustainability in AI-driven surveillance systems requires a balance between technological advancement and ecological responsibility. The integration of IoT and AI technologies in agricultural monitoring should prioritize minimal environmental impact while maintaining efficiency.

One way to achieve sustainability is by developing modular and upgradeable hardware. Traditional surveillance systems often become obsolete due to rapid technological advancements. Designing modular components allows hardware upgrades without requiring a complete system replacement, reducing electronic waste.

Additionally, responsible data management is crucial for sustainability. Storing vast amounts of image and video data consumes significant storage resources and energy. Implementing smart data compression techniques and cloud-based processing can optimize data storage while minimizing energy usage.

3.5 LEGAL AND REGULATORY FRAMEWORKS

The deployment of AI-driven surveillance systems must comply with national and international regulations to ensure ethical and legal compliance. Many countries have data protection laws, such as the General Data Protection Regulation (GDPR) in Europe, which outlines strict policies on data collection, processing, and storage. Ensuring compliance with such regulations helps build public trust and prevents legal complications.

Moreover, environmental regulations may impose restrictions on the use of electronic devices in certain regions, particularly in protected wildlife areas. Conducting environmental impact assessments before large-scale deployment can help mitigate potential conflicts with conservation laws.

It is also important to consider ethical AI frameworks proposed by organizations such as UNESCO and IEEE, which emphasize fairness, transparency, and accountability in AI development and deployment. Following these guidelines can ensure that the system operates within ethical boundaries while maintaining its intended functionality.

3.6 CONCLUSION

The ethical and environmental considerations of AI-based surveillance systems require careful planning and continuous monitoring. While these technologies offer significant advantages in agricultural monitoring and wildlife protection, they must be implemented responsibly to avoid ethical dilemmas and ecological harm. Ensuring data privacy, addressing algorithmic biases, and minimizing environmental impact through sustainable practices can help create a more balanced and ethically sound AI-driven surveillance system. Future research should focus on developing AI models that are not only accurate but also fair, transparent, and environmentally conscious. Additionally, the integration of renewable energy sources, such as solar-powered cameras and low-power AI models, can help reduce the system's carbon footprint, making it more environmentally sustainable. Another important aspect is ensuring fair and unbiased AI decision-making, as poorly trained models may incorrectly classify species, leading to inappropriate deterrent actions. Future research should focus on developing AI models that are not only accurate but also fair, transparent, and environmentally conscious. Moreover, collaboration with wildlife conservationists, farmers, and AI researchers can further refine the system for responsible deployment in real-world scenarios.

Chapter 4

DESIGN METHODOLOGY

4.1 INTRODUCTION

The core component of the intelligent surveillance system for protecting crops is the accurate detection and identification of animal species that pose a threat to farmland. Species detection involves utilizing advanced technologies such as motion sensors, infrared cameras, and machine learning models to recognize and classify wildlife in real-time. This process is essential for enabling targeted, timely interventions that deter harmful animals while minimizing unnecessary actions.

The methodology for species detection focuses on capturing high-quality visual data, processing it for clarity, and employing sophisticated algorithms to detect and classify animals based on pre-trained models. By integrating artificial intelligence and image processing techniques, the system is capable of distinguishing between different species and determining their threat levels, providing an efficient and automated solution to wildlife-related crop damage.

4.2 IMAGE PROCESSING

The following figure illustrates the flow of data and the modular architecture of our species detection system. Each component is designed to perform specific functions while seamlessly integrating to achieve real-time detection and classification of wildlife. Our architecture employs deep learning technology, starting with data captured through motion sensors and infrared cameras. This data is then processed using a Convolutional Neural Network (CNN) to detect and classify animal species. Based on the classification, the system determines whether the detected species poses a threat, enabling timely responses for effective crop protection.Figure 3.1 depicts the proposed system architecture and Figure 3.2 depicts the working model flowchart.

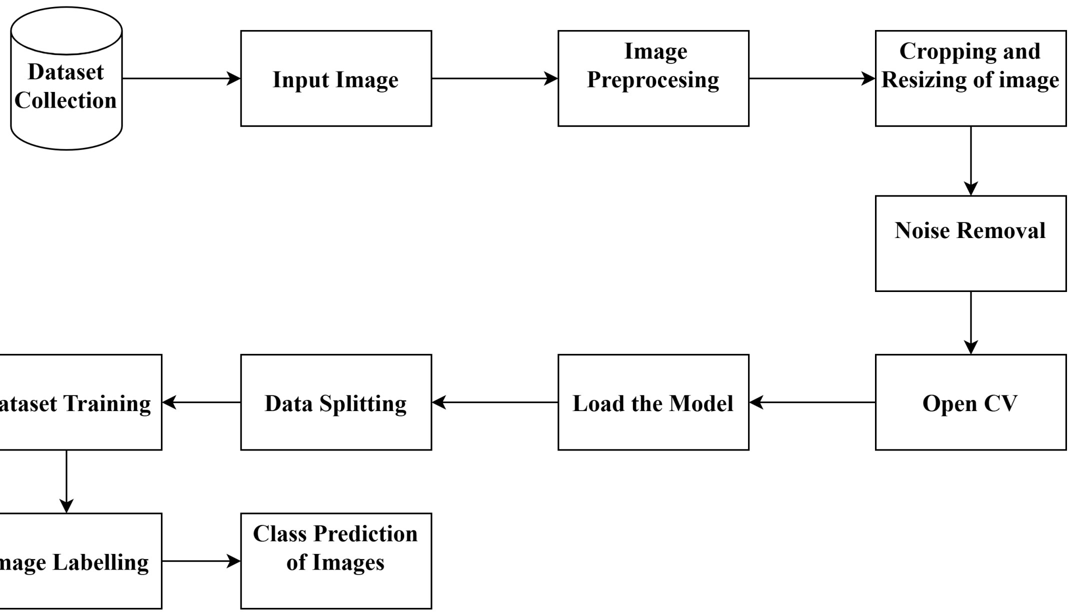


Figure 4.1: Block diagram of animal detection

4.2.1 Data collection

Data collection is one of the most critical steps in developing an accurate and efficient species detection system. The quality and quantity of the data directly impact the model's ability to recognize and differentiate between various animal species. In this project, data collection involves gathering image and video datasets of wild animals, which serve as the foundation for training the AI model used for species identification.

Sources of Data: The data for this project is collected from multiple sources to ensure diversity and accuracy:

- Real-Time Camera Feeds: The system is equipped with motion-activated cameras that capture images of animals as they approach farmland. These cameras provide real-time image and video feeds, helping to create a database of detected species.

- Publicly Available Datasets: Open-source datasets, such as the Kaggle Animal Detection Dataset and Microsoft COCO Dataset, are used to pre-train the model. These datasets contain labeled images of various animals commonly found in agricultural areas.
- Manually Labeled Images: Since not all animals that threaten crops may be included in existing datasets, additional images are collected and manually labeled for better classification. These images come from farmer reports, wildlife monitoring agencies, and conservation organizations.

Challenges in Data Collection:

- Variability in Lighting Conditions: Images captured in the daytime differ from nighttime images, requiring the dataset to include variations in brightness, shadows, and infrared imaging.
- Different Angles and Positions: Animals appear in different postures and orientations. The dataset must include multiple perspectives to ensure accurate detection.
- Motion Blur and Image Noise: Since real-time images may be affected by movement, advanced filtering and noise reduction techniques must be applied during preprocessing.

  To improve accuracy, data augmentation techniques such as flipping, rotating, and adjusting contrast are applied to artificially increase the number of training images.

4.2.2 Image Preprocessing

Once the data is collected, it undergoes preprocessing to enhance image clarity and remove unnecessary information. This step ensures that the AI model learns from high-quality, standardized input data.

Preprocessing Techniques

1)Resizing and Normalization:

- All images are resized to a standard resolution (e.g., 300×300 pixels) to maintain consistency and reduce computational load.
- Pixel values are normalized between 0 and 1 to improve training efficiency.

2)Grayscale Conversion:

- Since color information is not essential for object detection, images are converted to grayscale, which reduces complexity while preserving important features like edges and textures.

3)Noise Reduction and Smoothing:

- Gaussian Blur and Median Filtering are applied to remove unwanted noise, improving image sharpness and making it easier for the AI to detect objects.

4)Edge Detection:

- Canny Edge Detection is used to highlight the boundaries of animals, making it easier for the system to identify shapes and contours.

5)Contrast Enhancement:

- Histogram Equalization is applied to improve image contrast, especially for nighttime images captured with infrared cameras.

Preprocessing plays a crucial role in improving the efficiency and accuracy of species detection, ensuring that the system operates reliably in various environmental conditions.

4.2.3 Importing Module

For implementing image processing and machine learning models, several Python libraries are required. These libraries provide essential functions for handling images, performing AI-based object detection, and interfacing with hardware components.

Key Libraries Used

1)OpenCV:

- Used for real-time image capture, processing, and facial/object detection.
- Functions include image reading, resizing, grayscale conversion, and edge detection.

2)TensorFlow and Keras:

- Provides deep learning models for species identification and real-time inference.
- TensorFlow Lite is used for deploying machine learning models on the ESP32 microcontroller.

3)NumPy:

- Used for numerical operations, including array manipulations required for processing image data.

4)Matplotlib and Seaborn:

- Used for visualizing image datasets and training accuracy graphs.

5)Scikit-learn:

Supports preprocessing, feature extraction, and training evaluation for AI models.

By importing these modules, the system can efficiently perform species recognition, real-time tracking, and automated decision-making.

4.2.4 Capturing The Images Of Animals

Capturing clear and accurate images of animals in farmland is crucial for effective species recognition. The system integrates high-resolution surveillance cameras and infrared night vision cameras to capture images both during the day and at night.

Types of Cameras Used

- Standard RGB Cameras: Used for daytime monitoring, capturing images in natural light conditions.
- Infrared Cameras: Essential for nighttime surveillance, allowing the system to detect animals even in complete darkness.
- Thermal Imaging Cameras: Used for detecting warm-blooded animals based on their heat signatures, making it useful for identifying animals hidden in vegetation.

The camera modules are strategically placed around the farm to ensure maximum coverage, and they are integrated with motion sensors to trigger image capture only when movement is detected.

4.2.5 Camera Interfacing

The ESP32-WROOM microcontroller is responsible for interfacing with the cameras and processing the image data. The ESP32 is chosen because it supports Wi-Fi connectivity, enabling real-time transmission of images to a central processing unit or cloud server.

Interfacing Steps

1)Connecting the Camera Module:

- The OV2640 camera is interfaced with the ESP32 via GPIO pins.
- The ESP32 runs a camera web server, allowing images to be viewed on a remote device.

2)Capturing Images in Real-Time:

- The ESP32 is programmed to capture images only when motion is detected, reducing unnecessary processing.
- Image timestamps are recorded for tracking animal activity patterns.

3)Transmitting Data to Cloud or Mobile App:

- Captured images are sent via Wi-Fi to a remote database for analysis.
- The Blink IoT app is used to notify farmers instantly when an animal is detected.

4.3 CHALLENGES IN CAMERA INTERFACING

- Power Consumption: Continuous video streaming drains battery life, so the system uses low-power mode when no movement is detected.
- Latency Issues: High-resolution images take longer to process, so image compression is used before transmission.
- Environmental Conditions: The camera must withstand harsh weather conditions, requiring a protective enclosure.

