SNAKE DETECTION IN AGRICULTURAL FIELDS USING IOT

A Major Project Report Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Mr. Akkala Karthik	**(19071A05C1)**
Mr. Jagarlamudi Venkat	**(19071A05E3)**
Mr. Kalimera Teja Kumar	**(19071A05E4)**
Mr. Mogili Sunischal	**(19071A05F1)**
Mr. Thanikella Venkata Nikhil	**(19071A05H5)**

Under the Guidance of

Dr. S. Nagini

(Professor & HOD, Department of CSE, VNR VJIET)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade NBA Accredited for, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as “College with Potential for Excellence” by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade NBA Accredited, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as “College with Potential for Excellence” by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090 TS, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TAMASOMA JYOTHIRGAMAYA

CERTIFICATE

This is to certify that the project report entitled “**SNAKE DETECTION IN AGRICULTURAL FIELDS USING IOT**” is a bonafide work done under my supervision and is being submitted by **Mr. Akkala Karthik (19071A05C1), Mr. Jagarlamudi Venkat (19071A05E3), Mr. Kalimera Teja Kumar (19071A05E4), Mr. Mogili Sunischal (19071A05F1) and Mr. Thanikella Venkata Nikhil (19071A05H5)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, VNR VJIET, Hyderabad, during the academic year 2022–2023. Certified further that, to the best of our knowledge, the work presented in this thesis has not been submitted to any other university or institute for the award of any degree or diploma.

Dr. S. Nagini
Professor, HOD & Internal Guide
CSE Department, VNR VJIET

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as "College with Potential for Excellence" by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We declare that the major project work entitled "**SNAKE DETECTION IN AGRICULTURAL FIELDS USING IOT**" submitted in the department of Computer Science and Engineering, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is a bonafide record of our own work carried out under the supervision of **Dr. S. Nagini, Professor & HOD, Department of CSE, VNR VJIET**. Also, we declare that the matter embodied in this thesis has not been submitted by us, in full or in any part thereof, for the award of any degree or diploma by any other institution or university previously.

Place: Hyderabad.

Mr. A Karthik	Mr. J Venkat	Mr. K Teja Kumar	Mr. M. Sunischal	Mr. T V Nikhil
(19071A05C1)	(19071A05E3)	(19071A05E4)	(19071A05F1)	(19071A05H5)

ACKNOWLEDGEMENT

We would like to express our immense gratitude towards our institution, the VNR Vignana Jyothi Institute of Engineering and Technology, which created a great platform to attain profound technical skills in the field of Computer Science, thereby fulfilling our most cherished goal.

We are very thankful to our principal, **Dr. Challa Dhanunjaya Naidu** and our head of department, **Dr. S. Nagini**, for their cooperation in completing this project within the stipulated time.

We extend our heartfelt thanks to our supervisor, **Dr. S. Nagini,** and the project coordinators, **Dr. P. V. Shiva Kumar**, **Dr. C. Kiranmai,** and **Mrs. K. Haripriya** for their enthusiastic guidance throughout the course of our project.

Finally, our appreciable obligation also goes to all the staff members of the Computer Science and Engineering department and to our fellow classmates who directly or indirectly helped us, during the course of our project.

Mr. A. Karthik (19071A05C1)

Mr. J. Venkat (19071A05E3)

Mr. K. Teja Kumar (19071A05E4)

Mr. M.Sunischal (19071A05F1)

Mr. T. V. Nikhil (19071A05H5)

ABSTRACT

Snakebites have been a major problem among the farmers, especially in rural areas. According to a recent survey by WHO[32], 58,000 Indians (more than 50% of the world's snake bites) die from snake bites annually. Almost 1.2 million Indians have lost their lives since 2000 in this manner. "Time" is the most valuable asset here in these types of cases. We can save people if they can get quick medical help. If there is a delay in this, the snake bite may cause serious damage to the organs, or in the worst cases, the person may die. In this modern age, the IoT technology and machine learning can be used to prevent these accidental snake bites. Different sensors like vibration and PIR sensors can be used on the field to monitor unusual activity like that of different animals on the field and then the deep learning models can detect the snakes in the images, so that the farmers can be alerted by sending a notification through email to their mobiles and take precautionary measures against these dangerous reptiles.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 : INTRODUCTION

India is a very diverse country and home to many snake species. Of all those species, some are very poisonous and deadly. India is a land that depends a lot on agriculture for economic growth. Farmers are always in grave danger because they spend most of their time in the field, where the snake may be present. Thus, snake bites become a major problem that must be solved as human lives are in danger.

According to the World Health Organization (WHO)[32], although the precise count of snake bites is unclear, it is believed that 5.4 million individuals are bitten every year, with close to 2.7 million of those bites resulting in poison. Every year, between 81,000 and 138,000 individuals pass away as a result of snake bites. Also in India, it is found that 58,000 people are dying due to snake bites annually, which is higher than any other country in the world. Venomous snake bites can result in a variety of medical conditions, including breathing difficulties, bleeding abnormalities that can result in tissue injury and deadly haemorrhage, and irreparable kidney failure that can cause long-term incapacity and loss of limbs. The most impacted groups are kids and farm labourers. Due to their smaller body mass than adults, children frequently have more severe impacts.

Sometimes, farmers do not actually recognize the snake species, and if they are lucky enough, they will kill them right away so they do not get hurt. The fact that snakes are a crucial component of the food chain and that their widespread extinction might upset the balance makes this a potential issue as well. If snakes are killed in large numbers, this could disrupt the food chain. Therefore, we must reduce the harm humans do to them in order to protect them as well.

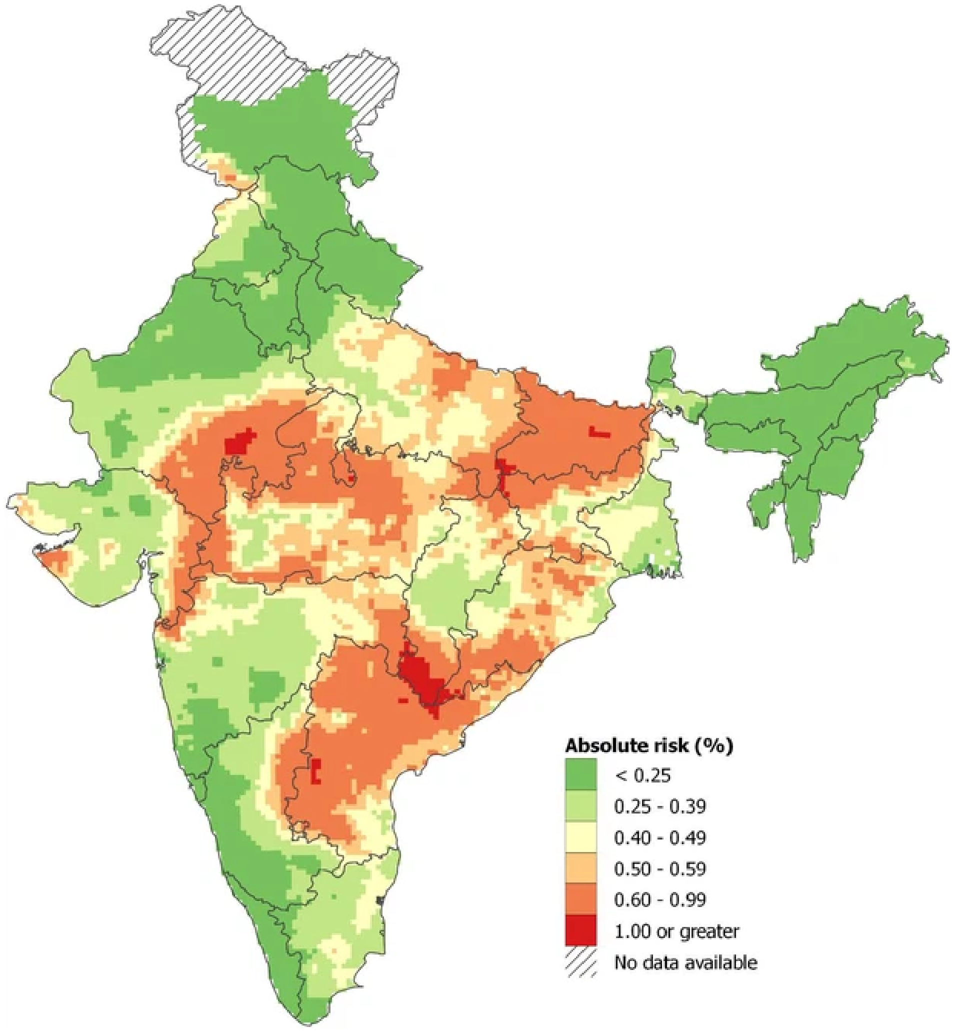

Fig 1.1: Spatial Distribution of Snakebite Mortality Risk in India for 2004-13 [10]

Figure 1.1 shows the absolute risk percentage of snakebite in India which is represented in different colours. Green represents very less and red represents a very high risk percentage.

So, there must be a system where the sensors placed on the borders of the field can detect the snake movements and capture an image that is then analysed and processed by various deep learning techniques to find out if it's a venomous snake or not. If it's venomous, the farmers are alerted immediately with the help of a message so that they will take some precautionary measures to quickly get out of trouble. The alerts will also be sent to members near the farm.

As a part of implementation, we have simulated the project by creating a model of a farm field. This model includes features like sand, a toy snake, toy trees, and grass, along with some sensors and a camera. Vibration sensor and PIR were used, and we selected a laptop webcam as the camera as it is just a simulation.

The main objective of this project simulation is to tell whether an image contains a snake or not. This is done by capturing a snap of the toy snake whenever there is an unusual activity and then sending it to the deep learning model for prediction. If the model recognizes a snake in the picture, it displays that the snake is detected on the computer screen.

In real-world scenarios, the same may be implemented by placing sensors along the fence and monitoring the situation. Expensive and advanced equipment like LiDAR may be used to achieve it. But, here in this simulation, we have focused on affordability and used basic sensors to their fullest capacity to detect snakes. Despite our efforts to optimise the model, it sometimes produces incorrect results. This may be due to the fact that the transfer learning model used is not as complex or capable, or that the dataset used is limited in size and diversity. The data used for the model was manually collected, which obviously makes it less diverse.

Fig 1.2: Created Model of the Farm Field

Figure 1.2 illustrates the model of the farm field which was created using the thermocol, plastic plants, toy snake and sand.

CHAPTER 2: LITERATURE SURVEY

To accomplish the animal detection in the farm fields, a system that makes use of image analysis, CNN-based networks, and deep learning strategies was proposed by the authors[1]. The automated image categorization system heavily utilises CNN. Typically, characteristics are extracted and used for categorization. The authors used a database that consists of 3050 RGB images of varying sizes, split into 28 species. For preprocessing data, the GrabCut algorithm is applied to images, data augmentation is performed, and the dataset is trained using DenseNet 161.

Authors focussed on the point that harming intruding and dangerous snakes is not an option in this paper[2]. So, for these dangerous animals, a system is proposed that recognizes the snake species accurately in less time, and an effective "convey system" is combined with the application or system to convey snakes to the closest emergency centre or catcher. The aim of the proposed system is to quickly assist people in finding the nearest medical help, such as a hospital, in the event of a snake bite. The system uses an Android app that takes images of snakes and predicts their species using TensorFlow. In addition, the system also allows for the analysis of snake bite occurrences to gain insights into snake populations. Since doctors may not be trained to identify snake species, the system aims to provide this information and help victims find the nearest appropriate medical care. The tensor flow algorithm has three phases: classification, detection, and result generation. This API can predict the type of snake present in the input photograph. The "convoy system" uses a combination of two algorithms, geo-sorting and the shortest path, to send a message to the nearest emergency centre and snake catcher, improving the performance and speed of the model.

Several examples of how dangerous animals pose a threat to farmers in agricultural fields was presented by the authors[3]. In particular, it focuses on the harmful impact of snakes, which cause the deaths of many farmers in the age range of 20 to 50 every year. Additionally, these animals also cause significant losses in productivity for farmers. To address these issues, the authors propose a solution involving the use of an algorithm and IoT systems to detect the presence of dangerous animals in agricultural fields. The algorithm employs feature extraction and classification techniques to identify the unique characteristics of snakes and other harmful insects. It then generates a buzzer to alert farmers of the animal's location, which is determined using GPS. The Raspberry Pi is used (for computation and processing) as the main platform for the system. The existing system only detects the presence of animals but does not provide any preventive measures. The paper suggests using a feature extraction method to identify moving objects and creating a database of unique features belonging to each snake and dangerous insect to train the classifier. Different technologies, such as ultrasonic sensors, light-dependent resistors, GPS, and buzzers, are used to achieve this goal.

A technique for protecting crops from animals that makes use of a wireless sensor network that is based on the Internet of Things (IoT) was suggested by authors[4]. It uses the aforementioned approach when animals are on the farm. When an animal is detected, sensors trigger a camera to take a picture of the animal, which is then sent to a Raspberry Pi for analysis. The Raspberry Pi compares the image with those in its database, and if there is a match, it sends a command to the GSM (Global System for Mobile Communications) module to communicate with the farm's owner. This allows the owner to take appropriate action, such as scaring the animal away, to prevent damage to their crops.