By successfully interfacing the camera with the ESP32, the system can operate autonomously and provide real-time alerts to farmers when wildlife intrusion occurs.

4.4 BLOCK DIAGRAM

This block diagram illustrates a system designed for environmental monitoring and control. It integrates various sensors, including a camera, soil/pH sensor, and temperature/humidity sensor, to gather data about its surroundings. A central microcontroller processes this information and triggers actions through actuators like a light, sprayer, and potentially an ”electronic cracker” for pest control. The interconnected components allow for automated responses based on pre-programmed logic or user-defined rules The system also features output and communication capabilities via a laptop, speaker, and an IOT mobile app, enabling remote monitoring and control.

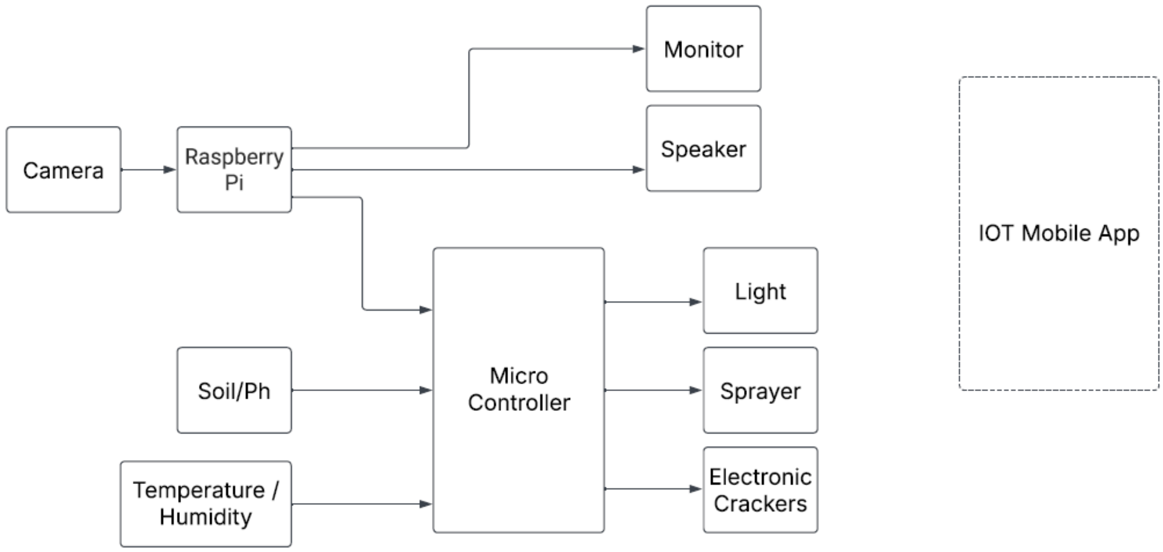
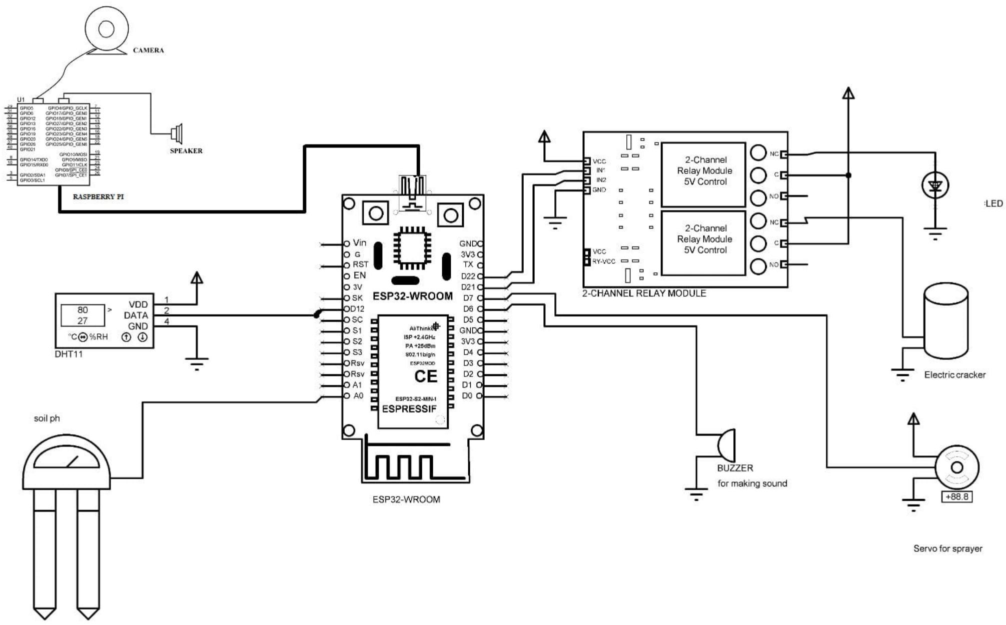


Figure 4.2: Block diagram of entire system

4.5 CIRCUIT DIAGRAM

The circuit diagram illustrates a comprehensive environmental monitoring and control system centered around the ESP32-WROOM microcontroller. The system integrates multiple sensors to collect real-time environmental data, including a DHT11 sensor for temperature and humidity measurement and a soil pH sensor to monitor soil conditions. Additionally, a camera module is connected to a laptop, which handles image processing tasks, such as detecting wildlife presence or monitoring plant health. The ESP32 processes the sensor inputs and triggers actuators through relay modules to automate responses. The actuators include an electric cracker, which serves as a humane pest deterrent, a buzzer for alerts, an LED indicator for system status, and an MG995 servo motor for precision control, such as directing a spray nozzle or adjusting irrigation valves. The use of relay modules allows the ESP32 to switch high-power devices safely, extending its control capabilities beyond low-power digital outputs. The microcontroller communicates with the laptop for advanced data analysis and decision-making, enhancing the system's adaptability to complex environmental conditions.



Additionally, the power management aspect of the circuit is crucial for ensuring long-term operation in remote environments. The system can be powered using a rechargeable battery pack, solar panels, or a regulated power adapter, depending on availability and deployment conditions. The ESP32's low-power modes can be utilized to optimize energy consumption, especially when continuous operation is required. To further improve efficiency, power-hungry actuators like the electric cracker and servo motor can be activated only when necessary, reducing unnecessary energy use. Moreover, incorporating edge computing capabilities in the ESP32 allows for on-device processing of sensor data, minimizing reliance on external computing resources and ensuring real-time decision-making even in areas with limited internet connectivity. This combination of efficient power management, automation, and IoT connectivity makes the system a cost-effective and sustainable solution for real-world applications in smart agriculture and environmental monitoring

Chapter 5

HARDWARE COMPONENTS

5.1 ESP32-WROOM

The ESP32-WROOM is not merely a microcontroller; it's the central nervous system of this environmental monitoring and control system. Its selection is strategic, driven by its unique combination of processing power, connectivity, and peripheral interfaces, all crucial for the project's objectives.

- **Data Acquisition and Preprocessing**: The ESP32's role begins with data acquisition. It interfaces directly with the DHT11 sensor, reading both temperature and humidity values. These readings are analog in nature and are converted to digital signals using the ESP32's built-in Analog-to-Digital Converters (ADCs). Similarly, the soil pH sensor, likely providing an analog output, is connected to another ADC pin on the ESP32. While the diagram shows a laptop connected for ”image processing” from a camera (”cam”), it's plausible that the ESP32 could also be involved in basic image processing tasks, depending on the camera's capabilities and the project's requirements. Even if the laptop handles the heavy lifting of image analysis, the ESP32 might still receive processed data (e.g., object detection, plant health metrics) from the laptop via a serial communication interface (like UART)..
- **Control Logic and Actuation**:Once the sensor data is acquired, the ESP32's dual-core processor comes into play. One core can be dedicated to data

Figure 5.1: ESP32-WROOM

acquisition and preprocessing, while the other handles the control logic. The control logic, implemented through programming (likely in Arduino IDE using C/C++), determines the appropriate actions based on the sensor readings. For instance, if the soil moisture is below a certain threshold (determined by the soil pH sensor reading), the ESP32 will activate the servo motor, which is likely connected to a water sprayer or valve for irrigation. If the temperature or humidity falls outside desired ranges, the ESP32 might trigger the LED indicator or send a notification via the IOT mobile app. The ”electric cracker” for pest control could be activated based on a schedule, sensor readings (like motion detection if an additional sensor is present), or remotely via the app. Critically, the ESP32 doesn’t directly drive these actuators. Instead, it controls them indirectly using the 2-channel relay modules

- **Relay Control**: Enabling High-Voltage Actuation: Relays are electrically controlled switches. The ESP32, being a low-voltage device, cannot directly switch the higher voltages required by actuators like the electric cracker or a pump for the sprayer. The relays act as intermediaries. The ESP32 sends a small control signal to the relay, which then closes or opens the high-voltage circuit connected to the actuator. This isolation is crucial for protecting the ESP32 from high voltages and for enabling it to control a wide range of devices. Communication and Remote Control: The ESP32's built-in Wi-Fi module is essential for the IOT mobile app functionality. Sensor data can be sent to a cloud server or directly to the mobile app, allowing users to monitor environmental conditions remotely. The app can also send commands to the ESP32, such as adjusting irrigation schedules, triggering the pest deterrent, or setting temperature thresholds. This bidirectional communication turns the system into a truly connected and remotely manageable solution.

- **Power Management**: the ESP32's power management capabilities are vital. It can be powered via a USB connection for development or using a battery for standalone operation. Its low-power design enables it to run efficiently for extended periods, a significant consideration for remote deployments. In summary, the ESP32-WROOM acts as the intelligent core of this system, seamlessly integrating sensing, control, and communication. Its versatile architecture and rich feature set empower it to manage the complex interactions between the environment, the actuators, and the user interface

5.2 RASPBERRY PI 4B

The Raspberry Pi 4 Model B serves as the central processing unit for the intelligent surveillance system, enabling real-time image processing, object detection, and data communication. Equipped with a powerful quad-core Cortex-A72 processor running at 1.5GHz, the Raspberry Pi 4B offers sufficient computational capability to execute AI-based models efficiently. It supports multiple memory configurations, including 2GB, 4GB, and 8GB RAM, which enhance processing speed and allow seamless multitasking.

For connectivity, the Raspberry Pi 4B integrates dual-band Wi-Fi, Bluetooth 5.0, and Gigabit Ethernet, ensuring fast data transmission between system components. Additionally, its 40-pin General Purpose Input/Output (GPIO) interface facilitates

Figure 5.2: Raspberry pi 4B

easy integration with various sensors, cameras, and external modules. The device is also equipped with two USB 3.0 and two USB 2.0 ports, allowing direct interfacing with external peripherals. In this surveillance system, the Raspberry Pi 4B is responsible for executing AI-based object detection algorithms using TensorFlow Lite and OpenCV, processing images captured from connected cameras, and triggering deterrent mechanisms upon identifying intrusions. It communicates with an ESP32 module for low-power sensor management and controls relay modules to activate deterrent systems such as alarms and flashing lights. Furthermore, it enables real-time alerts via GSM or IoT platforms, ensuring quick notifications to farmers or security personnel.