A project that contains a microcontroller, solar panel, battery, MOSFET, and MOSFET driver, as well as an extremely high threshold voltage that can be used to protect farms from wild animals was discussed[5]. The goal of this initiative is to safeguard both animal and human lives. If any animals attempt to cross the farm fence, which is completely covered with electrical cable with low current flowing through it, they will be shocked. LEDs emit infrared radiation, which travels in the direction in which it is oriented. When an obstruction blocks the passage, the IR rays will be broken up and form secondary wavelets, which mostly travel in the opposite direction to the primary waves and have the effect of reflecting IR photons. The microcontroller will receive the relevant signal if any object crosses the entryway and surrounding region. The alarm will ring once it receives this signal to notify a farm disruption. The MOSFET regulates voltage and is supplied to the threshold circuit, while the MOSFET driver circuit receives input from the microcontroller. The microcontroller sets conditions for all sensors and regulates them all.

The growing monkeys are a threat on Indian farming fields causing a lot of problems for farmers was discussed[6]. The enormous fruit and vegetable harvests, worth tens of thousands of rupees, were completely destroyed and ruined by the monkeys. Infrasound, seismic interaction, and light and photo options are just a few of the present detection methods that are both extremely costly and ineffective. So they suggested a detection system consisting of wireless sensor nodes connected to a noise producer that emits ultrasonic waves to scare away monkeys. The wireless nodes consist of sink nodes and sensor nodes. The sensor nodes are installed along agricultural fields' borders, which are used for the detection of monkey movement. When a detection is made, the sensor sends an alert, which is subsequently sent from the sensor nodes to the sink node. The ultrasonic sound producer is then activated by the sink node, producing ultrasonic sounds at a rate of 20 kHz that annoy the monkeys and finally cause them to leave the field.

"Sri Lanka" is an example for implementing the project of identification of particular species occurring most frequently there was discussed[7]. The paper highlights the problem of snake bite deaths among Sri Lankan farmers, specifically how the current system is inadequate and leads to delays in treatment. The authors propose a solution using a convolutional neural network with 2000 images of the six predominant snake species found in Sri Lanka. Five models were developed, including one from scratch, and the best model in terms of accuracy was determined for automatic snake identification. The authors collected images and videos of snakes using Google Images. These videos were augmented to create plenty of images to work with, and the backgrounds were removed. Of the 2,000 images collected, 1,200 were used for training neural network models, 400 for validation, and 400 for testing. Batches of size 32 were created for input. Adjustments were automatically made to the weights before the next iteration of the convolutional neural network (CNN) began. Models like InceptionV3, VGG16, ResNet50, MobileNet with ImageNet, and a new model were trained using TensorFlow as the backend and Keras as the front end. Various measures were taken to improve accuracy, such as inputting cropped images of different dimensions and trying raw images. This training was performed on a Core i7 machine. After training, a testing dataset was used to evaluate the models. These results provided accuracy for the model. The "Softmax activation function" was used to classify the images. A sequential model (in Keras) was built from scratch, achieving the highest accuracy among all other models.

Snakebites are common in certain parts of the world, but local knowledge is not enough to identify the species of snake that was discussed[8]. To address this, they propose a system that uses accurate predictions of snake species to enable faster treatment. The system is designed to help doctors quickly identify the species of snake, allowing them to provide timely treatment. The proposed system aims to prevent harm to snakes and provide accurate information about their species, enabling people bitten by snakes to receive the appropriate medication quickly. The system uses image processing, including YOLO (you only look once), to capture and classify images and videos of snakes. The system has two modules: object detection and classification. The object detection module uses YOLO, which is based on deep learning and convolutional neural networks (CNNs), to locate snakes in the images. The classification module then uses the bounding box surrounding the detected snake to classify it by species.

The difficulty of identifying snakes and the use of computer vision technology to improve accuracy was discussed[9]. It illustrates that extraction of features and classification are the two steps involved in machine learning-based image categorization. The classes are pre-set, and the procedure comprises categorising the test data using the trained model after a training stage that uses the training data. They point out that snakes' flexible and elongated bodies cause fluctuations in their posture and deformation, making it challenging to identify characteristics from dorsal body arrangements. Additionally, they note that a lack of specialised image datasets for snakes makes it difficult to train deep convolutional neural networks for this task. Additionally, they suggest that museum samples, which lack original colour and position, are useless for inclusion in full body image collections. This study aims to compare the accuracy of several machine learning methods in the classification of snake images. Out of 594 photographs, six different snake species could be distinguished; only those showing at least 50% of the snake's body were used. A method for extracting features was combined with conventional classification methods, and transfer learning was used to improve the performance of a deep neural network model on the small dataset. The results of this study will help identify the most effective machine learning methods for snake image classification.

A computer method to recognize several snake species from pictures was developed by the authors[10]. The system performed noticeably better than arbitrarily guessing when it was compared with human performance. Some species were easily identified by both the algorithm and humans, while others were more difficult to distinguish. The researchers discovered that when detecting photographs with visual artefacts or of low quality, human beings had an edge over the software. Future research on snakebite epidemiology may benefit from computer vision technologies, particularly when paired with location data and expert advice. The development of a computer vision method for classifying snake species was indeed the aim of this project. The algorithm was tested and compared with human performance in identifying snake species. The algorithm was found to be better than randomly guessing but had difficulty identifying certain groups of species. The algorithm could potentially be used by healthcare providers to quickly identify snakes involved in snakebite cases. The algorithm was developed using a collaborative approach with input from data science experts and enthusiasts. For picture categorization, it used massive, deep CNNs (convolutional neural networks).

A system whose goal is to determine the species of these dangerous reptiles using their unusual traits so that the appropriate course of action may be taken to prevent deaths from snake bites was discussed in the paper[11]. The authors of this system employed deep learning techniques like CNN and image analysis to obtain the same results. CNN was heavily utilised for picture classification by machine. Feature extraction, a technique used for classification, is used to extract the significant and distinctive aspects of the snakes. In order for necessary follow-up steps to be performed, it is expected that this system will accurately categorise snakes and report their species. Due to the similarity in the species' traits, including texture, colour, and head form, automated snake species detection is a difficult undertaking. Rapid snake species identification can be aided by deep learning and transfer learning. One of the writers has suggested a method in which MobileNetv2 is used for identification.

Wild animals are capable of harming crops. Monitoring the existence of animals in the area is therefore crucial. So, the authors proposed a system where fencing wire serves as a detector in their envisioned research[12]. When animals come into contact with this exposed cable, we will receive the first input that shows the presence of creatures at the barrier. The amplifier circuit then sent the signal on for further analysis after receiving the original input signal. After that, the microcontroller will receive it, a buzzer will sound immediately, a flashlight will turn on at night, and the farmer will be sent a text message. Solar panels or a controlled power source will provide the electricity. For sending the message to the farmer, a microcontroller employs the GSM module.

How snake bites cause deaths and may lead to amputations is discussed [13]. It also discusses the snakes in Malaysia and why it is crucial to differentiate between snakes that are poisonous and those that are not. The paper proposes a snake bite identification system that uses a back propagation neural network and Neuro-GA, which is a genetic algorithm. It compares two neural network techniques. The system mainly uses features of snakes like shape of head, shape of tail and colour of snake and uses them to classify whether the snake is poisonous or not using neural networks. The paper finds that BPNN is the best method to classify snakes, and a method that uses BPNN and GA could provide more accuracy if it involves large data.

A system that employs wireless sensor networks and the IOT to provide a wireless-based sensor network to detect and prevent wild animals from attacking farmlands was proposed [14]. The goal of this study is to identify wild animals close to the forest's edge. To accomplish this, we use a variety of sensors, most notably the Raspberry Pi, which acts as the main node and collects signals from the various sensors before forwarding them to the server to be processed further. Other detectors we use to accomplish this include the passive infrared motion sensor, the sound-recognition sensor, web cameras, some audio speakers, Arduino, and primarily the Raspberry Pi. The server processes the data and, using that data, sends signals to the speakers so they may play the sound to deter animals from entering farmland, and it also sends a signal to the farmer to be alert in case of an intruder. For farmers, the issue of crop devastation by wild animals has gotten much worse. To address and solve this significant issue, the paper provides a smart system for the earlier identification of wild animals and alerting farmers. The main goal of the paper is to stop crop losses and safeguard agricultural fields from wild animals that do significant harm to the crops. It's quite beneficial to take precautions at an early stage when there are signs of animals close to the forest boundary. The paper proposes a scientific method that will assist farmers in safeguarding fields, prevent them from suffering financial losses, and save them from making ineffective attempts to provide protection.

Using image processing methods like image enhancement and object detection can be used to recognize different species of snakes automatically was discussed[15]. This technique aids in decreasing the number of people who die from snake bites and quickly recommends the best anti-venom for the victim. The earlier research used machine learning and older deep architectures to create systems with comparatively smaller databases. Only a small number of snake species could be identified by the systems, and/or their accuracy was lower. The method may be strengthened by further proving the snake classification, though. The proposed method uses the more recent ResNeXt50-V2 deep learning architecture and a substantially bigger dataset to identify 772 classes of snake species. There are 412,537 photos in the whole data set that the Snake Species Identification Challenge has released. Images of 772 distinct snake species from 188 nations are included in this collection. A training set containing 347,405 photographs and a validation set with 38,601 photographs were created from the data set. Baseline convolutional neural network (CNN) models may be constructed in an attempt to boost performance and handle difficult problems. The residual network block is ResNet's central concept. This block is made up of a skip connection, which bypasses certain neural network layers by passing the output of one layer to the next one—not necessarily the one directly below it. The gradient is given a different route by employing a skip connection. We can avoid the vanishing gradient issue thanks to this. ResNeXt architecture was presented to enhance ResNet architecture's performance without adding more parameters. The procedures listed below were used to create the proposed automatic snake classification system: Data pre-processing, batch accumulation, model construction, and ensemble model testing.

The disadvantages of the traditional surveillance systems like more consumption of power was discussed[16]. Therefore, the authors suggested an integrated surveillance system employing ultra-low warning power that comprises a PIR sensor, a minimum power sensor, to minimise the power utilisation of the conventional monitoring system. Security and protection are among the most hotly debated subjects in practically every industry, including surveillance, business applications, workplaces, and generally in smart settings. In this study, they constructed a minimum-cost security system employing a tiny PIR (pyroelectric infrared) sensor built around a microprocessor with very minimal alert power. The system will be used to assess the construction of a home-integrated monitoring system. The PIR sensor's signal alerts the system to the presence of people who are not thermally equilibrated with their surroundings. It sends out the signal to awaken the MCU when it recognizes the existence of any visitor in any certain time frame. The application launches the Web camera once the MCU transmits the sensed data to the microcontroller. This sensing project will demonstrate the decrease in memory use needed for preserving historical data as well as the device's power usage.

A tiny, inexpensive, and highly accurate barrier breach detection mechanism that can identify many kinds of movement on the fence was discussed[17]. The hardware is built using a RISC CPU and a 3-axis accelerometer. The system includes a wireless gadget that allows it to operate remotely and interact with a base station. To distinguish between action and inactivity on the fence, we have created an algorithm. Furthermore, the algorithm can recognise the sort of breach; whether it was caused by rattling generated by high wind or by a human climbing on the fence. The recognition algorithm can be integrated into the local RISC microcontroller because computation costs are low. The system proposed has a high range of accuracy.

A system that uses neural networks to detect livestock theft and farm intrusions was proposed by the authors[18]. The paper proposes a system that uses an elaborate farm information system. The system uses various sensors to cover a large distance of the fence by installing sensors along the fence. It uses IR and optical fibre systems as well as some acoustical based systems, The project also uses neural networks to detect intrusion based on spikes detected by sensors (it uses acoustical based spikes to detect intrusions). It tries to enhance existing systems and tries to provide a system of fencing which is quite dangerous both for humans and animals.

How Asian elephants cause serious human injuries was discussed[19]. The main objectives of the research paper are to detect and locate damages to electric fences and warn communities which can be threatened by this fence intrusion. The project uses RTU (Remote

Transmitting Unit), LITs (Location identification tags), these devices are interlinked with the help of thin copper wires. The devices detect intrusion with the help of RTU, the devices communicate the intrusion using LIT/Bridge. The system has been tested in real life conditions containing elephants, after various testing power efficiency has been improved, the proposed system can increase efficiency of existing systems.

A system that can be used to monitor the activity of lizard Chamaeleo jacksonii was discussed[20]. The lizard that the paper is researching on is a tiny animal that was monitored and classified according to its degree of activity using a multiple radar system in a non-confined laboratory setting. The motion is identified by the classification system as either fidgeting or movement. A system like this is valuable because it can be adjusted to recognise certain behaviours of other species and can track the chameleon's extremely slow movements. Visual references and video capture have been used to confirm the data that was produced. The factors that should be considered while choosing the best Doppler radar system have been covered in this paper. The paper proposes a system that uses Continuous wave Doppler radar. This radar can be used to extract information about lizard velocity. The radars operate at 24 GHZ and 10.525 GHZ. After detecting lizard activity its motion is classified using eigen demodulation. The paper can be applied to detect and classify motions of animals be it large motion or small motion.

Importance of technology in our daily lives and how Internet of Things (IoT) demand has increased significantly, which has attracted major study was discussed[21]. Both in the academic community and the business community. IoT adoption has facilitated smart farming and precision agriculture, to name a few, solely in the agriculture industry. In order to avoid animal invasions into crop fields, this paper describes the creation of an Internet of Things application for crop security and to guard against damage from weather events and wild animal attacks, The farm is equipped with a repelling and monitoring system. The downsides of the existing approach, which employs electric fences or chemical fumes to scare animals away, are discussed in the study. These methods injure animals and are frequently ineffectual. In order to create a better crop protection system, this study aims to suggest a novel system that makes use of sensors and actuators that are connected to the cloud. It makes use of wireless innovations like WiFi, Zigbee, etc.The method that is suggested in the research uses solar energy and will be financially advantageous over the long term. The primary components of the system's proposed design are a Repeller device that uses pir sensors, a back-end system that utilises a CPU and is connected to an openTSDB NoSQL database, RIOT OS is the operating system in use. Wi-Fi is used to install the network. A weather monitoring system is also part of the system. The future scope, which includes expanding the monitoring system and current hardware and software, is also covered in the article. The study describes an inexpensive, internet-of-things (IoT)-based system that uses real-time data and provides notifications.

The way a human reacts when he sees or identifies a snake in an environment filled with other wild animals was discussed[22]. This paper provides behavioural evidence for the snake detection theory(SDT) and how humans react when they approach or identify snakes. The study presents a setting to investigate how a person responds to a snake. This essay explores how snakes still have the potential to prey on humans and poison them. The theory of snake detection is also covered. It gauges how people react psychologically when they come into contact with snakes. It illustrates an example of how, in contrast to other species, humans and monkeys have quicker reaction times while looking at pictures of snakes. It also covers how a person responds to a snake, which is by becoming more cautious and paying closer attention to their surroundings. This essay details the interactions between college students at California State University, San Marcos, and their reactions. They set up two snake models, two rabbit models, and two glass bottles across a hiking trail, observed how people reacted to them using iWorx to determine skin conductance response, and measured heart rate using aPT 104-thsymograph sensor connected to a biopotential amplifier. The participant's non-dominant hand's index finger was where the sensors were attached. According to the experiment's findings, everyone was able to correctly identify snakes and did so more readily by displaying distinct SCR and HR responses.

Easily understood technique that leverages snakes inside trees to detect and identify items automatically was proposed[23]. The recommended method makes use of recursive computation of snakes, also known as parametric active contours, which produces multi-layered snakes. The initial layer corresponds to the main object of interest, while the succeeding levels specify the numerous foreground elements. These snakes split regions and retrieved visual characteristics that were then translated into semantic notions. Decision trees are generated based on these characteristics, thereby semantically classifying the objects and automatically annotating the picture. These all include automated object identification and are computer vision-based applications. Automatic object detection is used for a variety of tasks, including object recognition, scene recognition, activity recognition, object tracking, or automated image annotation. These tasks can then be used for a variety of tasks, including licence plate recognition, robot navigation, pedestrian detection and surveillance, car detection and annotation, and many other tasks. As a result, this research suggests a unique, AI-based method for intelligent vision systems that is explainable by design. Our solution is based mostly on the recursive calculation of the machine-learning-based decision trees and the snake's computer vision technique.

A method to classify animal behaviour by detecting and tracing them in videos was proposed[24]. Lions are chosen as the example for this paper. AdaBoost classification is used extensively. Face detection and tracing is used by a particular model which uses feature tracking on some low or basic features. Accurate and temporally coherent identification and tracking of animal heads are made possible by merging the two techniques in a particular tracking model. The tracker's data are utilised to autonomously label the animal's locomotor behaviour. The Kanade-Lucas-Tomasi algorithm is used. The model is continuously trained by giving it new data so that more accuracy is achieved. The main focus of the future work is to detect all kinds of animals.

Serious injuries are happening because of accidents on the road with animals, as claimed by the authors[25]. In this paper, a system is proposed which is not very complex and is cheap. It detects animals on the highways and other roads where there may be more possibilities of accidents. Computer vision methods are used for the same purpose. Distance of these animals is tracked. Dataset taken has about 2200 pictures of the animals which are tested on videos available over the internet. After detection, the driver is alerted that there’s an animal if he is driving below 35 kilometres per hour. An accuracy of 82.5% is obtained. This system can be adapted in other countries also.

Dangerous animals entering the vicinity of neighbourhood or populated places has become very common now-a-days was discussed[26]. In this paper, they have proposed a system where it alerts people if a wild animal steps out of the forest. Cheap motion sensors and computers with less computing power are utilised to achieve this. A control centre is set up for computations. We also suggest a repellent that can stop wild creatures from fleeing the forest, such as producing a lot of noise via speakers. The fundamental tenet of IOT is to link various sensors, facilitate contact, and offer services. In this paper, we develop an alarm system using a number of IOT devices placed around a nature reserve. Smugglers as well as other persons breaking the law by entering the forest can also be caught using this approach. The place is surrounded using many sensor towers and some of them are connected to the control centre. They are placed at the boundaries to detect animal movement. Raspberry Pi is used. Cameras are also connected to it. It uses very little power. Computations performed here are transmitted to the service centre. PIR sensors are used to detect motion. Additionally, the sensor tower incorporates a GPRS/3G module for communication with the control room. Since they are nearer to people, the forest's borders have GPRS connectivity. This facilitates communication and makes it easier. The PIR sensor detects motion, when it does, we snap a photo of the area and transfer it back to the control centre. We further SMS the relevant official. In order to prevent wildlife from entering the forest border, the control centre may transmit a signal to the relevant tower to generate a loud noise. Each tower has a defined position, so we are aware of it. The authorities can promptly travel to the area if necessary and take additional action. Solar power is widely used at sensor modules to cut down the costs. As mentioned earlier, the main computing power which is raspberry pi is connected to a camera which is going to take photos whenever it detects some movement and then transmits it to the main server over the internet. As future work, much efficient and high performing sensors need to be used because they perform better in movement recognition. Night vision capable cameras make a perfect fit. Electrical fences may be replaced with something safer.

The least researched group of animals inside the archipelago are indeed the snakes mostly in Galápagos was discussed[27]. Only 4/9 known species have had their conservation status fully assessed, and early data indicates that most of the species may have totally vanished on a few islands. Moreover, considering that the system of classification of these reptiles in the archipelago have only just lately been resolved, practically all claims of Galápagos snakes made by park rangers and resident photographic identifiers of them are false. Their idea is to offer park rangers and visitors programmes that are simple to use and that can quickly identify different species using automated detection and recognition. We created an artificial intelligence platform—application software—that can identify a species of snake from an uploaded user photograph using deep learning techniques on a database of photos of various snake species. The program for the application functions as follows: after a user uploads a picture of something like a snake into the programme, a method processes it, assigns to one of the 9 snake species, provides the category of the expected species, and informs users by giving them details on the dispersion, natural course, preservation, and derivations of the snake. Inception V2, ResNet, and VGG16 were among the (R-CNN) architectures that were successfully built for the study, with an accuracy rate of 75%. The least researched group of animals inside the archipelago are indeed the snakes mostly in Galápagos. The idea is to offer park rangers and visitors programmes that are simple to use and that can quickly identify different species using automated detection and recognition. They created an artificial intelligence platform—application software—that can identify a species of snake from an uploaded user photograph using deep learning techniques on a database of photos of various snake species.

A number of methods for seismic activity, ultrasonic, & acoustic data-based persons detection were introduced[28]. Footstep recognition and formant analysis are performed on the auditory data. When human audio is not available, the auditory data are also utilized to distinguish between humans and animals and to determine the walking rhythm of animals. Human and animal categorization is done using seismic analysis. Hey classified the objects and estimated their presence using ultrasonic data. Using all three system methods, the authors got a high proportion of accurate categorization. Each algorithm made an effort to identify and categorise persons using the specific phenomenology of the detector. The given methods are computationally effective, use less energy, and are therefore suitable for implementation on sensor nodes like networked UGS.

Incorrect snake identification is the major cause of human deaths resulting from snake bites was discussed[29]. For the two main types of snakes, Elapidae & Viperidae, no automated classification approach has yet been presented to classify snakes by understanding the taxonomic aspects of snakes. So, they provided an automated categorization system for many snakes which is centred on inter-feature product similarity fusion identified 31 various elements from snake photos that are taxonomically significant for automatic detection research. Through a GPU-enabled parallel computing system, the classifier's adaptability and real-time development are examined. The designed systems are used in snake count management, bite analysis, and investigations of wild animals.

Snake Bites are a major neglected problem and snake bites can cause untreatable diseases and sometimes death was discussed[30]. So, to solve this problem, the authors proposed a snake detection system which contains a camera, pivot and IR rays and DC motor. The camera monitors the environment by rotating 360 degrees. If any snake is detected it triggers the alarm and laser light. The laser light will point out the location of the snake. The snake movements will be shared with the other gadgets such that the repellent effect will be made possible. The authors thought that by this way the number of snake bites can be reduced.

2.1 : EXISTING SYSTEM

2.1.1 : Snake repellents

Snakes are the animals which are present widely in the agricultural fields and in home environments which can cause a hazard in many ways. Therefore, it is crucial to adhere to the precautions taken in the places where people live to prevent them. There are a few steps that may be taken to scare off the snakes and stop them from returning, even though it is very difficult to totally eradicate them from a property.One of the least expensive ways to stay away from snakes is to use repellents. Effectively, they can be used for home reasons. They are often divided into two categories: those for household use and those for commercial applications.

While there are options available, no snake repellent can be 100% effective. Garlic spray, clove and cinnamon oil, lemon grass, and Guinea fowl are homemade snake repellents. Commercial products such as Ortho Snake-B-Gon and Victor VP364B Snake-A-Way Snake Repelling Granules can be sprinkled around a yard to deter snakes.

Ortho Snake-B-Gon: This is one of the repellents which is used widely around the world.It consists of tiny granules which should be sprinkled around the fields or the households. The formulation of the granules help in preventing the snakes from entering the property.

Pufado Snake Repellent for Outdoors: This is one of the techniques derived from the ancestors who used essential oils to keep snakes and insects away from the homes. Here the Pufado Snake balls are used which contain the essential oils in them. These balls are spread around the property in groups which create an intense scent which will repel the snakes from coming closer to the fields.

Fig 2.1: Commercial Snake Repellent

Figure 2.1 shows the commercial snake repellent which is used to stop snakes entering into the farm fields. Snake repellents are affordable but they are dangerous to the other animals if they are eaten. Snake repellents stop all the snakes entering into the field which increases the rats count in the farm fields.

2.1.2 : Ultrasonic Pest Repellent: Unlike the other snake repellents here the usage of granules is not involved.The Pest Control Ultrasonic pest repellents use electricity It produces "bionic, electromagnetic, and ultrasonic waves" that disorient the snakes and other pests. This device makes inaudible noises that are harmless to both humans and animals. All that is required is for the device to be plugged into a standard power outlet and turned on. The ultrasonic waves cover an area of up to 1,200 square feet.

Fig 2.2: Ultrasonic Pest Repellent

Figure 2.2 shows the ultrasonic pest repellent which produces ultrasonic waves to repel snakes. To build an ultrasonic pest repellent system around a 1 acre land, it may cost approximately 30,000 rupees which is not an affordable price.

Some reasons why this system is not so good are given below:

- Limited effectiveness: Ultrasonic pest repellents have limited effectiveness in deterring snakes. Snakes can become accustomed to the sound or smell of these repellents and may eventually ignore them.
- Environmental impact: Repellents like Pufado can have negative environmental impacts. The chemicals used in these repellents can pollute the soil and water, potentially harming other wildlife in the area.
- High cost: Repellents can be expensive, especially when compared to the relatively low cost of implementing an IoT and deep learning-based system.
- Continuous monitoring: Repellents require continuous reapplication, while an IoT system can continuously monitor the area without the need for human intervention.

The snake bites can be very dangerous needing immediate medical attention. Due to the limitations of these repellents, we need some different system to handle this gracefully. Let us understand how dangerous snake bites are for humans.

Snake Species	Venomosity	Death time due to snake bite	Organs affected/Harmful effects
King Cobra	Venomous	30 to 60 minutes	Affects the cardiovascular system, respiratory system, and nervous system.
Russell's Viper	Venomous	10 to 15 minutes	Affects blood vessels, kidneys, and clots blood.
Rock Python	Non-venomous	-	Severe suffocation, swelling and fever.
Common Krait	Venomous	2 to 4 hours	Causes paralysis and respiratory failure.
Saw-scaled Viper	Venomous	Around 75 minutes	Cardiovascular system, respiratory system, and nervous system.
Rat snake	Non-venomous	-	Causes pain and swelling with infections.

Table 2.1: Snakes and Harmful Effects on Humans

From table 2.3 we can understand the severity of snake bites and their effects on humans. It is crucial to study them as they can result in various health complications, ranging from mild symptoms such as pain and swelling to severe cases that may lead to paralysis, organ failure, or even death. For instance, the bites of King Cobras and Russell's Vipers can be fatal if not treated promptly with the appropriate antivenom.The severity of the snakes are so high that the time the affected may survive can be as low as 15 minutes to almost 4 hours. And even if the person survives after a proper treatment there would be serious damages to the respiratory system, nervous system and can also affect the blood vessels. Therefore, the development of an efficient and reliable system for snake detection in agricultural fields, such as the proposed IoT and deep learning-based system, can help prevent and mitigate the risk of snake bites, saving lives and reducing the impact of snake bites on human health.