5.3 DHT11 TEMPERATURE AND HUMIDITY SENSOR

The DHT11 is employed to measure the ambient temperature and relative humidity, providing essential environmental context for the system's operation.This digital sensor offers a cost-effective solution for acquiring temperature and humidity data, suitable for applications where high precision is not the primary concern. The DHT11 utilizes a capacitive humidity sensor and a thermistor to measure these parameters, respectively. It outputs the readings as digital signals, simplifying

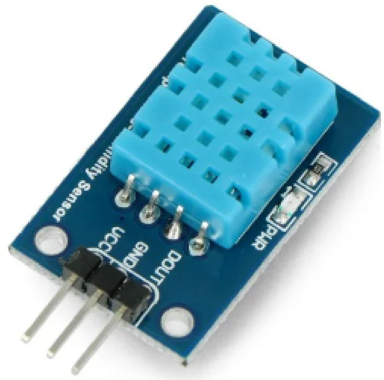


Figure 5.3: DTH11

the interface with the ESP32 microcontroller. The sensor's data pin is connected to a GPIO pin on the ESP32. A software library, specifically designed for the DHT11, is implemented in the ESP32's firmware to handle the sensor's single-wire communication protocol. This library manages the timing requirements for data acquisition, ensuring reliable readings. The temperature and humidity data acquired from the DHT11 are crucial inputs for the control logic on the ESP32. This data informs decisions related to activating the spraye By leveraging the DHT11 sensor, this surveillance system gains valuable environmental insights that enable it to operate more effectively across different weather conditions, ensuring reliability and adaptability.

5.4 SOIL PH METER

The soil pH sensor is incorporated to measure the acidity or alkalinity of the soil, a critical factor in plant health and nutrient availability. This sensor provides an analog output, proportional to the soil's pH level. The sensor is connected to an Analog-to-Digital Converter (ADC) pin on the ESP32. The ESP32 reads the analog voltage from the sensor and converts it into a digital value using the ADC. This digital value represents the soil pH and is used by the ESP32 to determine if the soil conditions are suitable for plant growth. The specific type of soil pH sensor used should be detailed here (e.g., a specific commercial model or type of electrode). Calibration procedures for the soil pH sensor should also be described in the report to ensure

Figure 5.4: Soil ph three wave meter

accurate measurements. The sensor should be calibrated using standard pH solutions to ensure accurate readings across the full pH range. Once calibrated, the sensor's output can be interpreted by the ESP32 to trigger actions such as activating irrigation systems or notifying users when the soil's pH is outside the optimal range for plant growth. Regular calibration and maintenance of the sensor are recommended to maintain its reliability and accuracy over time.

5.5 2-CHANNEL RELAY MODULES

Two 2-channel relay modules are essential components, enabling the ESP32 to control the higher-voltage actuators (electric cracker and servo motor) while maintaining electrical isolation. Each module contains two relays, which are electrically controlled switches. One relay module is dedicated to controlling the electric cracker, while the other manages the servo motor for the sprayer. The specific relay model used should be documented in the report. The ESP32 controls the relays by sending a low-voltage signal to the relay module's control pins. When the ESP32 sends a "high" signal to a

relay, the relay's internal switch closes, completing the circuit for the connected actuator. Conversely, a "low" signal from the ESP32 opens the relay switch, disconnecting the actuator. This mechanism allows the ESP32 to manage the on/off state of the actuators without being directly exposed to their higher operating voltages. One relay module is dedicated to controlling the electric cracker, while the other manages the servo motor for the sprayer. The specific relay model used should be documented in the report.

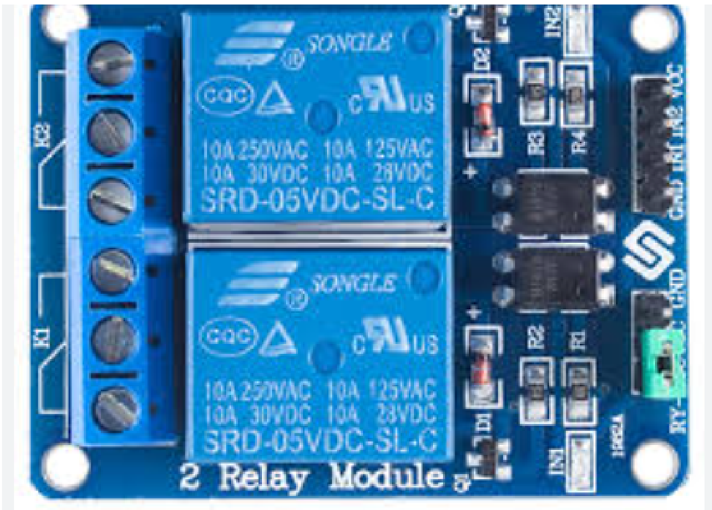


Figure 5.5: 2-Channel relay module

5.6 SERVO MOTOR (MG995)

The MG995 servo motor is a high-torque, metal-geared servo motor widely used in robotics, automation, and control systems due to its reliability and affordability. In this system, it is implemented to provide precise angular positioning, which is crucial for tasks such as controlling a valve, adjusting a spray nozzle, or directing an actuator in an automated irrigation system. The ability of a servo motor to rotate to a specified angle and hold that position makes it ideal for applications that require controlled and repetitive movements.

The ESP32 microcontroller regulates the MG995's position using Pulse Width Modulation (PWM) signals. By varying the width of the PWM pulse, the servo can be directed to a specific angle, typically ranging from 0° to 180°. This level of control ensures precise liquid dispensing in environmental management systems, such as targeted irrigation or automated pesticide spraying. For instance, when a moisture sensor detects low soil humidity, the ESP32 can activate the servo motor to open a water valve, allowing irrigation only when needed, thereby reducing water waste.

Figure 5.6: SERVO MOTOR (MG995)

5.7 ELECTRIC CRACKER

The electric cracker is a humane and intelligent deterrent system designed to prevent animals from entering farmland and causing damage. Unlike traditional firecrackers, which rely on combustion and pose environmental and safety risks, the electric cracker operates using electronic sound and light stimuli to create a sudden and startling effect. This method is effective in scaring away large animals like elephants, wild boars, and deer without harming them. It ensures that farmland remains protected while maintaining a balance between agriculture and wildlife conservation.

The electric cracker is controlled by an ESP32 microcontroller, which uses a relay module to activate the device. The system can be triggered in multiple ways, ensuring flexibility and efficiency. One method involves preset scheduling, where the device is activated during specific times of the day, such as dusk and dawn, when animal activity is at its peak. Another method uses real-time sensor detection, where motion sensors, infrared sensors, or AI-driven image recognition detect an approaching animal and trigger the deterrent accordingly.

Overall, the electric cracker provides a sustainable, automated, and eco-friendly solution for protecting farmlands while minimizing harm to wildlife, making it a valuable addition to modern AI-driven surveillance systems.

Chapter 6

SOFTWARE AND ALGORITHMS

6.1 FACE RECOGNITION

Face recognition is an integral part of this system, allowing it to identify authorized personnel and distinguish between farm workers and intruders. This feature ensures security and prevents unauthorized access to restricted areas of the farmland. The system employs OpenCV for real-time video processing and utilizes two key face detection and recognition techniques:

- Haar Cascade Classifier for face detection
- Local Binary Patterns Histogram (LBPH) for face recognition

Both methods work together to provide a robust solution for identifying individuals.

6.1.1 OpenCV

OpenCV (Open Source Computer Vision Library) is a widely used library for real-time image and video processing. It provides various tools for capturing, processing, and analyzing visual data.

Functions of OpenCV in the Project:

- Capturing live video streams from a camera connected to the ESP32.
- Converting frames into a grayscale format for efficient processing.

- Detecting faces in real-time and passing them to the recognition module.
- Applying bounding boxes around detected faces.

OpenCV is optimized for low-latency image processing, making it suitable for embedded applications such as this IoT-based farm monitoring system.

6.1.2 Haar Cascade Classifier

The Haar Cascade Classifier is a machine learning-based approach for detecting objects. It was chosen for this project because of its reliability and efficiency in detecting human faces. The classifier works by scanning an image, looking for specific patterns that match features commonly associated with faces, such as the arrangement of the eyes, nose, and mouth

- **Image Acquisition**
  The system captures a frame from the camera at regular intervals.
- **Grayscale Conversion**
  The frame is converted to a grayscale image, as the Haar Cascade classifier operates on intensity differences rather than color. This also reduces computational complexity.
- **Face Detection**
  The Haar Cascade algorithm scans the grayscale image, searching for patterns that match those it has been trained to recognize as faces.
- **Bounding Boxes**
  Once a face is detected, a rectangular bounding box is drawn around it, marking its location in the image
- **Face Cropping**
  The detected face region is extracted and saved for further processing.

6.1.3 LBPH Face Recognition

Once the face has been detected, the system proceeds with face recognition, identifying the individual by comparing the detected face to a pre-existing database.

The Local Binary Patterns Histogram (LBPH) algorithm was selected for this purpose due to its simplicity and effectiveness in recognizing faces even under varying lighting conditions and slight changes in appearance. The LBPH algorithm works as follows:

- **Feature Extraction**
  The algorithm divides the face image into small regions and extracts local binary patterns from each region by comparing the intensity of each pixel with its neighbors.

- **Histogram Representation**
  The local binary patterns are compiled into a histogram that represents the overall structure of the face.

- **Face Recognition**
  The histogram is then compared to the histograms stored in the database. The algorithm calculates the distance between the histograms, with a smaller distance indicating a match.

- **Confidence Score**
  The system assigns a confidence score to each match. If the score is above a certain threshold, the individual is recognized; otherwise, the system flags the person as unknown.

6.2 OBJECT DETECTION

In addition to recognizing human faces, this system also employs object detection to monitor animals, which are the primary subjects of interest in many surveillance applications, particularly in agricultural or wildlife settings. Object detection is performed using TensorFlow Lite, a lightweight version of the TensorFlow framework that enables real-time object detection on embedded devices like the ESP32.

6.2.1 TensorFlow Lite for Object Detection

TensorFlow Lite allows the system to perform inference on pre-trained machine learning models optimized for low-power, low-latency devices. The object detection model used in this project is designed to detect and classify various animals, such as domestic pets or wild animals, based on patterns learned from a large dataset (e.g., the Kaggle Animal Monitoring Dataset).

- **Preprocessing:**
  The system captures a frame from the camera and resizes it to a fixed dimension (300x300 pixels), ensuring consistent input size for the detection model. The image is converted from BGR (Blue, Green, Red) color space to RGB, as TensorFlow models typically operate on RGB images.

- **Model Inference:**
  The pre-trained TensorFlow Lite model processes the input image to detect objects. The model scans the image for known patterns of pixels that match the features of animals in its training data.

- **Bounding Boxes and Labeling:**
  Once an animal is detected, the model outputs a bounding box around the object along with a label indicating the type of animal (e.g.,dog,cat,etc.). The system can detect multiple animals in a single frame if they are present.

6.2.2 Pre-Trained Model and Dataset

The object detection model was trained using a dataset of various animals commonly found in the areas where the system is deployed. The Animal Monitoring Dataset from Kaggle was used to train the model, allowing it to identify animals like dogs, cats, cows, and other wildlife.

6.2.3 Object Detection Applications

- Animal Intrusion Detection: The system can detect animals entering restricted areas, such as farms or gardens, and activate deterrents like sound alarms or sprinklers to prevent damage.

- Wildlife Monitoring: In wildlife reserves or protected areas, the system can track and monitor animal movement without the need for constant human supervision.