Using IoT sensors and deep learning for snake detection can provide a more accurate and reliable solution. The use of advanced image recognition algorithms can accurately identify the presence of snakes in real-time, even in low light conditions or when only parts of the snake are visible. Furthermore, the implementation of IoT sensors can provide real-time monitoring of the environment and trigger immediate alerts to farmers, allowing them to take timely action and prevent damage or harm caused by snakes.

CHAPTER 3: SYSTEM REQUIREMENTS ANALYSIS

3.1 : Functional requirements: These requirements are a set of all the goals that the system has to achieve after completion. They include the features that are going to be present in the system, its capabilities, and other functions that can be performed. They list the desired outputs of the system after it is put to work.

- The system should be able to detect the toy snake on the physical model in real time using image data and sensors. Lightweight and accurate transfer learning models should be used for classification.
- The system should be able to send an alert with a photograph of the snake to the user, who is the farmer, after it detects the snake in the field (model).
- The system should be able to send an email as an alert to the corresponding users.
- The system should be able to work in real-time.
- The system should be able to work in all kinds of lighting conditions.
- The system should work with IoT devices without a hassle.

3.2 : Non-Functional requirements: These requirements are a set of all the goals that describe the qualities of the system. They talk about the system's performance, reliability, scalability, maintenance, and user experience. They describe the behaviour of the system in detail.

- The system should have good performance: low latency and response time combined with high speed.
- The system's predictions should be very accurate, with a minimal number of false positives or negatives.
- The system should be scalable easily when more sensors are added to the farm field.
- The system should be reliable with no downtime whatsoever.
- The system's maintenance cost should be low, and the components should be replaceable or repairable easily.
- The system should be able to work with other software and hardware with minimal tweaks.
- The system should be user-friendly.

CHAPTER 4: SYSTEM DESIGN

4.1 UML Diagrams

A diagram or a picture helps in understanding any complex system easily. Any system's general understanding can be quickly stored in the brain and linked to any other actions that need to be carried out using the pictorial representations. The Unified Modeling Language (UML) is used in software engineering to visually portray a system. It is used to display and portray the software system's artefacts.. It is used to visualise the workflow of the system and can be directly used to interrelate it to the object oriented languages.They help in understanding the requirements of the project, produce the designs and communicate with the stakeholders clearly.

To represent a large software system one or two diagrams may not be adequate. Therefore, there are various diagrams that describe the system in a unique way and add something to the information they provide.There are two broad categories in the UML diagrams like the structural diagrams which gives the static view of the system and the behavioural diagrams which represents the dynamic perspective of the system.Some of the structural diagrams are class diagram, object diagram, component diagram, etc. And the behavioural diagrams are the use case diagram, sequence diagram,activity diagram, etc.Though Unified Modeling Language is not a programming language , there are few tools which can be used to directly produce other programming languages which makes it easy to convert it into the object oriented paradigm.

The UML aims to visualise five distinct viewpoints using various modelling diagrams. These five viewpoints are: the user's viewpoint; structural; behavioural; environmental; and implementation.

1) **User’s View:** This contains the diagrams which represent the interaction of the user with software. Here the internal operations are not at all mentioned. The requirements of the stakeholders and the designs regarding the interfaces are given utmost importance.
2) **Structural View:** This reveals the software's static structure. Although it lists the functionalities that are present in the model, it doesn't describe the actions that those functionalities take. This provides a rough idea of the model's structure.

3) **Behavioural View:** The behavioural view contains the diagrams which show the behaviour of the software. They give us information about the relations between various components and the interaction between them. The workflow of the software can also be understood through these diagrams.

4) **Environmental View:** This view explains the behaviour of the system after the deployment. It gives us the information about the software effects, user interactions and the deployment behaviour at various conditions.

5) **Implementation View:** This consists of all the implementation parts of the system. The internal operations happening in the system are represented here which can be monitored easily if any changes occur.This view is mainly related to the developer of the software.

4.1.1 UseCase Diagram:

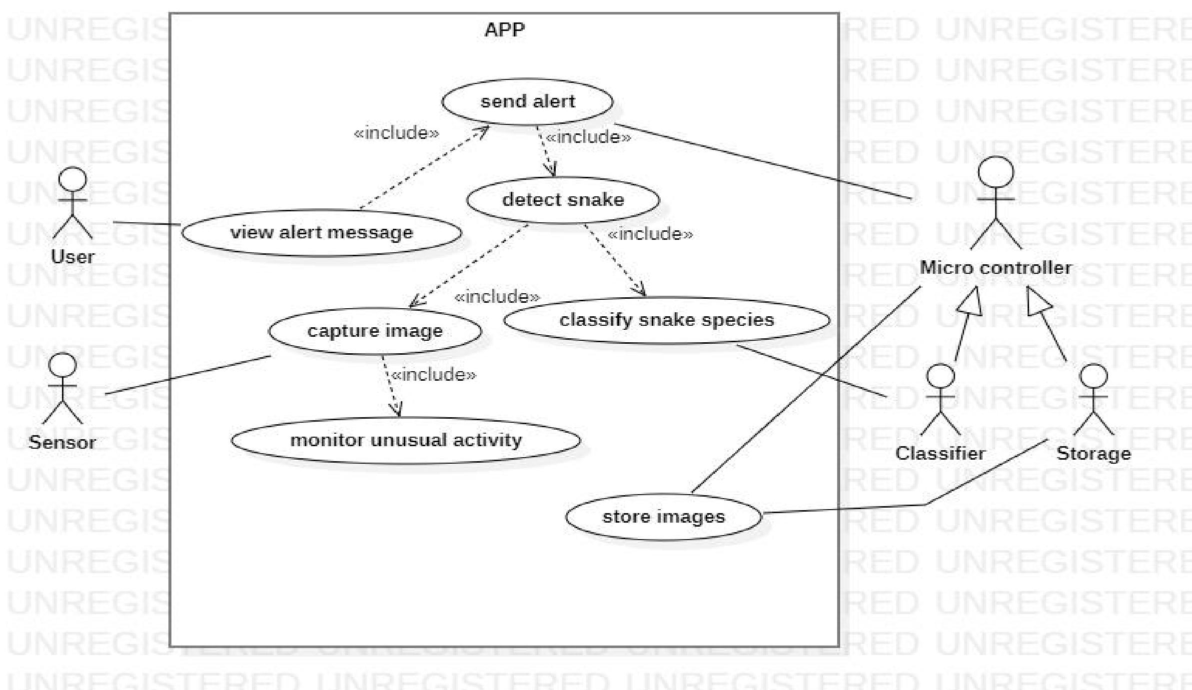

Fig 4.1: Use Case Diagram

Figure 4.1 represents the users and the interaction with the system. The main user here is the farmer who can view the alert message through his mobile. And the microcontroller consists of the classifier which enables it to detect the snake in the image captured by the camera when the sensors are triggered.

4.1.2 Class Diagram:

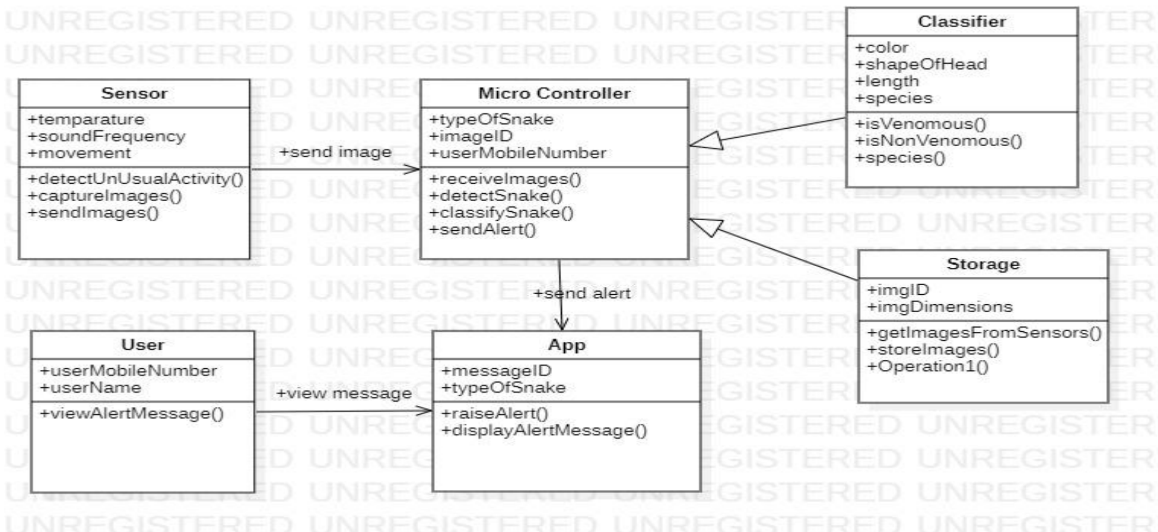

Fig 4.2: Class Diagram

Figure 4.2 gives us the overall view of the system. The classes present in the system are the sensors, micro controller, classifier, storage and the notification system.The operations of the classes tell us the responsibilities and the functionalities.The sensor has the responsibility to detect the unusual activity and trigger the camera to capture the image, the micro controller handles the sensors, camera and the classifier.The relations between the classes are used to understand the connectivity and the flow of the system.

4.1.3 Sequence Diagram:

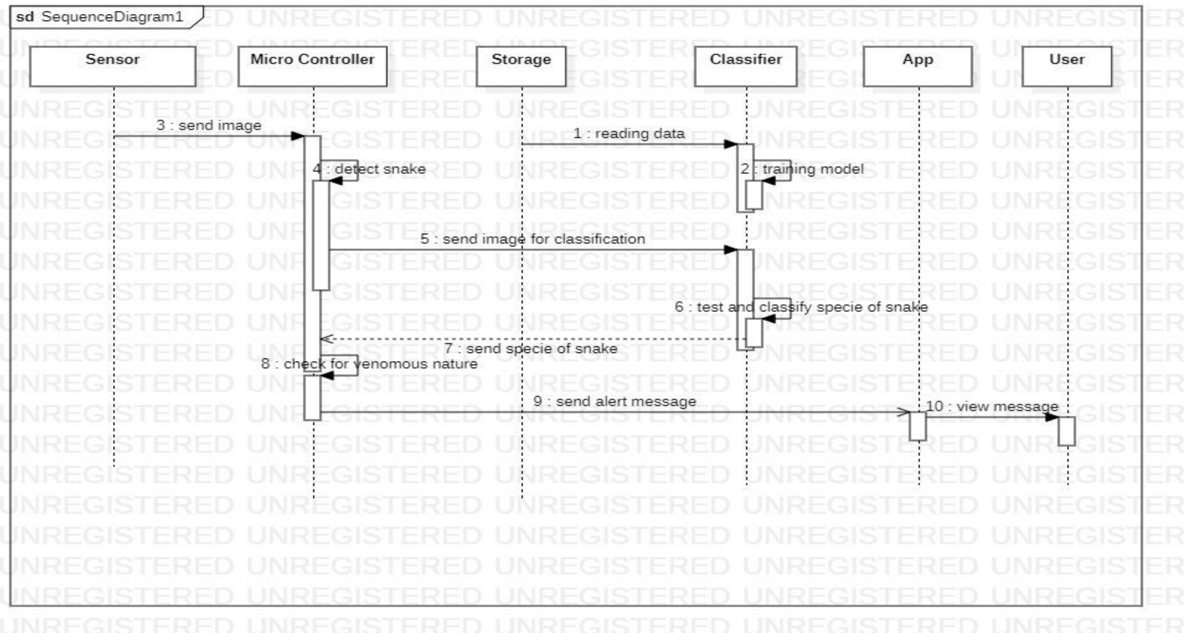

Fig 4.3: Sequence Diagram

Figure 4.3 gives the interaction between the objects and their lifelines. The objects present here are the sensors, micro controller, the storage which consists of the datasets, the classifier, the notification system. Here, it is possible to deduce the range of each item and the functions of each component. Based on the scope of each object we can calculate the production costs, computational power required and many others.

4.1.4 Activity Diagram:

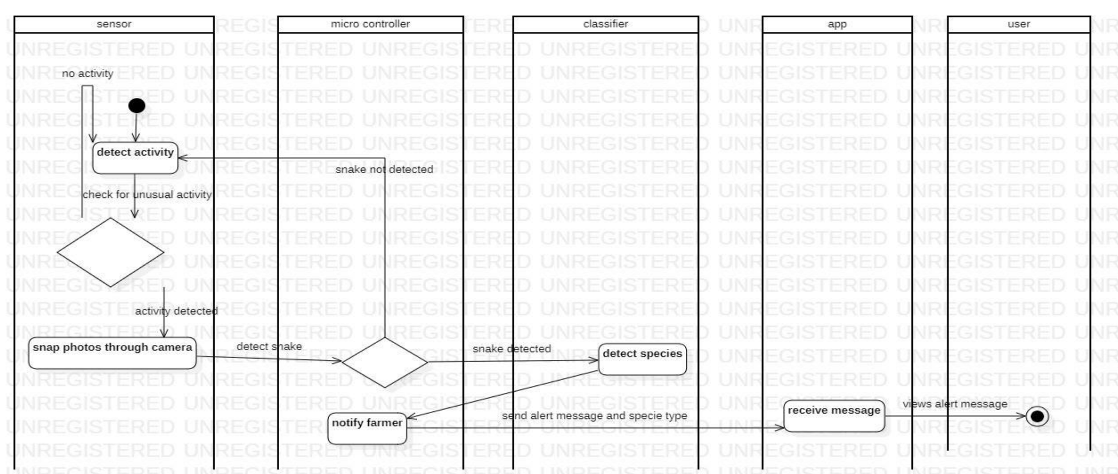

Fig 4.4: Activity Diagram

Figure 4.4 is used to understand the dynamic behaviour of the system.It represents the flow from one activity to another . The activity of each object is represented here and the interaction flow with the other objects is also shown which helps in modelling the system easily.The sensor is the first object interacting in the agricultural field which in turn triggers the microcontroller. Then the classifier comes into the action and the user can get the output based on its result.