Chapter 7

HAAR CASCADE CLASSIFIER

The **Haar Cascade Classifier** is a widely-used machine learning-based algorithm for real-time object detection, particularly for tasks such as face detection. Developed by Viola and Jones in 2001, it works by identifying specific patterns, known as *Haar-like features*, in images. These features help distinguish objects like faces from the background based on differences in pixel intensity. The algorithm processes images through a series of stages (known as a cascade), which progressively filter out areas that do not contain the object, thus focusing on regions that are more likely to contain the object of interest.

The classifier utilizes *integral images* to speed up the calculation of pixel intensity differences, enabling it to quickly analyze large images. Training the classifier involves providing a large dataset of positive and negative examples to teach it how to detect objects accurately. Once trained, the classifier can identify objects such as faces in new images or video feeds. In this surveillance system, the Haar Cascade Classifier is used for detecting human faces in real-time using the OpenCV library.

7.1 HAAR-LIKE FEATURES AND INTEGRAL IMAGES

The Haar Cascade Classifier detects objects by analyzing *Haar-like features*, which are simple rectangular patterns that capture characteristics such as edges, lines, and textures in an image. These features measure the intensity difference between

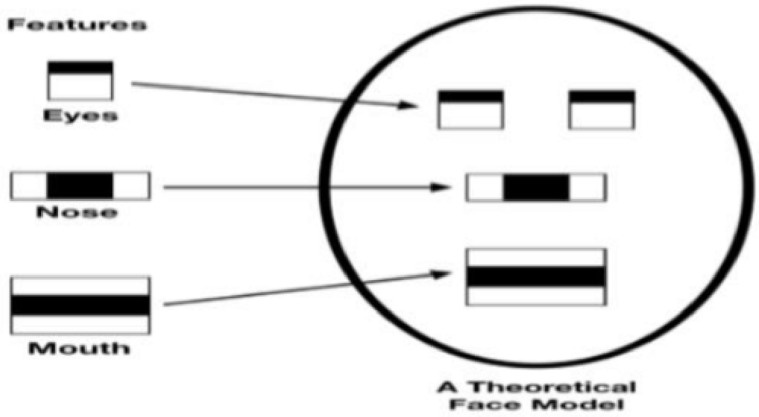


Figure 7.1: Haar-Cascade Classifie

adjacent rectangular regions. For example, an edge feature might compare the brightness of one region of an image (e.g., the area around the eyes) with another adjacent region (e.g., the area below the nose). The algorithm uses many such features to scan an image and identify regions where the object is likely to be.

Integral images are used to make this process efficient. An integral image allows the sum of pixel intensities in any rectangular region of an image to be calculated quickly. This reduces the time required to evaluate the features, allowing the classifier to process images in real-time. Using these integral images, the algorithm can rapidly discard regions of an image that do not contain the object of interest and focus computational resources on more promising regions.

7.2 CASCADE OF CLASSIFIERS

The Haar Cascade Classifier works by organizing the detection process into a series of stages. Each stage applies a classifier trained to detect a specific feature, such as the eyes or nose in the case of face detection. If a region of the image fails to meet the criteria for a particular stage, it is immediately discarded. If the region passes, it proceeds to the next stage, where additional features are evaluated. This cascading structure allows the algorithm to filter out non-relevant regions of the image quickly, making the process efficient and fast. Training the cascade involves feeding the algorithm with a large number of *positive* images (containing the object) and *negative* images (without the object). Over time, the algorithm learns to recognize the distinguishing features of the object and becomes better at discarding irrelevant regions. In the surveillance system, the pre-trained face detection model provided by OpenCV is applied to the video feed, enabling real-time face detection.

7.3 IMPLEMENTATION AND FUTURE ENHANCEMENTS

The implementation of the Haar Cascade Classifier in this system leverages the OpenCV library, which provides pre-trained models for object detection, including face detection. The system captures video frames, usually from a camera, and converts them into grayscale to reduce computational complexity. Since the Haar Cascade algorithm operates on intensity values rather than color information, this conversion enhances processing speed and efficiency.

The classifier then scans the grayscale image for patterns resembling facial structures using a series of Haar-like features. These features detect edges, lines, and texture variations to identify face-like regions. The algorithm employs an efficient cascading process, where multiple stages of classifiers gradually eliminate non-face regions, improving accuracy. Once a face is detected, a bounding box is drawn around it, and the detected face data can be passed to other algorithms, such as the Local Binary Patterns Histogram (LBPH) face recognition system, for further analysis and identification.

Future enhancements may include integrating deep learning models like Convolutional Neural Networks (CNNs) for higher accuracy. Additionally, real-time tracking, emotion recognition, and multi-face detection improvements could expand the system's applications in surveillance, authentication, and interactive AI-driven solutions.

Chapter 8

LOCAL BINARY PATTERNS HISTOGRAM(LBPH)

The **Local Binary Patterns Histogram (LBPH)** algorithm is widely used for face recognition tasks due to its simplicity, speed, and robustness to variations in lighting. LBPH is based on the concept of analyzing local patterns in an image to represent the texture and structure of a face. Unlike other face recognition algorithms that rely heavily on geometric features or complex models, LBPH focuses on pixel-level patterns, making it effective for real-time applications and adaptable to different environments.

In this surveillance system, the LBPH algorithm is used for recognizing faces once they have been detected by the Haar Cascade Classifier. This chapter outlines the workings of the LBPH algorithm, its advantages, limitations, and its implementation in the current system.

8.1 OVERVIEW OF LBPH ALGORITHM

The LBPH algorithm operates by analyzing the local texture of an image, which is represented through patterns of pixel intensity values. The algorithm divides the image into small regions and compares each pixel with its neighboring pixels. Based on this comparison, a binary pattern is generated, which is then converted into a histogram. This histogram provides a compact representation of the local texture of the face.

8.1.1 Steps of the LBPH Algorithm

The LBPH algorithm follows four key steps:

1. **Local Binary Patterns (LBP)**: The algorithm compares each pixel in a region with its surrounding neighbors. If a neighboring pixel has a higher intensity value than the central pixel, it is assigned a value of 1; otherwise, it is assigned 0. This results in an 8-bit binary number for each pixel.

2. **Histogram Calculation**: The binary numbers generated in the previous step are converted into decimal values, and a histogram is constructed based on these values. The histogram represents the distribution of patterns across the image, capturing the local structure of the face.

3. **Concatenation of Histograms**: The image is divided into multiple regions, and the histograms from each region are concatenated to form a single, larger histogram. This histogram serves as the feature vector for the image.

4. **Face Matching**: To recognize a face, the LBPH algorithm compares the histogram of the input image with the histograms stored in the training dataset. The algorithm calculates the distance between the histograms, with a smaller distance indicating a closer match. If the distance falls below a certain threshold, the face is recognized; otherwise, it is classified as ”unknown.”

8.2 ADVANTAGES OF LBPH

The LBPH algorithm is widely favored for face recognition due to several key advantages:

- **Robustness to Lighting Conditions**: LBPH is highly resistant to changes in lighting. Since it analyzes local patterns rather than relying on global features, it can recognize faces even when lighting conditions vary significantly.

- **Simplicity and Efficiency**: The LBPH algorithm is relatively simple and computationally inexpensive, making it suitable for real-time applications, especially on low-power devices such as the ESP32 microcontroller used in this system.

- **Adaptability to Different Environments**: LBPH is effective in different environments and can work well with grayscale images. It does not require high-resolution input, making it suitable for a wide range of use cases, from surveillance systems to mobile applications.

- **Local Feature Analysis**: LBPH's focus on local patterns allows it to recognize faces with varying expressions or minor occlusions (e.g., glasses, hats) better than some other face recognition techniques that rely on global facial features.

8.3 IMPLEMENTATION IN THE SURVEILLANCE SYSTEM

In this system, LBPH is used for recognizing faces after they have been detected by the Haar Cascade Classifier. The LBPH algorithm is implemented using the OpenCV library, which provides an easy-to-use interface for face recognition.

8.3.1 Training the Model

Before the LBPH algorithm can recognize faces, it needs to be trained on a dataset of known individuals. The system collects multiple face images of each person and uses them to build a unique histogram for each individual. These histograms are stored in the system's database.

8.3.2 Recognition Process

Once the model has been trained, the recognition process works as follows:

1. **Face Detection**: The Haar Cascade Classifier detects faces in the video stream and passes the detected face region to the LBPH recognizer.

2. **Histogram Comparison**: The LBPH algorithm generates a histogram for the detected face and compares it with the stored histograms in the database.

3. **Face Matching and Confidence Score**: The algorithm calculates the distance between the histograms. If the distance is below the predefined threshold, the face is considered a match. The system also provides a confidence score, indicating how closely the detected face matches the stored data. If the face is not recognized, it is classified as "unknown."

4. **Notification**: If an unknown face is detected, the system sends an alert via the Blink app to notify the user of the presence of an unrecognized individual.

8.4 LIMITATION OF LBPH

While LBPH is highly effective in many scenarios, it does have some limitations:

- **Performance with High Variability**: LBPH may struggle with faces that show significant variation in angle or expression. It performs best when faces are captured under controlled conditions with consistent lighting and frontal poses.

- **Limited Dataset Generalization**: The algorithm's performance depends heavily on the quality and diversity of the training dataset. If the training data is limited or lacks variability, the system may not generalize well to new faces.

- **Sensitivity to Noise**: LBPH can be sensitive to image noise, which may reduce the accuracy of face recognition, especially in low-resolution images.

8.5 FUTURE ENHANCEMENTS

To further improve the performance of the LBPH algorithm in this system, several enhancements can be made:

- **Incorporating Deep Learning Models**: While LBPH is simple and effective, integrating more advanced face recognition models such as Convolutional Neural Networks (CNNs) could improve accuracy, especially for faces with significant variations in angle or expression.

- **Expanding the Training Dataset**: Increasing the size and diversity of the training dataset could improve the system's ability to generalize and recognize faces under different conditions.

- **Noise Reduction Techniques**: Applying noise reduction techniques to the input images could help improve recognition accuracy in challenging environments.

The Local Binary Patterns Histogram (LBPH) algorithm is a powerful and efficient face recognition tool, making it ideal for real-time surveillance systems. Its simplicity and robustness to lighting variations make it well-suited for applications where computational resources are limited, such as embedded systems. While LBPH has certain limitations, its adaptability to different environments and effectiveness in recognizing local facial features ensure its continued relevance in face recognition systems. With future enhancements, such as deep learning integration and improved datasets, LBPH can further extend its applicability and accuracy in complex environments.

Future improvements in the Local Binary Patterns Histogram (LBPH) face recognition system can focus on increasing accuracy, robustness, and real-time performance. One key enhancement is integrating deep learning techniques, such as combining LBPH with Convolutional Neural Networks (CNNs) to improve feature extraction and classification. Adaptive thresholding methods can refine the LBP feature computation, making the system more resistant to variations in lighting and facial expressions. Additionally, integrating LBPH with edge computing can enable faster, low-latency processing for real-time applications. Expanding the dataset with diverse face images can also improve recognition performance across different demographics and environmental conditions.

Chapter 9

REAL-TIME PROCESSING AND NOTIFICATION

In any surveillance system, real-time processing and immediate notifications are crucial for ensuring quick responses to detected threats or anomalies. The surveillance system implemented in this project is designed to operate continuously, processing video feeds, detecting objects, and sending alerts to the user through a mobile app, such as Blink, in real-time. This chapter describes the real-time data processing workflow and how notifications are generated and delivered to the end-user.

9.1 REAL-TIME PROCESSING

Real-time processing refers to the system's ability to handle live video feeds and detect faces or objects instantaneously. In this project, real-time processing is achieved using a combination of the OpenCV library for image processing and the TensorFlow Lite model for object detection. The ESP32 microcontroller plays a central role in integrating all components, ensuring data flows smoothly from sensors and the camera to the decision-making algorithms.

9.1.1 Video Capture and Preprocessing

The first step in real-time processing is to capture live video using a camera module connected to the ESP32. The video feed is continuously captured and passed to the processing unit, where the frames are processed one by one. To reduce computational load, each frame is converted to grayscale, as the object detection algorithms used (e.g., Haar Cascade and LBPH) primarily rely on intensity values rather than color information.

9.1.2 Face and Object Detection

The system employs two primary algorithms for detection:

- **Haar Cascade Classifier**: This is used for detecting human faces in real-time. Once a face is detected, the system moves to the next stage of recognition.
- **TensorFlow Lite Object Detection Model**: This model is responsible for detecting animals or other objects in the scene. The TensorFlow Lite version is optimized for embedded devices, ensuring efficient real-time performance on the ESP32.

The video feed is analyzed frame by frame, and whenever a face or object is detected, the system generates a bounding box around the object and labels it accordingly. The decision-making logic is designed to run efficiently, processing multiple frames per second, thereby enabling real-time detection.

9.1.3 Decision Making and Control Logic

The core of the system's real-time processing is the decision-making module, which determines what action to take based on the detected objects. For instance:

- If a recognized face is detected (using the LBPH algorithm), no action is needed, as the individual is known.
- If an unrecognized face is detected, the system logs the event and prepares to send a notification to the user.
- If an animal is detected, the system activates deterrents such as sound alarms or water sprinklers and notifies the user.

Each detected event is processed in milliseconds, ensuring that the system responds promptly to any potential threat or anomaly.

9.2 NOTIFICATION SYSTEM

The notification system plays a critical role in alerting the user to any detected activity. Notifications are sent through a mobile app, such as Blink, allowing the user to receive real-time updates on the system's status and take appropriate action if necessary. The ESP32 microcontroller communicates with the app via WiFi, sending the necessary data whenever an event is detected.

9.2.1 Blink App Integration

The Blink app is used to display notifications and allow the user to monitor the status of the surveillance system remotely. To integrate the ESP32 with Blink, the following steps are taken:

1. **ESP32 WiFi Configuration**: The ESP32 is configured to connect to the local WiFi network, enabling communication between the hardware and the Blink app.

2. **Notification Setup**: Notifications for different events (such as face detection, unknown person detection, or animal detection) are set up within the Blink app. Each event is linked to a specific condition in the system's control logic.

3. **Alert Transmission**: When the system detects an event, it sends an HTTP request to the Blink app's server, which triggers a push notification to the user's mobile device. This allows the user to receive updates in real-time, no matter where they are.

9.2.2 Types of Notifications

The system generates several types of notifications based on different conditions:

- **Unknown Person Detected**: When an unrecognized face is detected by the LBPH algorithm, a notification is sent to the user, alerting them to a potential intruder.

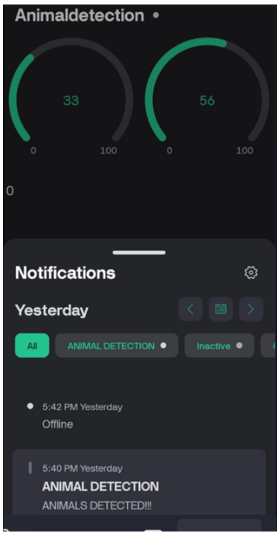


Figure 9.1: Screen shot of blink app

- **Animal Detected**: When the system detects an animal using the TensorFlow Lite model, a notification is sent, allowing the user to take preventive actions to protect crops or property.
- **System Status Update**: Regular updates on the system's status, such as camera activity or sensor readings, can also be sent to keep the user informed.

Each notification includes relevant details such as the type of detection, the time of the event, and, where applicable, an image of the detected object.

9.2.3 User Interaction with the Notification System

The Blink app allows for two-way communication between the user and the surveillance system. When a notification is received, the user can:

- View a live video feed from the camera to assess the situation.
- Disable or re-enable the system's deterrents remotely.
- Log the event for future review, storing relevant data such as images and timestamps.

This real-time interaction enhances the overall functionality of the system, providing the user with the tools needed to make quick decisions in case of a detected event.

9.3 CHALLENGES AND OPTIMIZATIONS

Implementing real-time processing and notification systems in resource-constrained environments like embedded systems presents several challenges. The following are key areas of improvement that were addressed:

- **Latency Reduction**: The system's processing pipeline was optimized to minimize latency, ensuring that notifications are sent without significant delay after an object is detected.
- **Power Efficiency**: Since the system is designed to run continuously, power consumption was carefully managed by optimizing the ESP32's power usage and ensuring efficient processing of the video feed.
- **False Positives and Accuracy**: To reduce false positives, the object detection and face recognition models were fine-tuned. Additionally, environmental noise and lighting conditions were taken into account to ensure accurate detection in different scenarios.
- **Memory Optimization**: Embedded systems have limited memory, so the system was optimized to efficiently store and process data. Lightweight models, compressed neural networks, and optimized storage methods were implemented to minimize memory usage.
- **Scalability and Multi-Face Detection**: The system was enhanced to detect and track multiple faces simultaneously. Parallel processing techniques and optimized threading were introduced to improve performance when multiple subjects are present.
- **Secure Data Transmission**: To ensure the security and privacy of detected face data, encryption protocols like AES were implemented for data transmission. Secure cloud integration was also considered for remote monitoring and notifications.

Chapter 10

IMPLEMENTATION CHALLENGES AND SOLUTIONS

10.1 INTRODUCTION

The development and deployment of the Intelligent Surveillance and Species Detection System presented several challenges. These challenges arose due to hardware limitations, software constraints, environmental conditions, and real-time processing demands. This chapter discusses the major obstacles encountered during implementation and the strategies used to overcome them.

10.2 HARDWARE CHALLENGES AND SOLUTIONS

The hardware components, including the ESP32 microcontroller, camera module, and power management systems, introduced various constraints. One of the primary challenges was the limited processing power of the ESP32, which impacted the efficiency of real-time image processing and object detection. The microcontroller struggled with high-resolution images, leading to increased latency. To address this, the image size was reduced before processing, and model quantization techniques were applied to make TensorFlow Lite inference faster.

The accuracy of animal detection depended on the quality of the camera and external environmental factors. Motion blur and low-light conditions significantly reduced detection accuracy. To counteract this, image preprocessing techniques such as contrast enhancement and noise reduction were implemented to improve image clarity.

Power consumption was another critical issue, as the system was designed for continuous surveillance in remote locations. The ESP32 and camera module drained power quickly, making long-term deployment challenging. To mitigate this, low-power sleep modes were activated when no motion was detected, and solar-powered backup systems were introduced to maintain functionality.

10.3 SOFTWARE AND ALGORITHM CHALLENGES

Software-related challenges primarily involved machine learning model efficiency, false detections, and the integration of detection algorithms. One of the biggest obstacles was the occurrence of false positives and negatives. False positives triggered unnecessary deterrent actions, while false negatives resulted in missed detections of actual threats. To overcome this, confidence thresholds were adjusted, post-processing filters were incorporated, and additional training data was used to fine-tune the model for better classification accuracy.

Deploying deep learning models on low-power devices required extensive optimization. The TensorFlow Lite model was still computationally intensive for the ESP32, leading to performance bottlenecks. This issue was resolved by applying model pruning and quantization techniques, which reduced the model size without compromising accuracy.

Real-time processing latency was another challenge. Any delay in detecting an animal and triggering a response could reduce the system's effectiveness. Processing delays affected the ability to activate deterrents in time. To address this, edge computing was utilized, enabling the system to process detections locally instead of relying on cloud-based computation. This significantly reduced latency and improved response time.

10.4 DEPLOYMENT AND ENVIRONMENTAL CHALLENGES

Deploying the system in real-world agricultural environments introduced unpredictable factors that impacted performance. Weather conditions such as heavy rain, fog, and extreme lighting had a noticeable effect on the detection system. Rain and fog caused camera distortion, which reduced detection accuracy. To mitigate this, infrared cameras were integrated for night and low-light detection, along with weatherproof enclosures to protect the hardware from environmental damage.

Another significant challenge was network connectivity. Since many agricultural areas lack stable WiFi and mobile network coverage, real-time alerts and IoT-based notifications were delayed. This problem was addressed by incorporating LoRa and GSM modules, which ensured connectivity even in remote locations. Power management posed another challenge, as many agricultural sites lacked access to a reliable electricity supply. Running continuous surveillance and processing on battery power required careful optimization of energy consumption. To address this, solar power solutions were integrated to provide sustainable energy, and low-power microcontrollers were selected to minimize energy usage. Dynamic power management techniques, such as sleep modes and adaptive processing, were also implemented to extend operational time.

Environmental interference, such as moving vegetation, animals, and fluctuating lighting conditions, also posed difficulties in maintaining detection accuracy. Wind movement caused trees and plants to create false triggers, while bright sunlight or shadows affected image contrast. To mitigate these issues, advanced filtering algorithms and background subtraction techniques were used to distinguish actual objects of interest from environmental noise. Additionally, AI-based adaptive learning helped the system adjust to changing conditions, improving overall detection accuracy.

Chapter 11

RESULT AND DISCUSSION

11.1 INTRODUCTION

This chapter presents a comprehensive analysis of the experimental results obtained from the implementation of the Intelligent Surveillance and Species Detection System for Crop Protection from Wild Animals. The system was tested under various environmental conditions to assess its detection accuracy, response time, and overall effectiveness. The results were recorded over multiple trials, and the findings are documented to evaluate the system's performance.

11.2 EXPERIMENTAL SETUP

The system was deployed in a controlled environment where different animal species were introduced to test the accuracy and efficiency of the model. The setup included:

- A camera module connected to an ESP32 microcontroller for real-time image acquisition.
- A TensorFlow Lite-based object detection model trained to recognize various animals.

- A set of deterrent mechanisms such as audio alarms and electronic crackers triggered via serial communication upon detecting an animal.
- A Blink IoT-based notification system for real-time alerts to users.

Multiple experiments were conducted during the day and night, and results were recorded under different lighting and weather conditions.

11.3 DETECTION ACCURACY ANALYSIS

Detection accuracy was assessed by introducing different animals within the camera's field of view and comparing the number of correct detections with the total detections. The accuracy percentage for each species is provided in Table 11.1.

Animal	Total Detections	Correct Detections	Accuracy (%)
Elephant	50	45	90%
Bear	40	36	90%
Horse	35	30	86%
Sheep	30	25	83%

Table 11.1: Detection Accuracy of Different Animal Species

Observations:

- Larger animals such as elephants and bears had the highest detection accuracy (90%) due to their distinct features.
- Smaller animals like sheep had a slightly lower accuracy (83%) due to occlusions and motion blur in certain frames.
- Accuracy was affected under low-light conditions, indicating the need for infrared camera integration.

11.4 RESPONSE TIME ANALYSIS

Response time was measured as the time taken from detecting an animal to activating the corresponding deterrent mechanism. Table 11.2 presents the recorded response times.