CHAPTER 5: PROPOSED SYSTEM

Following design is used as an architectural representation of the proposed system.

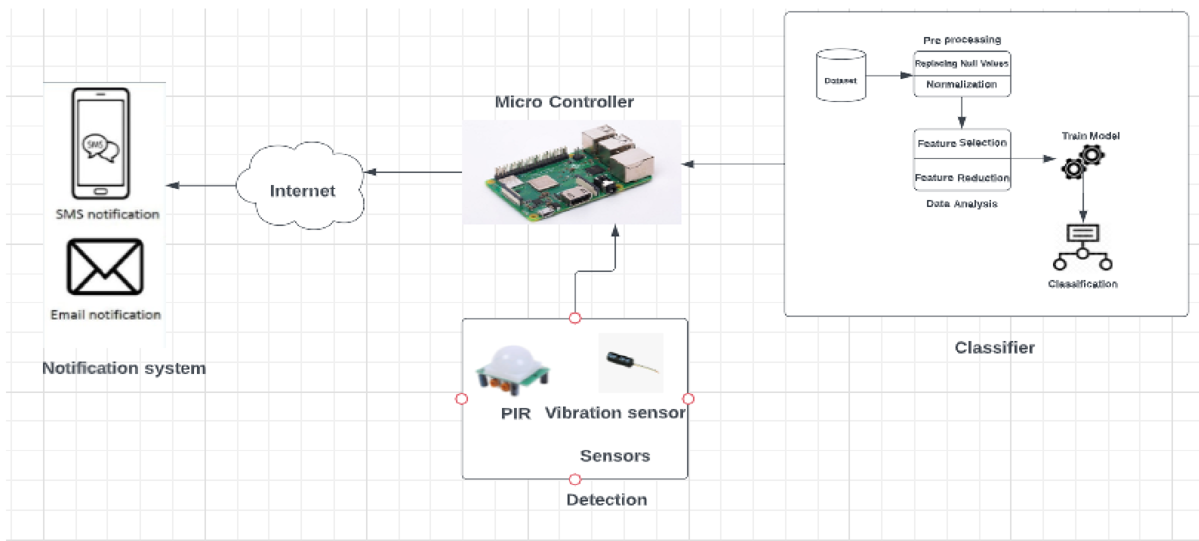

Fig 5.1: Architecture Diagram

The system consists of 4 main components. They are a detection system, microcontroller, classifier and notification system.

Detection Module:

The detection system consists of IOT sensors like passive infrared sensor (PIR) and a vibration sensor which are used to monitor any unusual activity that occurs in the agricultural field.These sensors can be placed around the fences of the fields or throughout the field. If any unusual activity is detected in the field (like any animal entering the surroundings) then the camera is triggered which will take the snapshot of the fields.The sensors assist in lowering the power consumption that the camera typically uses when used to continuously monitor the field.

Microcontroller:

The microcontroller executes the program provided by the developers. Here it is used to handle the sensors and the camera. The camera should be called when the sensors are activated in order to take a picture of the field, which is done by the microcontroller. After taking the snapshot the image is sent to the classifier by the microcontroller which tells us if any snake is present in the field.The microcontroller used here is the ATMEGA328P which is of 8-bit and works based on the RISC architecture.

Classifier Module:

The classifier is used to determine whether or not there are snakes in an image that was captured by the camera. With the aid of the supplied datasets, it uses pre-trained deep learning models like MobileNetV2, EfficientNetB1, EfficientNetB7, Resnet50 to assist in classifying the image. The classifier uses the image as input and performs a number of computational operations to determine the presence of snakes. Based on the clarity of the image and the range of the dataset the performance of the classifier varies. The accuracy and loss are carefully watched by tweaking various hyperparameters like batch size, epochs, etc., and the suitable model is chosen based on the requirement.

Notification Module:

The notification system is the final component of the system which is used to alert the people if any snake is detected. Based on the output of the classifier the notification system is triggered. If a snake is detected by the classifier then an alert to the registered people is sent through the mail with an image of the field. If the snake is not detected the notification system is not involved and the processing starts from the beginning again.

CHAPTER 6: IMPLEMENTATION

Snake detection in agricultural fields in this project consists of various steps. They are collecting images, building, training, and testing the model, and sending alert messages to the user.

6.1: Collecting Images:

As there is no proper data available for this project, we built the data by simulating the agricultural fields through a model. The snapshots of this model are taken with a camera and used as the data. Here two datasets are used, one with snakes present in the agricultural fields known as the "Snake detected dataset" and one where there are no snakes known as the "Snake not detected dataset". Through various augmentation techniques, the size of the datasets has been increased so that it helps train the model effectively.

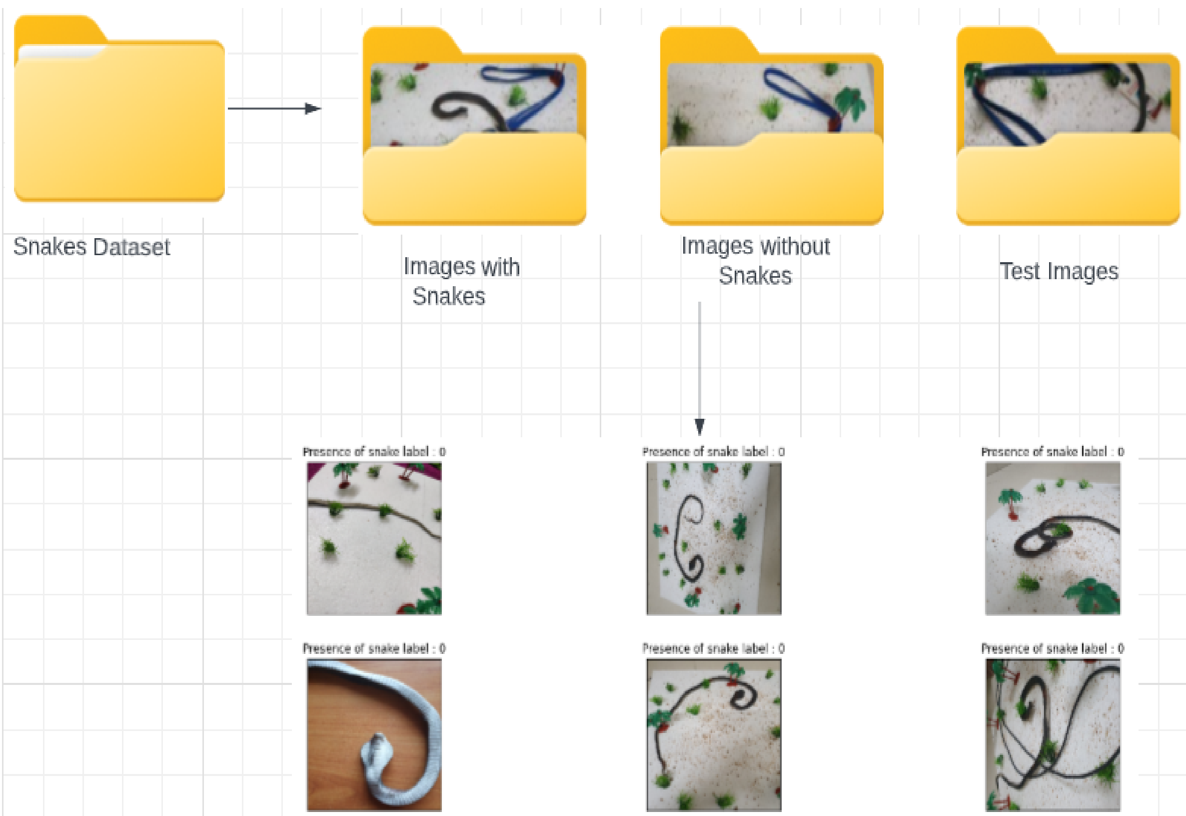

Fig 6.1: Dataset Collection

Data Loading:

2) Data loading

```
main_folder = "Snakes Dataset"
RANDOM_SEED = 6

categories = os.listdir(main_folder)
try:
    categories.remove(".DS_Store")
except:
    pass
print(categories, len(categories))
```

```
['Images with snakes', 'Images without snakes'] 2
```

```
TOTAL_CATEGORIES = len(categories)
IMAGE_SIZE = (224,224,3)

def load_images_labels(categories):
    img_lst=[]
    labels=[]
    for index, category in enumerate(categories):
        print(index, category)
        for image_name in tqdm(os.listdir(main_folder+"/"+category)):
            file_ext = image_name.split(".")[-1]
            if (file_ext.lower() == "jpg") or (file_ext.lower() == "jpeg") or (file_ext.lower() == "png") or (file_ext.lower() == "bmp"):
              try:
                    imgname = main_folder+"/"+category+"/"+image_name
                    imgname = imgname.replace("//", "/")
                    img = cv2.imread(imgname)
                    img = cv2.resize(img, (IMAGE_SIZE[0], IMAGE_SIZE[1]))
                    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
img_rotated_90 = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE)
img_rotated_180 = cv2.rotate(img, cv2.ROTATE_180)
img_rotated_270 = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
img_flip_ver = cv2.flip(img, 0)
img_flip_hor = cv2.flip(img, 1)
#img_cropped = img[50:200, 50:200]  # Example crop, adjust values as needed

img_array = Image.fromarray(img, 'RGB')
img_lst.append(np.array(img_array))
labels.append(index)

img_array = Image.fromarray(img_rotated_90, 'RGB')
img_lst.append(np.array(img_array))
labels.append(index)

img_array = Image.fromarray(img_rotated_180, 'RGB')
img_lst.append(np.array(img_array))
labels.append(index)

img_array = Image.fromarray(img_rotated_270, 'RGB')
img_lst.append(np.array(img_array))
labels.append(index)

img_array = Image.fromarray(img_flip_ver, 'RGB')
img_lst.append(np.array(img_array))
labels.append(index)

img_array = Image.fromarray(img_flip_hor, 'RGB')
img_lst.append(np.array(img_array))
labels.append(index)
```

```
            except:
                print(imgname)
                print(img)


    return img_lst, labels

images, labels = load_images_labels(categories)
print()
print("Total Images Loaded:", len(images))
```

```
  0%|                                                        | 0/244 [00:00<?, ?it/s]
0 Images with snakes
100%|████████████████████████████████████████| 244/244 [01:18<00:00,  3.09it/s]
  0%|                                                        | 0/243 [00:00<?, ?it/s]
1 Images without snakes
100%|████████████████████████████████████████| 243/243 [00:55<00:00,  4.36it/s]
Total Images Loaded: 2922
```

Fig 6.2: Data Loading

The figure 6.2 shows how the data is pre-processed and loaded which will be used later. The categories are given labels such as 0 or 1. Data augmentation is performed on these pictures.

6.2: Building Model using Transfer Learning with MobileNet, ResNet and EfficientNet:

We have used MobileNetV2, ResNet50, EfficientNetB1, and EfficientNetB7 architectures for training and building our model. MobileNetV2 is very efficient for mobile devices where computational resources are limited and high speed is required. This model was chosen for this project as it is lightweight, efficient, and very quick. It is not complex like the ResNet and EfficientNet models, which also take up a lot of storage and time. Producing high accuracy while still being fast and efficient on small datasets is one of the strengths of MobileNetV2, and it thus seemed like a great option. So, we have taken the MobileNetV2 model, which is pre-trained on millions of images from the ImageNet' dataset, and adapted it to this task using the new dataset that was collected manually for this particular task.

To build the new model, which is adapted and optimised for this project, we have removed a few final layers and added our own layers to it for the task at hand. The pre-trained layers are frozen to save time and resources A flatten layer, a dropout layer, and a final dense layer with a softmax activation function were added to the model. The 'softmax' function is most commonly used in the last layer in transfer learning techniques. It is used particularly in the last layer to normalise the output values to a probability distribution over all classes existing in the dataset. So, the softmax activation function produces a probability distribution for all the classes involved, which can then be used to know which class has a higher probability, which is the predicted class/final answer.

The accuracy and loss curves were thoroughly analysed, and measures were taken to produce good fit curves to achieve a model that is consistent and has good accuracy. For this, hyper-parameter tuning was employed to analyse the curves for different sets of hyper-parameters. Hyperparameters like epochs, batch size, learning rate, dropout, and regularisation were tuned multiple times to achieve the best results.

```
[10] EPOCHS = 15
     BATCH_SIZE = 32
     LEARNING_RATE = 0.00001
     L2_RATE = 0.05
     DROPOUT_RATE = 0.4
```

```
[13] eff_net = EfficientNetB7(input_shape=IMAGE_SIZE, weights='imagenet', include_top=False)
     for layer in eff_net.layers:
         layer.trainable = False

     Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb7_notop.h5
     258076736/258076736 [==============================] - 2s 0us/step
```

```
x = Flatten()(eff_net.output)
x = Dropout(DROPOUT_RATE)(x)
predictions = Dense(TOTAL_CATEGORIES, activation='softmax', kernel_regularizer=regularizers.l2(L2_RATE))(x)
eff_net_model = Model(inputs=eff_net.input, outputs=predictions)
eff_net_model.summary()
```

Fig 6.3: Model Training 1

In figure 6.3, the hyper-parameters were decided initially and then the model was loaded with the top layer removed. Here, the EfficientNetB7 was loaded with pre-trained layers frozen. After that, flatten, dropout and dense layers are added on top of the model. As we can see, the dropout rate and l2 rate is fixed here and a summary of the model is shown. This summary is required to know whether the layers are as expected or not. The softmax activation function is used in the dense layer for the output.

```
Model: "model_1"
____________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
====================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 rescaling (Rescaling)          (None, 224, 224, 3)  0           ['input_1[0][0]']

 normalization (Normalization)  (None, 224, 224, 3)  7           ['rescaling[0][0]']

 tf.math.truediv (TFOpLambda)   (None, 224, 224, 3)  0           ['normalization[0][0]']

 stem_conv_pad (ZeroPadding2D)  (None, 225, 225, 3)  0           ['tf.math.truediv[0][0]']

 stem_conv (Conv2D)             (None, 112, 112, 64  1728        ['stem_conv_pad[0][0]']
                                )

 stem_bn (BatchNormalization)   (None, 112, 112, 64  256         ['stem_conv[0][0]']
                                )

 stem_activation (Activation)   (None, 112, 112, 64  0           ['stem_bn[0][0]']
                                )

 block1a_dwconv (DepthwiseConv2  (None, 112, 112, 64  576        ['stem_activation[0][0]']
 D)                             )
```

Fig 6.4: Model Training 2

The model summary, a comprehensive overview of the architecture and layers of the deep learning model, is displayed in Figure 6.4. The model summary provides crucial information such as the number of trainable and non-trainable parameters, layer types and shapes, and the total number of layers in the model.

```
[17] from tensorflow.keras.optimizers import Adam
     opt = Adam(learning_rate=LEARNING_RATE)
     eff_net_model.compile(loss='sparse_categorical_crossentropy', optimizer=opt, metrics=['accuracy'])

[18] early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=7, restore_best_weights=True)

     history = eff_net_model.fit(x_train, y_train, validation_data=(x_test, y_test), steps_per_epoch=x_train.shape[0]//BATCH_SIZE, epochs=EPOCHS, callbacks=[early_st

     Epoch 1/15
     73/73 [==============================] - 72s 546ms/step - loss: 0.6961 - accuracy: 0.7568 - val_loss: 0.4622 - val_accuracy: 0.8802
     Epoch 2/15
     73/73 [==============================] - 34s 459ms/step - loss: 0.4814 - accuracy: 0.8855 - val_loss: 0.3895 - val_accuracy: 0.9271
     Epoch 3/15
     73/73 [==============================] - 33s 451ms/step - loss: 0.4072 - accuracy: 0.9258 - val_loss: 0.3556 - val_accuracy: 0.9410
     Epoch 4/15
     73/73 [==============================] - 33s 449ms/step - loss: 0.3723 - accuracy: 0.9319 - val_loss: 0.3341 - val_accuracy: 0.9531
     Epoch 5/15
     73/73 [==============================] - 33s 451ms/step - loss: 0.3474 - accuracy: 0.9436 - val_loss: 0.3153 - val_accuracy: 0.9583
     Epoch 6/15
     73/73 [==============================] - 33s 451ms/step - loss: 0.3193 - accuracy: 0.9570 - val_loss: 0.3029 - val_accuracy: 0.9670
     Epoch 7/15
     73/73 [==============================] - 33s 450ms/step - loss: 0.2988 - accuracy: 0.9709 - val_loss: 0.2918 - val_accuracy: 0.9653
     Epoch 8/15
     73/73 [==============================] - 33s 453ms/step - loss: 0.2897 - accuracy: 0.9727 - val_loss: 0.2822 - val_accuracy: 0.9774
```

```
Epoch 8/15
73/73 [==============================] - 33s 453ms/step - loss: 0.2897 - accuracy: 0.9727 - val_loss: 0.2822 - val_accuracy: 0.9774
Epoch 9/15
73/73 [==============================] - 33s 451ms/step - loss: 0.2849 - accuracy: 0.9731 - val_loss: 0.2751 - val_accuracy: 0.9774
Epoch 10/15
73/73 [==============================] - 33s 450ms/step - loss: 0.2695 - accuracy: 0.9787 - val_loss: 0.2687 - val_accuracy: 0.9809
Epoch 11/15
73/73 [==============================] - 33s 451ms/step - loss: 0.2697 - accuracy: 0.9800 - val_loss: 0.2629 - val_accuracy: 0.9809
Epoch 12/15
73/73 [==============================] - 33s 451ms/step - loss: 0.2592 - accuracy: 0.9839 - val_loss: 0.2597 - val_accuracy: 0.9809
Epoch 13/15
73/73 [==============================] - 33s 450ms/step - loss: 0.2495 - accuracy: 0.9892 - val_loss: 0.2548 - val_accuracy: 0.9861
Epoch 14/15
73/73 [==============================] - 33s 450ms/step - loss: 0.2516 - accuracy: 0.9835 - val_loss: 0.2502 - val_accuracy: 0.9861
Epoch 15/15
73/73 [==============================] - 33s 451ms/step - loss: 0.2383 - accuracy: 0.9896 - val_loss: 0.2483 - val_accuracy: 0.9844
```

```
[28] loss, accuracy = eff_net_model.evaluate(x_test, y_test, batch_size=32)
     print("Efficient Net B7 Loss:", loss)
     print("Efficient Net B7 Accuracy:", accuracy*100, "%")

19/19 [==============================] - 7s 342ms/step - loss: 0.2488 - accuracy: 0.9846
Efficient Net B7 Loss: 0.24883276224136353
Efficient Net B7 Accuracy: 98.46153855323792 %
```

Fig 6.5: Model Training 3

Figure 6.5 illustrates the process of model fitting and training, including the calculation of accuracy and loss. This visualisation gives us insight into how the model is being optimised during the training process. By closely monitoring these metrics during training, adjustments can be made to hyperparameters, such as the learning rate, to improve overall model performance.

6.3 Hardware

The sensors that we have used in the project are the passive infrared sensor (PIR) and the vibration sensor. These two sensors are used to detect any unusual activity, which in turn will trigger the camera to take a snapshot. The sensors are connected to the Arduino microcontroller, which takes care of image capture and triggers the deep learning model. A program is compiled in the Arduino compiler, which deals with the process that needs to be done by the microcontroller.

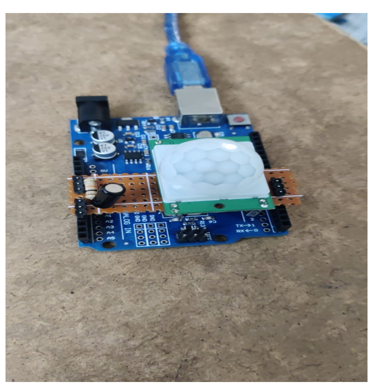

Fig 6.6: Hardware

```
#define vibPin A0
#define pirPin 7
unsigned long previousMillis = 0;
const unsigned long interval = 400;

void setup(){
  Serial.begin(9600);
  pinMode(vibPin,INPUT);
  pinMode(pirPin,INPUT);
}
void loop(){
  int vibData = (analogRead(vibPin)>500) ? 1:0;
  int pirData = digitalRead(pirPin);
unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
  Serial.print("Data,");
  Serial.print(vibData);
  Serial.print(", ");
  Serial.println(pirData);
  }


}
```

Fig 6.7: Hardware Compiling Code

Figure 6.6 shows the image of a hardware component and figure 6.7 illustrates the code for compiling the hardware. In the code, we need to define the location of the PIR and vibration sensors which are linked to the PCB board. Then we need to start taking inputs from the sensors. If the voltage of the vibration sensors is 5V then we need to consider it as vibration detected. The PIR sensor reads the logic state of a pin (0 - “no movement detected”, 1- “movement detected”).

6.4 Front End

An UI was built using HTML, with the backend being Flask and Python to use the sensors and microcontroller and display the results on the website. Live camera feeds are always displayed on the website. The website works in real time by capturing images, which are then analysed by the DL model. According to the results given, the message is displayed on the screen using the HTML DOM functions. The message can be "Snake Detected" or "No

Snake Detected".

```
<body>
    <div class="container" style="width: 40%;">
        <h1 class="Title" >Snake Detection</h1>
        <div class="video-container" style="top: 0;">
            <img id="image" src="{{ url_for('video_feed') }}" >
        </div>
        <br>
        <h1 class="output" id="out" ></h1>
    </div>
```

Fig 6.8: Index.html 1

```
</body>
<script>
    setInterval(function load(){
      var ft =fetch('/', {
            method: 'POST', // or 'PUT'
            body:"post"
            })
        .then(response => response.json())
        .then(data => {
            var ar_data = data;
         console.log(ar_data)
         if(ar_data["msg"] !="" ){
            document.getElementById('out').innerHTML =ar_data["msg"];
            document.getElementById("image").style = "width:30%;"
            setTimeout(() => {
            document.getElementById("image").src="";
            document.getElementById("image").src="{{ url_for('video_feed') }}";
            document.getElementById('out').innerHTML ="";
            document.getElementById("image").style = "width:100%;"
          },3000)
          fetch("/clear");
         }

        })
        .catch((error) => {
            console.error('Error:', error);
        });
    }
, 1000);
</script>
</html>
```

Fig 6.9: Index.html 2

Figure 6.8 and 6.9 illustrate the webpage of our project. It gives the user interface to the farmer where the live-feed of the agricultural field is shown and the output of the classifier is displayed. The images that are captured by the camera are sent to the classifier for further computation which is present in the backend.

```python
from flask import Flask, jsonify,render_template,request,Response,redirect,flash
import threading
import os
# from werkzeug.utils import secure_filename
import serial as sr
# from mailsend import sendmail


import cv2
import numpy as np
from PIL import Image
#import systemcheck

# import the necessary packages
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
from tensorflow.keras.models import load_model


IMAGE_SIZE = (224, 224, 3)
CATEGORIES = {0:'Snake detected',1:'No snake detected'}

model = load_model("Snake_detection_mobilenet_model_8.h5")

#########################################################################################
def model_warmup():
    dummy_image = []
    for i in range(224):
        dummy_image.append([[0]*3]*224)
    image = np.array(dummy_image)
    # print(image.shape)
    image = np.expand_dims(image, axis=0)
    pred = model.predict(image)
    # print(pred)
#########################################################################################
def predict_snake(img):
    img = cv2.resize(img, (IMAGE_SIZE[0], IMAGE_SIZE[1]))
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

Fig 6.10: App.py 1

The app.py script in figure 6.10 is used to take the sensor readings in the form of 0s and 1s, which indicate whether the sensor detects any activity. A PIR reading of 1 indicates movement was detected, and a vibration sensor value of 0 indicates vibration was detected. These values are analysed to see whether the image needs to be captured by the camera to send it for prediction. If the prediction is positive, an email is sent with the image captured, indicating that the snake is present.

```python
def predict_snake(img):
    img = cv2.resize(img, (IMAGE_SIZE[0], IMAGE_SIZE[1]))
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    # print(img.shape)
    img_rotated_90 = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE)
    img_rotated_180 = cv2.rotate(img, cv2.ROTATE_180)
    img_rotated_270 = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
    img_flip_ver = cv2.flip(img, 0)
    img_flip_hor = cv2.flip(img, 1)
    images = []
    images.append(img)
    images.append(img_rotated_90)
    images.append(img_rotated_180)
    images.append(img_rotated_270)
    images.append(img_flip_ver)
    images.append(img_flip_hor)
    images = np.array(images)
    images = images.astype(np.float32)
    images /= 255
    op = []
    # make predictions on the input image
    for im in images:
        image = np.array(im)
        image = np.expand_dims(image, axis=0)
        pred = model.predict(image)
        pred = pred.argmax(axis=1)[0]
        op.append(pred)
        # print("Pred:", pred, CATEGORIES[pred])
    op = np.array(op)
    print("Final Output:", CATEGORIES[np.bincount(op).argmax()])

    # if CATEGORIES[np.bincount(op).argmax()] == 0:
    #     sendmail("Snake in your farm")


    return CATEGORIES[np.bincount(op).argmax()]

model_warmup()
```

Fig 6.11: App.py 2

In figure 6.11, the predicted snakes function is shown which is used for prediction of the image captured and sent. Pre-processing of the image is done just like in the training of the model. After that the augmented images are given for prediction and then max predicted result is shown to the user on the screen.