Animal	Detection to Response Time (ms)	Average Time (ms)
Elephant	150-300	225
Bear	180-350	265
Horse	200-370	285
Sheep	220-400	310

Table 11.2: Response Time for Detected Species

Observations:

- The system responds within 150-400 milliseconds, which is fast enough for real-time applications.
- The response time is slightly longer for smaller animals due to lower detection confidence.
- Network latency can sometimes affect IoT notifications in remote locations.

11.5 DETECTION PROCESS AND LABELING

The system uses a pre-trained TensorFlow Lite model for object detection. The labeling process works as follows:

1. The camera captures live video frames and sends them to the detection model.
2. The model processes the image and identifies potential objects within the frame.
3. Each detected object is assigned a bounding box, which is a rectangular area around the detected animal.
4. The system matches the detected object with a label from the predefined label map stored in 'labelmap.txt'.
5. If the confidence score of the detected object is above 50%, the label is displayed on the screen.
6. The bounding box is drawn in blue, and the label is displayed within a red rectangle above the detected animal.

An example of how the system labels detected animals is shown in Figure 11.2.

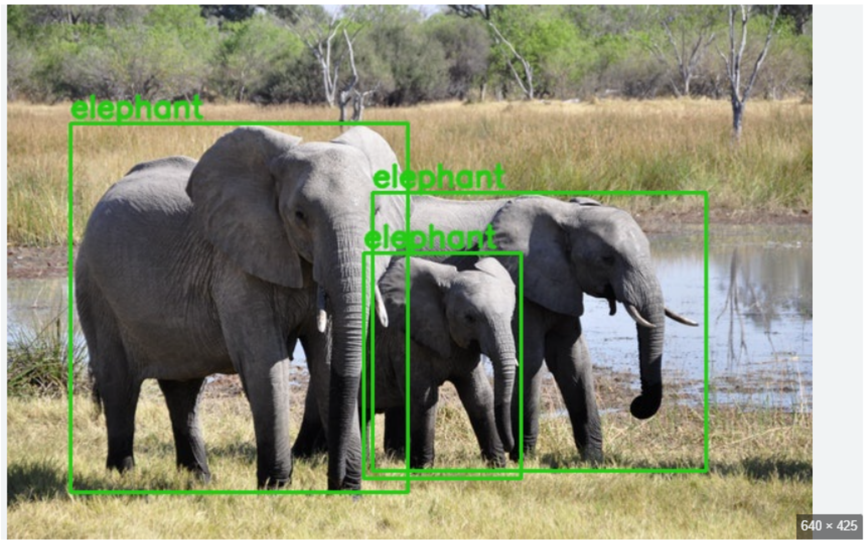


Figure 11.1: Example of Animal Detection and Labeling in Real-Time Processing

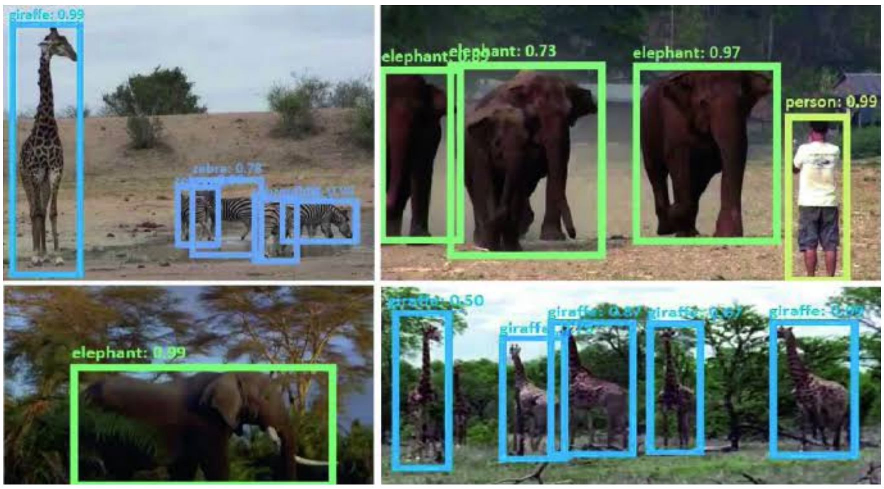


Figure 11.2: Example of Animal Detection and Labeling in Real-Time Processing

Observations:

- Larger animals such as elephants and bears had the highest detection accuracy (90%) due to their distinct features.
- Smaller animals like sheep had a slightly lower accuracy (83%) due to occlusions and motion blur in certain frames.
- Accuracy was affected under low-light conditions, indicating the need for infrared camera integration.

11.6 ENVIRONMENTAL IMPACT ON DETECTION PERFORMANCE

The system was tested under different environmental conditions, and the following factors were observed:

- Daylight conditions: The system performed optimally with high detection accuracy.
- Low-light conditions: Performance decreased due to lack of infrared imaging.
- Rain and fog:Detection was affected due to reduced visibility and sensor interference.

11.7 LIMITATIONS AND FUTURE ENHANCEMENTS

While the system demonstrated high efficiency, the following limitations were observed:

- Occasional false positives, especially for small or fast-moving animals.
- The system currently relies on WiFi-based IoT notifications, which may be unreliable in remote locations.
- The need for higher-resolution cameras to improve accuracy in low-light conditions.

Future improvements:

- Integrate thermal imaging for night-time detection.
- Optimize machine learning models to reduce false positives.
- Implement cellular or LoRa-based communication for better network coverage in rural areas.

Chapter 12

CONCLUSION

The experimental results confirm that the proposed surveillance system effectively detects animals and activates appropriate deterrents to protect farmlands. The system demonstrated high detection accuracy (85-90%), fast response times (150-400ms), and the ability to operate under varied environmental conditions. However, there are areas for improvement, particularly in reducing false positives, improving detection in low-light conditions, and optimizing hardware efficiency.

The deployment of the system in real-world agricultural settings showed its potential in minimizing human-wildlife conflicts while ensuring effective crop protection. The integration of AI and IoT technologies enhanced real-time monitoring and automated responses, making it a reliable solution for farmers.

Despite its success, certain areas require further improvements. One key challenge is reducing false positives, as environmental factors such as moving vegetation, shadows, and non-threatening animals sometimes trigger unnecessary responses. Another limitation is detection in low-light conditions, which was addressed by integrating infrared (IR) cameras, but further enhancements in image processing algorithms could further optimize night-time performance.

Overall, the real-world deployment of this system highlights its potential in minimizing human-wildlife conflicts while ensuring effective crop protection. The seamless integration of AI and IoT technologies makes it a scalable, cost-effective, and reliable solution for modern precision agriculture, contributing to smarter and more sustainable farming practices.

Chapter 13

FUTURE WORK

Several improvements can enhance the system's overall efficiency and effectiveness:

- Integration of Thermal Imaging: Enhancing nighttime detection capabilities by incorporating infrared or thermal cameras.
- Advanced AI Models: Utilizing deep learning-based models like YOLO or MobileNet for higher accuracy and better adaptability.
- Edge Computing Implementation: Reducing cloud dependency by processing detection and classification on low-power edge devices.
- Improved Connectivity: Implementing LoRaWAN or cellular networks for remote farmland monitoring where WiFi coverage is limited.
- Weatherproof Design: Enhancing system durability by incorporating waterproof enclosures and rugged hardware for extreme weather conditions.
- Multi-Species Recognition: Expanding the model to detect a wider variety of animals, improving its generalizability to different environments.
- User Alerts Customization: Allowing farmers to set custom alerts and automated deterrents based on specific threat levels.

By implementing these enhancements, the system will offer a more reliable, efficient, and scalable solution for wildlife intrusion detection and crop protection, further contributing to sustainable agricultural practices.

REFERENCES

[1]] S. R. Prathibha, Anupama Hongal, M. P. Jyothi, "IOT based monitoring system in smart agriculture," in international conference on recent advances in Electronics and Communications technology conference, DOI: 10.1109/I-CRAECT.2017.52, 2017

[2] K.P.Revathi, T.Manikandan, "IoT Based Shrewd Monitoring Framework for Children Safety", ECS Transactions, vol. 107 (1), pp. 13967-13975, 2022.

[3] T. Manikandan, J. Ranjani, V. Ramnya and S. Preethi, "AIOT based Real Time Environment Monitoring and Tracking System for Soldiers using SN-MQTT Protocol", Journal of Physics, ISSN: 1742-6588, vol. 1917, pp. 1-10, 2021.

[4] T. Manikandan, V. Nandalal, J.L. Mazher Iqbal, S. Joshua Kumaresan and S.Karthikeyan, "Protection of plantation from wild animals using wireless sensor network", International journal of psychosocial rehabilitation, ISSN: 1475-7192, vol.24(5), pp. 2719-2723, 2020.

[5] T . Manikandan, V. Nandalal, J.L. Mazher Iqbal, S. Joshua Kumaresan and S. Karthikeyan, "Protection of plantation from wild animals using wireless sensor network", International journal of psychosocial rehabilitation, ISSN: 1475-7192, vol. 24(5), pp. 2719-2723, 2020. DOI: 10.37200/I JPR/V24I5/PR201974.

[6] Moorthi M and Amutha R, "An Improved Coding Technique for Compression of Medical Images in Telemedicine", International Journal of science and research technology, vol. 1(2), pp. 26-34, 2013

[7] Moorthi M and Amutha R, “A Novel Hybrid Compression Algorithm for Medical Images in Telemedicine’Emerging Research in Computing, Information, Communication and Applications, Vol.3, pp 687-694, 2014

[8] M.S.D. Abhiram, Jyothsnavi Kuppili, N.Alivelu Manga, “Smart farming system using IOT for efficient crop growth,” in IEEE International students’ conference on Electrical, Electronics and Computer Science, DOI: 10.1109/SCEECS48394.2020.147, 2020

[9] Vikas Bavane, Arti Raut, Swapnil Sonune, “Protection of crops from wild animals using intelligent surveillance system,” in Special Issue National Conference – CONVERGENCE 2018.

[10] Billi Bhargav, “Automated system to protect the farm land from wild animals,” International Research Journal of Engineering and Technology (IRJET), Vol. 08, Issue 06, p. 2395-0072, 2021.

APPENDIX

Code for Face Detection

Code for Dataset

```
import cv2
import os

face_id = input('enter your id ')
# Start capturing video
vid_cam = cv2.VideoCapture(0)

# Detect object in video stream using Haarcascade Frontal Face
face_detector = cv2.CascadeClassifier('
    haarcascade_frontalface_default.xml')

# Initialize sample face image
count = 0



# Start looping
while (True):
    pwd=os.getcwd()

    # Capture video frame
    _, image_frame = vid_cam.read()
```

```
    # Convert frame to grayscale
    gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)

    # Detect frames of different sizes, list of faces rectangles
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

    # Loops for each faces
    for (x, y, w, h) in faces:
        # Crop the image frame into rectangle
        cv2.rectangle(image_frame, (x, y), (x + w, y + h), (255, 0,
            0), 2)

        # Increment sample face image
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count
            ) + ".jpg", gray[y:y + h, x:x + w])

        #os.chdir(pwd+'/dataset') #change direcytory
        # Display the video frame, with bounded rectangle on the
            person's face
        cv2.imshow('frame', image_frame)

    # To stop taking video, press 'q' for at least 100ms
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break

    # If image taken reach 100, stop taking video
    elif count >= 100:
        print("Successfully Captured")
        break

# Stop video
vid_cam.release()

# Close all started windows
cv2.destroyAllWindows()
```

Code for Training

```
import os,cv2;
import numpy as np
from PIL import Image;

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector= cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    #create empth face list
    faceSamples=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id from the image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces=detector.detectMultiScale(imageNp)
        #If a face is there then append that in the list as well as Id of it
        for (x,y,w,h) in faces:
            faceSamples.append(imageNp[y:y+h,x:x+w])
            Ids.append(Id)
    return faceSamples,Ids

faces,Ids = getImagesAndLabels('dataset')
s = recognizer.train(faces, np.array(Ids))
```

```
print("Successfully trained")
recognizer.write('trainer.yml')
```

Code for Recognization

```
import cv2
import numpy as np
import serial

ser=serial.Serial('COM3',9600)



recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);



cam = cv2.VideoCapture(0)
# recognizer = cv2.face.LBPHFaceRecognizer_create()
font = cv2.FONT_HERSHEY_SIMPLEX
while True:
    ret, im =cam.read()

    gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    faces=faceCascade.detectMultiScale(gray, 1.2,5)
    for(x,y,w,h) in faces:
        cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
        Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
        if(conf<70):
            if(Id==1):
                Id="akshaya"




        else:
            Id="Unknown"
            ser.write(b"u#")
```

```
        cv2.putText(im, str(Id), (x,y-40), font, 1, (255,255,255),
            3)


    im=cv2.resize(im,(600,400))

    cv2.imshow('face',im)
    if cv2.waitKey(10) & 0xFF==ord('q'):
        break

cam.release()
cv2.destroyAllWindows()
```

Code for Animal Detection

```
import cv2
import tensorflow as tf
import numpy as np
import os
import serial
from playsound import playsound  # Import the playsound library

ser = serial.Serial('COM3', 9600)

# Path to label map file
PATH_TO_LABELS = os.path.join('labelmap.txt')

# Load the label map
with open(PATH_TO_LABELS, 'r') as f:
    labels = [line.strip() for line in f.readlines()]

if labels[0] == '???':
    del(labels[0])

# Load model
```

```
interpreter = tf.lite.Interpreter(model_path="detect.tflite")
interpreter.allocate_tensors()
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

lis = ["elephant", "bear", "horse", "sheep"]
cap = cv2.VideoCapture(0)

while True:
    # Capture image
    ret, img_org = cap.read()
    key = cv2.waitKey(1)
    if key == 27:  # ESC
        break

    # Prepare input image
    img = cv2.cvtColor(img_org, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, (300, 300))
    img = img.reshape(1, img.shape[0], img.shape[1], img.shape[2])
       # (1, 300, 300, 3)
    img = img.astype(np.uint8)

    # Set input tensor
    interpreter.set_tensor(input_details[0]['index'], img)

    # Run
    interpreter.invoke()

    # Get output tensor
    boxes = interpreter.get_tensor(output_details[0]['index'])
    classes = interpreter.get_tensor(output_details[1]['index'])
    scores = interpreter.get_tensor(output_details[2]['index'])

    for i in range(boxes.shape[1]):
        object_name = labels[int(classes[0, i])]
        if object_name in lis:
            if scores[0, i] > 0.5:
                box = boxes[0, i, :]
```

```
                x0 = int(box[1] * img_org.shape[1])
                y0 = int(box[0] * img_org.shape[0])
                x1 = int(box[3] * img_org.shape[1])
                y1 = int(box[2] * img_org.shape[0])
                box = box.astype(int)
                cv2.rectangle(img_org, (x0, y0), (x1, y1), (255, 0,
                    0), 2)
                cv2.rectangle(img_org, (x0, y0), (x0 + 100, y0 - 30)
                    , (0, 0, 255), -1)

                cv2.putText(img_org, object_name, (x0, y0), cv2.
                    FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
                print(object_name)

                # Play audio for detected object
                if object_name == "elephant":
                    playsound('audio/bee.mp3')
                    ser.write(b"e#")
                if object_name == "bear":
                   # playsound('audio/elephant.mp3')
                    ser.write(b"b#")
                if object_name == "horse":
                    #playsound('audio/horse.mp3')
                    ser.write(b"h#")
                if object_name == "sheep":
                   # playsound('audio/sheep.mp3')
                    ser.write(b"s#")

    cv2.imshow('image', img_org)

cap.release()
cv2.destroyAllWindows()
```

- Reference paper

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/329671392

Protection of Crops from Wild Animals Using Intelligent Surveillance System

Conference Paper · December 2018

CITATIONS
19

READS
35,591

5 authors, including:


Vikas Bavane
5 PUBLICATIONS **58** CITATIONS
SEE PROFILE


Arti Raut
1 PUBLICATION **19** CITATIONS
SEE PROFILE


Pradip Jawandhiya
Pankaj Laddhad Institute of Tech & Management Studies,Buldana
70 PUBLICATIONS **854** CITATIONS
SEE PROFILE

All content following this page was uploaded by Vikas Bavane on 09 May 2019.

The user has requested enhancement of the downloaded file.

International Journal of Research in Advent Technology (IJRAT)
(E-ISSN: 2321-9637)
Special Issue National Conference "CONVERGENCE 2018", 09th April 2018

Protection of Crops from Wild Animals Using Intelligent Surveillance System

Vikas Bavane[1], Arti Raut[2], Swapnil Sonune[3],
UG Student Mechanical Engineering Department,
Pankaj Laddhad Institute of Technology and Management Studies, Buldana.
Prof. A.P.Bawane[4]
Professor, Pankaj Laddhad Institute of Technology and Management Studies, Buldana.
Dr.P.M.Jawandhiya[5]
Principal, Pankaj Laddhad Institute of Technology and Management Studies, Buldana.
Email: vikasbavane97@gmail.com[1], artiraut175@gmail.com[2], swapnilsonune001@gmail.com[3], apb_mech@plit.ac.in[4], principal@plit.ac.in[5]

Abstract- Surveillance plays a major role in many fields be it at home, hospitals, schools, public places, farmlands etc. It helps us to monitor a certain area and prevent theft and also provides proof of evidence. In the case of farmlands or agricultural lands surveillance is very important to prevent unauthorized people from gaining access to the area as well as to protect the area from animals. Various methods aim only at surveillance which is mainly for human intruders, but we tend to forget that the main enemies of such farmers are the animals which destroy the crops. This leads to poor yield of crops and significant financial loss to the owners of the farmland. This problem is so pronounced that sometimes the farmers decide to leave the areas barren due to such frequent animal attacks. This system helps us to keep away such wild animals from the farmlands as well as provides surveillance functionality.

Index Terms- Surveillance, monitor, unauthorized people, human intruders, leave the areas barren

1. INTRODUCTION

Animal attacks in India are a common story nowadays. Due to the unavailability of any detection system these attacks kill villagers and also destroy their crops. Due to lack of proper safety measures, these villagers are left helpless to their fate. Therefore a proper detection system could help save their lives and also to the preservation of crops. Also the crops of villagers are destroyed due to frequent interference of animals. The crops and paddy fields cannot be always fenced. So the possibility of crops being eaten away by cows and goats are very much present. This could result in huge wastage of crops produced by the farmers. To make the best use of mobile communication technology, the objectives of this paper therefore utilizes global system for mobile communication (GSM) and provide short message service (SMS). This system helps us to keep away such wild animals from the farmlands as well as provides surveillance functionality. It has been found that the odour of rotten egg helps to keep the wild pigs and deer from destroying the crops, hence the farmers manually spray the rotten egg solution on their fields, and firecrackers are used to ward off the wild elephants that destroy the crops.This project is based on surveillance with an animal ward-off system employed in farmlands in order to prevent crop vandalization by wild animals. In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID's, various PIR sensors are deployed in the area to detect any motion and hence turns ON a camera when movement is detected, thereby providing real time monitoring. It involves automation of certain methods used to prevent the wild animals from entering the farmlands and destroying the crops, an electronic fire cracker (for bigger animals, like elephant) and a rotten egg spray (for smaller animals like wild pigs and deer) which have been found useful to ward off the wild animals, we use Haar feature based cascade classifiers for object detection to distinguish between the animal and human. When such intrusions occur, a message will be automatically generated and the cameras employed are turned ON which capture an image and start recording the video for some time which will be stored on the SD card as well as stored on cloud i.e, dropbox, the land owner can then view the video on any smart device, as well as access it later. All the sensors and components are interfaced to the Raspberry pi board. Hence we come up with such a product that can be very useful for farmers, it prevents the loss of crops and increases the yield, also protects the farm from intruders.

2. LITERATURE REVIEW

T.Gayathri et al, [10] proposed the system for monitoring the growing status of the corn (maize) plant continuously and intimate the agriculturist using wireless sensor network (WSN). But in practice, cultivator faces too much effort in the farmland. This paper makes eases the work of the farmer in cultivated land through the usage of different kind of sensors. The two LDR sensors are interfaced with PIC16F877A microcontroller whereas its top array receives solar radiation for supply current and the bottom of the LDR array is for measuring leaf area index (LAI). The humidity sensor will compute the moisture level in the corn field, if the level decreases, then it automatically switches ON the DC motor. All the particulars of farmland are sent to the farmer through GSM and revel in the LCD screen. The temperature sensor will find the intensity of heat present in the soil. PH sensor is used to find the soil alkalinity which is essential for plant nutrition.

V Nainwal, et al, [16] Sensors are used to detect the presence of objects in the surveillance area and the information is collected over time to extract the event of interest. The information gathered by the surveillance camera i.e., video or still images could be used for further analysis and detection of the intruding object. This system does not utilize advanced techniques for alerting the owner of that area.

Sneha Nahatkar et al, [1] proposed a home embedded surveillance system which evaluates the development of a low cost security system using small PIR (Pyroelectric Infrared) sensor built around a microcontroller with ultra-low alert power. The system senses the signal generated by PIR sensor detecting the presence of individuals not at thermal equilibrium with the surrounding environment. On detecting the presence of any unauthorized person in any specific time interval, it triggers an alarm & sets up a call to a predefined number through a GSM modem. After the MCU sends the sensor signals to the embedded system, the program starts the Web camera which then captures the images which can be viewed and analysed later.

Puja G, Mohammad Umair Bagali proposed the system. This project is based on surveillance with an animal ward-off system employed in farmlands in order to prevent crop vandalization by wild animals. In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID's, various PIR sensors are deployed in the area to detect any motion and hence turns ON a camera when movement is detected, thereby providing real time monitoring. It involves automation of certain methods used to prevent the wild animals from entering the farmlands and destroying the crops, an electronic fire cracker.

3. EXISTING SYSTEM

The existing systems mainly provide the surveillance functionality. Also these systems don't provide protection from wild animals, especially in such an application area. They also need to take actions based on the on the type of animal that tries to enter the area, as different methods are adopted to prevent different animals from entering such restricted areas. Also the farmers resort to the other methods by erecting human puppets and effigies in their farms, which is ineffective in warding off the wild animals, though is useful to some extent to ward off birds .The other commonly used methods by the farmers in order to prevent the crop vandalization by animals include building physical barriers, use of electric fences and manual surveillance and various such exhaustive and dangerous methods.