```python
DB = {
    "msg":"",
    "flag" : 0
}

#####################################################################################
def checkPort():
    for i in range(0,150):
        try:
            ser = sr.Serial(f'COM{i}',9600,timeout=1)
            print(f'COM{i}')
            return ser
        except Exception as e:
            pass
#####################################################################################

def getData():
    global DB
    port = checkPort()
    while 1:
        data = port.readline().decode("utf-8")
        # print(data)
        if len(data) > 0:
            try:
                data = data.split(",")
                # print(data)
                if int(data[1]) == 0 and int(data[2]) == 1 and DB["flag"] == 0:
                    DB["flag"] = 1
                # update(d)
            except Exception as e:
                print(e)

#####################################################################################

app = Flask(__name__)
video = cv2.VideoCapture(0)## 0 for internal  : 1 for external
```

Fig 6.12: App.py 3

Check port and getData functions are shown here in figure 6.12. We have to first check if the connected sensors are having problems or not. That is done in the check port function. If there is any unusual activity, the getData function reads the data given by the sensors and stores it. As shown, this function runs all the time when the application is being executed.

```python
@app.route('/',methods=["get","post"])
def server_app():
    global DB
    if request.method=="POST":
        return DB
    return render_template("index.html")
##########################################################################################
@app.route('/clear',methods=["get","post"])
def Clear():
    global DB
    DB["flag"] = 0
    DB["msg"] = ""
    return "OK"
##########################################################################################
def gen(video):
    global DB
    cam=cv2.VideoCapture(0)## 0 for internal  : 1 for external
    cam.set(3,1080)
    cam.set(4,1080)
    while True:
        try:
            _,image=cam.read()
            image = cv2.flip(image, 1)
            # DB["msg"] = predict_snake(image)
            if DB["flag"] == 1:
                temp=predict_snake(image)
                print(temp)
                DB["flag"] = 2
                DB["msg"] = predict_snake(image)
                print("yes",DB["msg"])
                cv2.imwrite("img.png",image)
                if(CATEGORIES[0] == DB["msg"]):
                    os.system("python mailsend.py")
            # print()
            ret, jpeg = cv2.imencode('.jpg', image)
            frame = jpeg.tobytes()
            yield (b'--frame\r\n'
                b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
```

Fig 6.13: App.py 4

The figure 6.13 shows the generator function which gives the live feed of the camera on the website. After detecting unusual activity, this function also shows the captured image on the screen along with the prediction result. The mail is also sent after this by triggering mailsend.py script.

```python
        except Exception as e:
            print("Error : ",e)
##########################################################################################
@app.route('/video_feed')
def video_feed():
    global video
    return Response(gen(video),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

kwargs = {'host': '0.0.0.0', 'port': 8080, 'threaded': True, 'use_reloader': False, 'debug': True}
if __name__ == '__main__':
    threading.Thread(target=app.run, kwargs=kwargs).start()
    threading.Thread(target=getData).start()
    #app.run(port=6050)
```

Fig 6.14: App.py 5

Figure 6.14 shows the routing information.

```python
import smtplib
import imghdr
from email.message import EmailMessage

Sender_Email = "akkalakarthik@gmail.com"
Password = "rutagksrnzehdlrs"


Reciever_Email = "tejakumarkalimera0107@gmail.com"

def sendmail(subject = "Snake DETECTED..." ):
    # return
    try:
        print("Sendimg Mail...", end = "")
        newMessage = EmailMessage()
        newMessage['Subject'] = subject
        newMessage['From'] = Sender_Email
        newMessage['To'] = Reciever_Email
        newMessage.set_content('Snake Detected...')

        with open("img.png", 'rb') as f:
            image_data = f.read()
            image_type = imghdr.what(f.name)
            image_name = f.name

        newMessage.add_attachment(image_data, maintype='image', subtype=image_type, filename=image_name)

        with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
            smtp.login(Sender_Email, Password)
            smtp.send_message(newMessage)