4. STRATEGIES TO PROTECT CROPS

Successful farmers always seek to determine the satisfactory level of wild animal crop protection using one of the following technologies:

1. ***Agricultural fences***
   - *Wire fences*
   - *Plastic fences*
   - *Electric fences*
2. ***Natural repellents***
   - *Smoke*
   - *Fish or garlic natural emulsion*
   - *Chilli peppers*
   - *Lavender and beans*
   - *Egg based repellent*
3. ***Chemical repellent***
4. ***Biophysical barriers***
5. ***Electronic repellent***
   - *Ultrasonic Electronic repellent*
   - *Sonic Electronic repellent*

5. OBJECTIVES AND SCOPE OF STUDY

1. To design a security system for farm protection
2. Prohibit the entry of animal into the farm
3. Use GSM module for alerting us
4. Design a system that sounds through solar animal repellent when animal tries to enter into the farm
5. In night flash light will focus on that side.
6. The camera continuously monitors the fields and provides the video feed to the farmer at home 24×7 for the whole day
7. The system ensures that the alarm is not triggered by the presence of a human in the field, or via any random motion.
8. The system is capable of turning On/Off automatically and warding off the animals thus protecting the fields from any damage also we can setup a Timer as per farmer's requirement

6. PROPOSED SYSTEM

The proposed system uses a Raspberry Pi board which forms the main heart of the system; the different

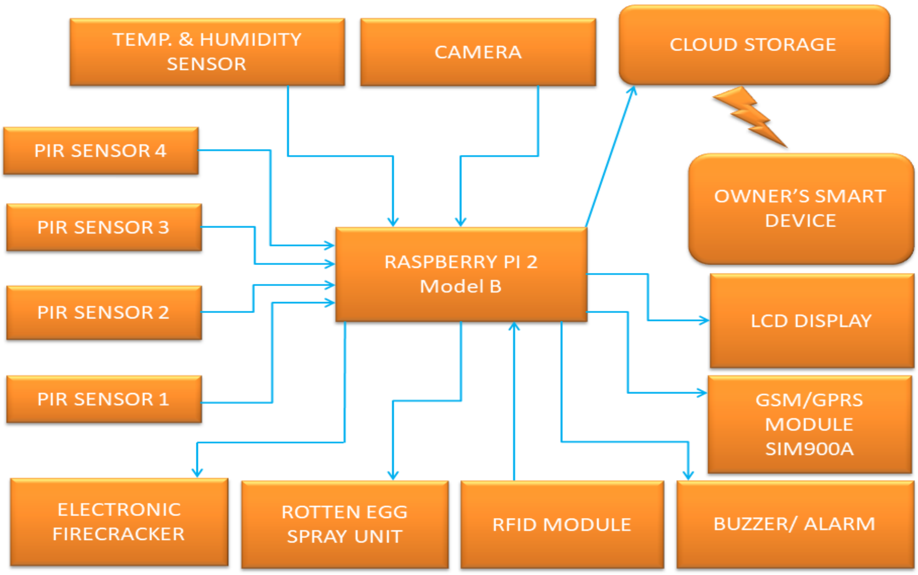


FIG.1. BLOCK DIAGRAM OF PROPOSED SYSTEM

sensors and camera are interfaced to the puppet. As soon as the PIR sensors go High on detecting motion within a range of 10 meters, the camera will be turned ON which first captures an image and then starts recording the video for about five to six minutes, which will be stored on board as well as cloud, simultaneously a message will be generated automatically to the registered number using a SIM900A module to inform about the intrusion along with the details of the temperature and humidity obtained by interfacing dht11 temperature and humidity sensor. If the motion detection is due to an authorized person with a valid RFID, who is mostly a farm worker, his attendance gets recorded automatically. Whereas if the motion detection is due to that of an unauthorized person without the valid RFID tag, the system further processes the image and video using Haar feature based Cascade Classifiers for object detection, and decides if the entity is an animal or human intruder.

We use Passive Infrared Sensors (PIR) to detect any motion of human body, once the employed PIR sensors detect motion the cameras capture an image and start recording the video as well as the owner of the farmland gets notified about the intrusion. This information along with the captured video is stored onto cloud from where the person in charge can access it once he receives the message. We use bash scripting for uploading the video to Dropbox. We also use RFID tags to differentiate between the authorized person and the intruders; if the person is an authorized one then no action is taken by the system. Whereas if the person is an unauthorized one the alarm or buzzer is turned ON to notify other people about the intrusion. Before which the system determines if the unauthorized person is an animal or human intruder based on Haar feature based cascade classifiers. If found to be an animal, the system then checks for the number of PIR sensors that have gone HIGH, if fewer number of sensors are high it denotes a smaller animal and all or more than half the sensors that turn high denoted it is a bigger animal and hence necessary action is employed to keep them away from destroying the crops. In order to automate the animal ward off system discussed, we take a decision based on the number of sensors that have gone high. The basic working principle is, if fewer numbers of sensors are able to detect the motion then it denotes an animal smaller in height such as a wild boar, deer etc., and we immediately turn on the rotten egg spray unit, which helps to keep away the pigs. Similarly if more than half or all of the employed PIR sensors have gone high it is naturally because of a huge animal such as the elephant which is another major threat to such farmlands, we initiate the electronic firecrackers to turn ON, the loud noise which in turn helps to ward off the bigger animals.

7. MERITS AND FEATURES

This system is very effective and carries following features and merits in comparison to the other solutions that exist in the current time.

1. ***Effective, accurate and adaptive:-***

This system is very effective in driving off the animals from the fields and keeping them away. It accurately determines the presence of animals in the fields and sounds the buzzer. It does not sound the buzzer due to the presence of a human being or due to some random motion. The ultrasonic buzzer is very effective against animals and causes no noise pollution.

2. ***Requires no human supervision:-***

This system requires almost no human supervision, except for the task of switching the system on and off. The system is capable of turning the buzzers on automatically and warding off the animals thus protecting the fields from any damage.

3. ***Economical:-***

This system is economical as compared to many of the existing solutions like electric fences, brick walls and manual supervision of the fields. Thus it saves a lot of money of the farmer.

4. ***Real time monitoring:-***

This system works in real time to detect the animals in the fields. The system enables the farmer to have a real time view of his fields from any place via internet and even provides manual buzzer controls if the need arises to use them. Thus the farmer is in effective control of the system and can manually sound the buzzer if needed.

5. ***Causes no harm to animals and humans:-***

This system is totally harmless and doesn't injure animals in any way. It also doesn't cause any harm to humans. Also this system has a very low power requirement thus reducing the hazards of electric shocks.

8. FUTURE SCOPE

1. In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID's.
2. We use Haar feature based cascade classifiers for object detection to distinguish between the animal and human.
3. When such intrusions occur the cameras employed are turned ON which capture an image and start recording the video for some time which will be stored on the SD card as well as stored on cloud i.e. dropbox, the land owner can then view the video on any smart device.
4. If the motion detection is due to an authorized person with a valid RFID, who is mostly a farm worker, his attendance gets recorded automatically.
5. We can design a IOT based application to provide an image and video feed to farmer on any smart device and farmer will be notified when there is an intrusion in the farm by animal along with additional information of humidity and temperature

9. CONCLUSIONS

The problem of crop vandalization by wild animals has become a major social problem in the current time. It requires urgent attention and an effective solution. Thus this project carries a great social relevance as it aims to address this problem. Hence we have designed a smart embedded farmland protection and surveillance based system which is low cost, and also consumes less energy. The main aim is to prevent the loss of crops and to protect the area from intruders and wild animals which pose a major threat to the agricultural areas. Such a system will be helpful to the farmers in protecting their orchards and fields and save them from significant financial losses and also saves them from unproductive efforts that they endure for the protection of their fields. This system will also help them in achieving better crop yields thus leading to their economic wellbeing.

10. ACKNOWLEDGEMENT

I would take this opportunity to thank all those who helped me complete my project on time and a special thanks to Mr. V G Bhattad, VP Operation Engineer and Experimental Physicist who guided us for construction of our project prototype.

REFERENCES

[1]. Ms. Sneha Nahatkar, Prof. Avinash Gaur, Prof. Tareek M. Pattewa "Design of a Home Embedded Surveillance System with Pyroelectric Infrared Sensor & Ultra-Low Alert Power" International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 1, Issue 3, September 2012.

[2]. M. Sathishkumar1, S.Rajini "Smart Surveillance System Using PIR Sensor Network and GSM" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 1, January 2015.] S. Sivagamasundari, S. Janani, "Home surveillance system based on MCU and GSM", International journal of communications and engineering, 2014, volume 06– no.6.

[3]. Padmashree S. Dhake, Sumedha S. Borde, "Embedded Surveillance System Using PIR Sensor", International Journal of Advanced Technology in Engineering and Science, www.ijates.com Volume No.02, Issue No. 03, March 2014.

[4]. Sudhir G. Nikhade, "Wireless Sensor Network System using Raspberry Pi and Zigbee for

Environmental Monitoring Applications", 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. Pp.376-381.

[5]. Pramod P. J, S. V Srikanth, Vivek N, Mahesh U Patil, Sarat Chandra Babu N, "Intelligent Intrusion Detection System (In2DS) using Wireless Sensor Networks", Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan, March 26-29, 2009.

[6] Fackelmeier A, Biebl E.M, "A Multistatic Raddar for Detecting Wild Animals" Tech University, Munich, ISBN 978-1-4244-4747-3.

[7] Sumit Kumar Tetarave, Ashish Kumar Shrivatsava, "A Complete Safety For Wildlife Using Mobile Agents and Sensor Clouds in WSN" IJCSI International Journal of Computer Sciences, Vol 9, Issue 6, No 3, Nov 2012, ISBN 1694-0814

[8] A Mammerri, "Multi static radar for detection of wild animals" Christ Church University, U.K.

[9] Tito Bukhart, "Tracking animals in wildlife videos using face detection" Boston University, USA

[10] Surabhi Agarwal and Chandrika Chanda, "Wireless Design Of Low Cost Detection System" Vellore Institute of Technology, Tamil Nadu, India.

[11] Mohamed Amine Kafi, Yacine Challal, Djamel Djenouria, Messaoud Doudou, Abdelmadjid Bouabdallah, Nadjib Badache "A Study of Wireless Sensor Networks for Urban Traffic Monitoring: Applications and Architectures" The 4th International Conference on Ambient Systems, Networks and Technologies CERIST, Centre for Research on Scientific and Technical Information, Algiers, ALGERIA.

[12] Mohd Fauzi Othmana, Khairunnisa Shazalib "Wireless Sensor Network Applications: A Study in Environment Monitoring System" Centre for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, Jalan Semarak,54100 Kuala Lumpur, Malaysia, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Semarak,54100 Kuala Lumpur, Malaysia.

AUTHOR'S BIOGRAPHY:

Dr. P. M. Jawandhiya
Principal, PLITMS, Buldana.
SGBAU Amravati, (MS), India.

Prof. A.P.Bawane
Professor, Mechanical Engineering Department,
PLITMS, Buldana.

Mr. Vikas S. Bavane
UG Student, Mechanical Engineering Department,
PLITMS, Buldana.

Ms. Arti S. Raut
UG Student, Mechanical Engineering Department,
PLITMS, Buldana.

Mr. Swapnil N. Sonune
UG Student, Mechanical Engineering Department,
PLITMS, Buldana.

Department of Engineering

Vidya Academy of Science & Technology

Thalakkottukara, Thrissur - 680 501

(http://www.vidyaacademy.ac.in)