        print(" Done...")
    except:
        print("Failed")
if __name__ == '__main__':
    sendmail()
```

Fig 6.15: MailSend.py

The mailsend.py script is shown in figure 6.15 which is used to send a mail to the receiver with the image of the snake captured and a message saying “Snake detected”.

6.5 Implementation in Farm Fields

In this project we have simulated the working using a small thermocol model of plastic trees and toy snakes. But, in real world scenarios, we have huge farm fields and real snakes. For real snakes, we have to perform snake detection using a dataset having different snakes images from various angles. The dataset must be very large with over 8,000 images for training and over 2,000 images for testing. The sensors have to be put in strategic locations in the farm field to sense data and capture images.

The system requires many sensors which go up or down depending on the field size. These sensors should be connected to a central hub where the processing power relies. All the readings of the sensors are sent to the central hub for computation. The central hub has the deep learning model which is used for prediction. Other factors like temperature, humidity, and other environmental factors can also be sent to train to the deep learning model for accurate predictions.

As mentioned earlier, the dataset should contain various snake and non-snakes images for training. The dataset can contain images of the snakes' species which are more common in the region. Those images can be given for training based on the region. For example, the species which are very common in the state of Telangana and Andhra Pradesh are cobra, rock python and Russell's viper. Of these, rock python is non-venomous and therefore not harmful for the farmers. But the cobra and russell's viper are venomous and can be harmful for the farmers if they accidentally bite them. These species can be fed to the transfer learning model and trained after giving suitable labels. As it may be multi-class classification, the softmax activation function in the last layer can be very helpful as it calculates the probabilities of each class. The class with the highest probability is the predicted class. The pre-processing, training, and hyper-parameter tuning is mostly common like the implementation simulated but the accuracy may suffer because of unpredictable environmental conditions. So, training on a diverse dataset and reinforcement learning is very important in these situations.

The computational power can be a RaspberryPi microcontroller. The model can be deployed on this microcontroller which will then predict images using it. Other microcontrollers also can be used and cloud options can also be explored. It is the responsibility of this microcontroller to send the notification to the farmer. The notification can be an email or an SMS. For SMS, a GSM module can be used which covers a range of more than 35 kilometres. The notification can include the species and venomous nature of the snake so that, the farmer and other people can take precautions.

CHAPTER 7: TESTING

Testing is the process of assessing a system, item, or programme to ascertain its usability, effectiveness, dependability, and other important features. The purpose of testing is to find any flaws or problems that could be present in the system and to make sure it complies with the specifications and quality norms.

7.1 TEST CASES

S.no	Testcase_Name	Expected_Result	Actual_Result	Status
1.	Connecting and initialising the sensors	The sensors should be able to detect activity after connected to the system	No readings detected as system's bluetooth was on even after connection was successful	Fail
2.	Connecting, initialising and reading inputs from the sensors	The sensors should take the input from environment in the form of binary format	The sensors were reading inputs (after connecting establishment) from the environment successfully as system's bluetooth was off	Pass
3.	Capturing images	The camera should capture a images when the sensor reads a specific input	The camera captured the images.	Pass
4.	Detecting snake	Deep learning should identify the snakes in the captured images.	Snakes are detected.	Pass
5.	Sending an email	The model should send an email to the user along with the captured image.	An email is sent to the user.	Pass
6.	Sending an email requirements	Email should be sent to the receiver email address provided in the python script.	Email wasn't sent even after detecting the snake because of no internet.	Fail

7.	Detecting very quick vibrations	Should sense all the vibrations and work normally.	If the vibrations are given very quickly one after the other, then the system is not processing them at the same speed. It is taking time to read.	Fail
8.	Working in different browsers	Should work normally in any browser.	Working normally in all popular browsers like Chrome, Edge and Brave.	Pass
9.	Network connectivity	System should connect without any issue and work normally.	The system fails to connect due to poor signal strength.	Fail
10.	System response time during light load	System should work as expected.	System working normally	Pass
11.	System response time during heavy load	System should work as expected.	System sometimes taking a lot of time to process and produce the results	Fail

Table 7.1: Test Cases

Table 7.1 shows the test cases where our project is giving incorrect and correct results.

A test dataset was manually prepared for evaluation purposes. These images are completely new for the deep learning model. It is solely used to view the results of the deep learning model and understand the edge cases.

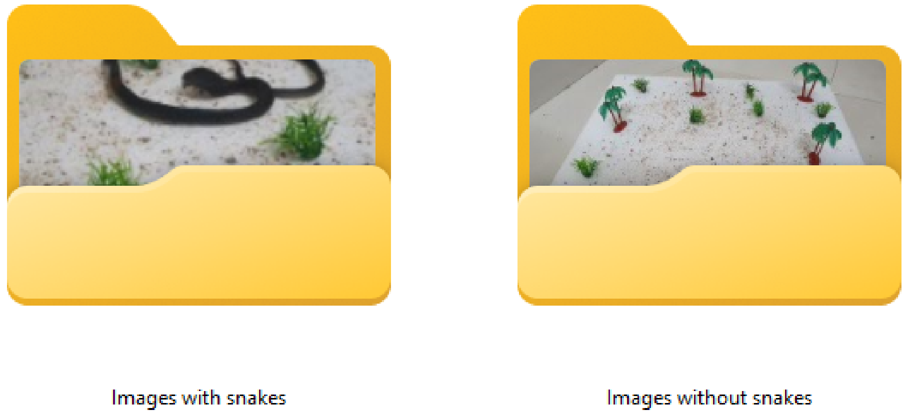

Fig 7.1: Dataset

Our model should detect snakes and non snake objects accurately. The images with snakes folder contains snake images which are captured at different angles and lengths. The images without snakes folder contains the images of different objects which look like snakes. We tested our model in the absence of any object, the results should be no snake detected.

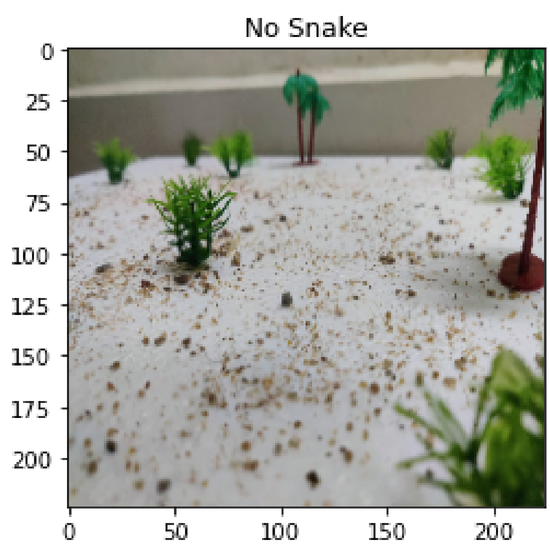

Fig 7.2: Without Any Object

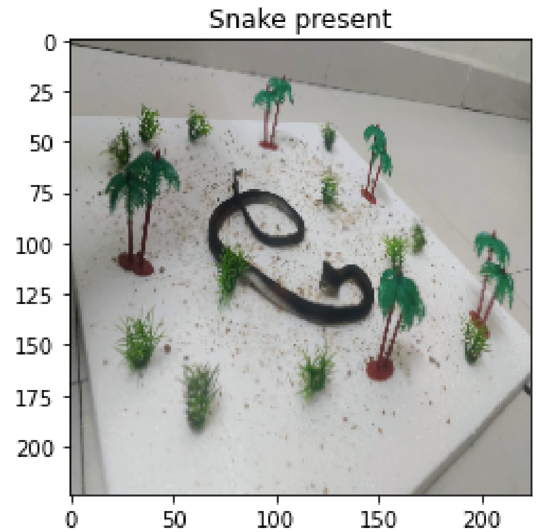

Fig 7.3: Testing in the Presence of Full Snake

The figures 7.6 & 7.7 shows, in the presence of a full snake the model tested correctly. But the model predicted it wrong in the presence of a half snake.

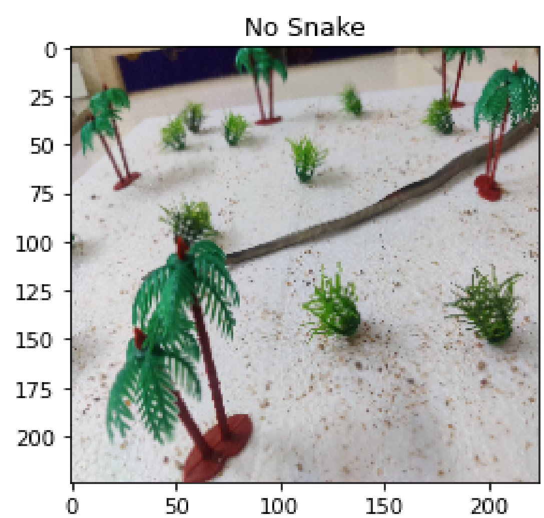

Fig 7.4: Testing in the Presence of a Snake Tail

Figure 7.8 shows, testing in the presence of a snake tail, the model tested wrong. The model also predicted wrong for some objects which look like snakes.

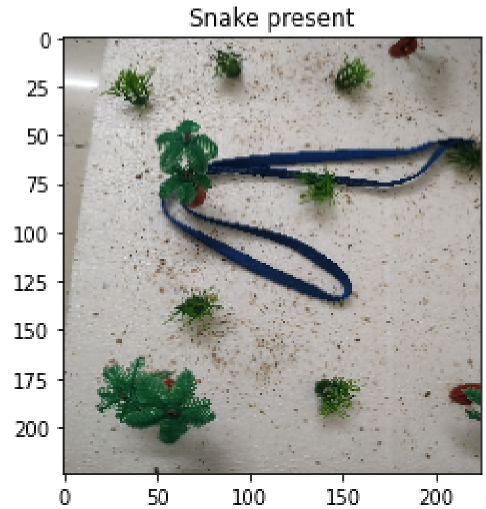

Fig 7.5: Testing in the Presence of ID Card

Figure 7.9 shows, in the presence of an ID card the model tested wrong because of the similarity between ID card and snake.

The model predicted well for several photos during testing but made some incorrect predictions for photos that had things that resembled snakes and half snakes. After altering the hyperparameters, some of those cases were overcome.

CHAPTER 8: RESULTS

The deep learning model that was built during this project has shown good accuracy and performance in real-time use cases. The hyperparameters like number of epochs, learning rate, batch size, regularisation rate, and dropout were fine-tuned as much as possible to arrive at a model that performed relatively well in all the models we built. The actual number of images that were collected for the dataset is 487. These images are only used for training the DL model. An additional 72 images were captured for testing the model and recording the results it produced. Data augmentation was used to reduce overfitting and increase the diversity in the model to some extent. After augmentation, the images used for training came out to be 2922. These images were trained on these models: ResNet50, MobileNetV2, EfficientNetB1, and EfficientNetB7.

MODEL	EPOCHS	BATCH SIZE	LEARNING RATE	DROP OUT	L2 RATE	ACCURACY	OVERFIT
RESNET50	10	16	0.001	-	-	88.3%	AVERAGE
MOBILE NETV2	10	64	0.0001	0.8	0.1	95.2%	BELOW AVERAGE
EFFICIENT NETB1	20	32	0.00001	0.4	0.05	97.7%	AVERAGE
EFFICIENT NETB7	15	32	0.00001	0.4	0.05	98.4%	MINUTE

Table 8.1: Model Comparison

From table 8.1, we can clearly see that the EfficientNetB7 was the best performing among all with only minute overfitting at the end of the training session. It also gave the lowest number of incorrect results for the test dataset. EfficientNetB1 performed well, but it suffered from overfitting. MobileNetV2 performed better than the ResNet50 model with slight overfitting. The MobileNetV2 model was producing a lower number of incorrect results than ResNet50 and EfficientNetB1 for the test dataset. It is also a lightweight model, which is very fast and occupies less storage space. When we take accuracy and the number of wrong predictions into account, the EfficientNetB7 is superior to the MobileNetV2. On the grounds of storage space, EfficientNetB7 occupied a lot of storage space (approximately 20 times that of

MobileNetV2).

MobileNetV2 model:

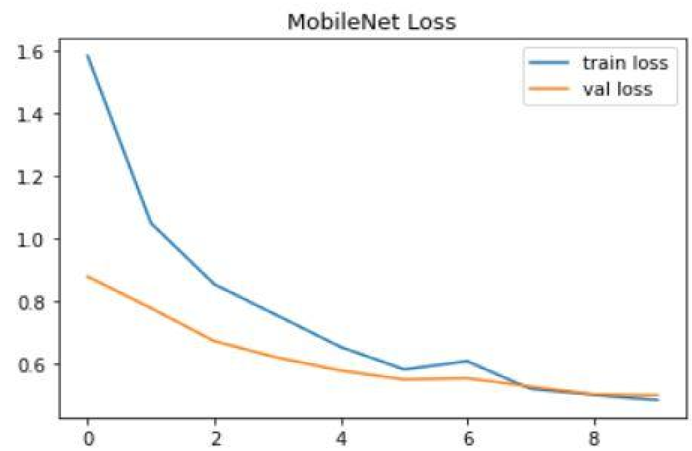

Fig 8.1: MobileNetV2 Loss

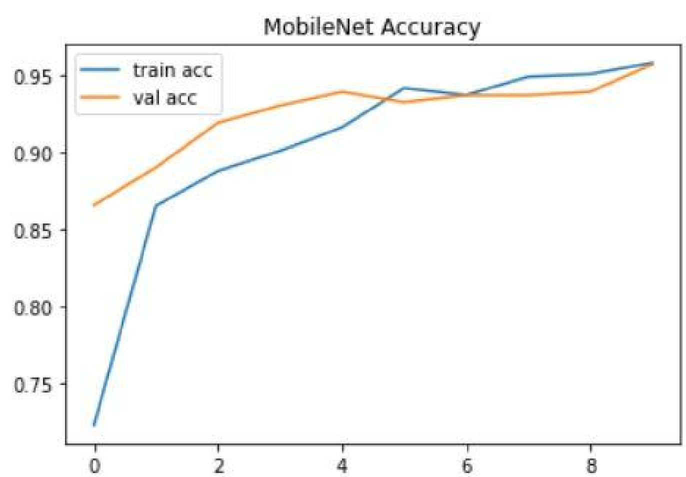

Fig 8.2: MobileNetV2 Accuracy

From figure 8.1 & 8.2, we can infer that there were some fluctuations in the train vs. value accuracy curves, and they crossed each other a few times, which indicates overfitting. But, they converged at the end, indicating that the overfitting has reduced a little as the epochs went by. In our testing on the completely new dataset, it produced very few wrong predictions, which tells us that the DL model is generally reliable on new kinds of data. On the test dataset, this MobileNetV2 model predicted four wrong images out of the total 72 images.

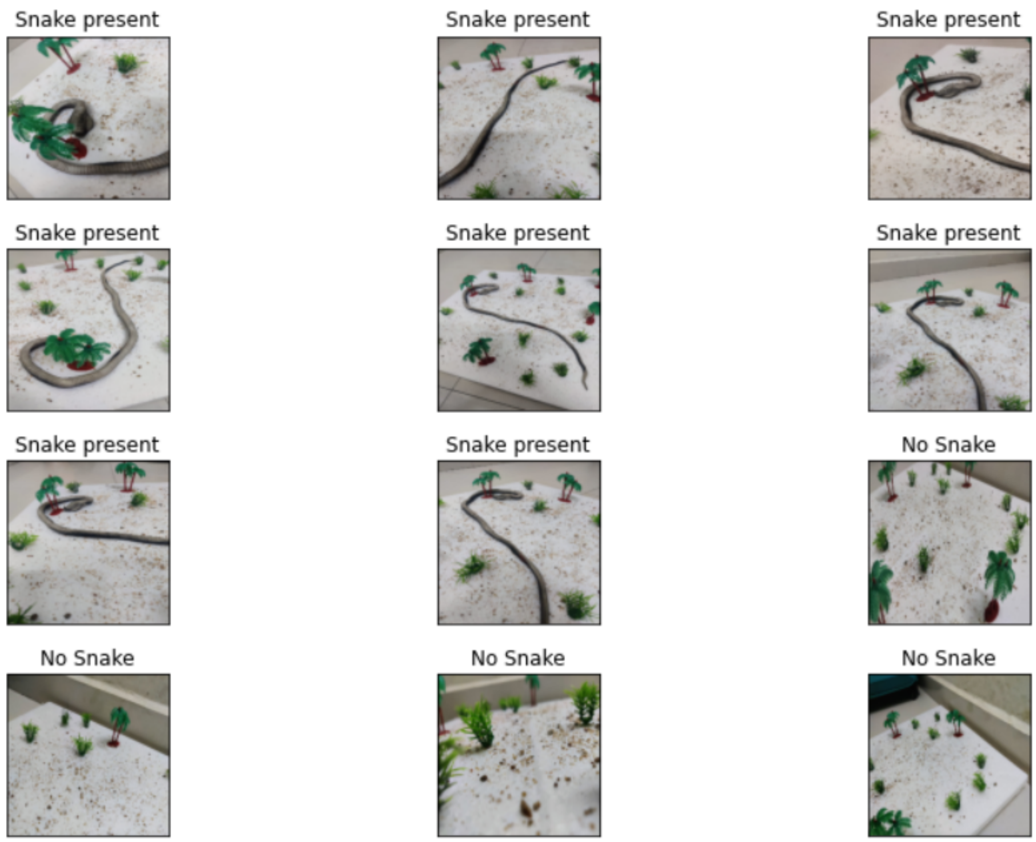

Fig 8.3: Test Images (MobileNetV2)

The results of the MobileNetV2 model on a completely new test dataset are illustrated in Figure 8.3. The title of each image indicates the prediction of the model: either “Snake present” or “No Snake”.

EfficientNetB7 model:

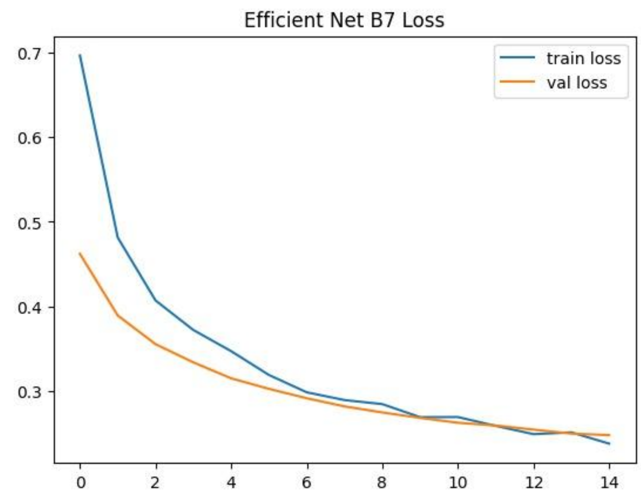

Fig 8.4: EfficientNetB7 Loss

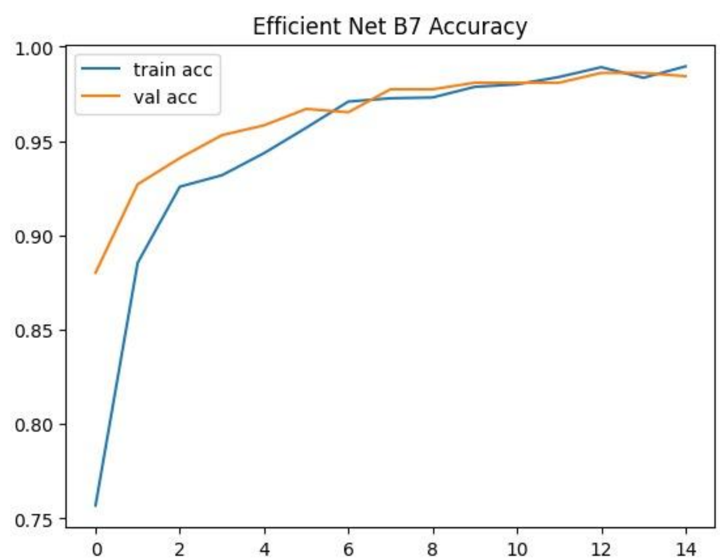

Fig 8.5: EfficientNetB7 Accuracy

From figure 8.4 & 8.5, we can see that the curves overlap for most of the epochs, which is a great sign of a good fit curve. Although the validation loss is slightly higher than the training loss at the end, which is the 15th epoch, it is not a big problem. In our testing on the completely new dataset, it produced negligible wrong predictions, which tells us that the model is very reliable on new kinds of data. On the test dataset, this model incorrectly predicted only one image out of the total 72 images.

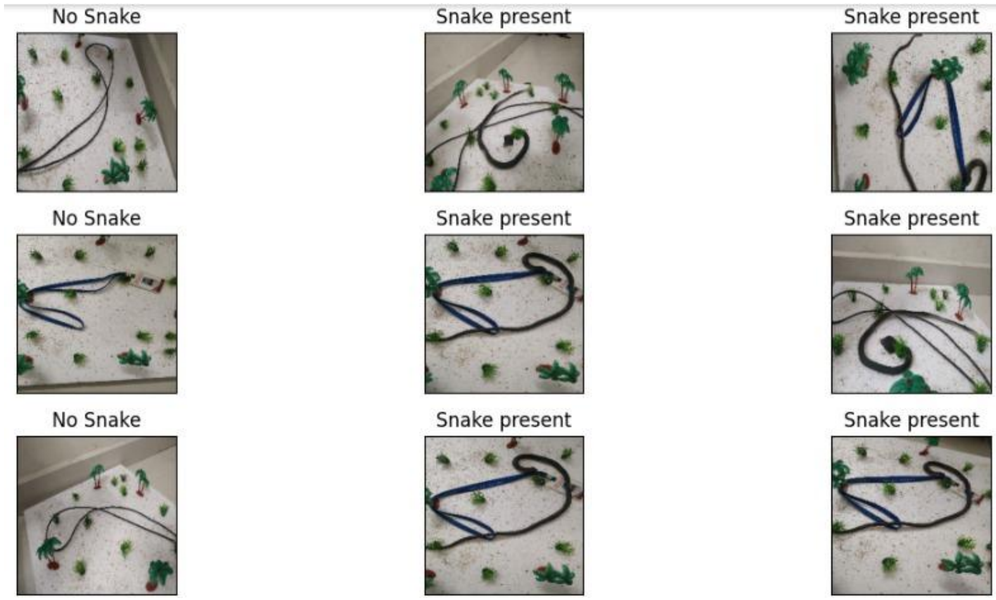

Fig 8.6: Test Images (EfficientNetB7)

Figure 8.6 depicts the prediction results of the EfficientNetB7 model on the test dataset.

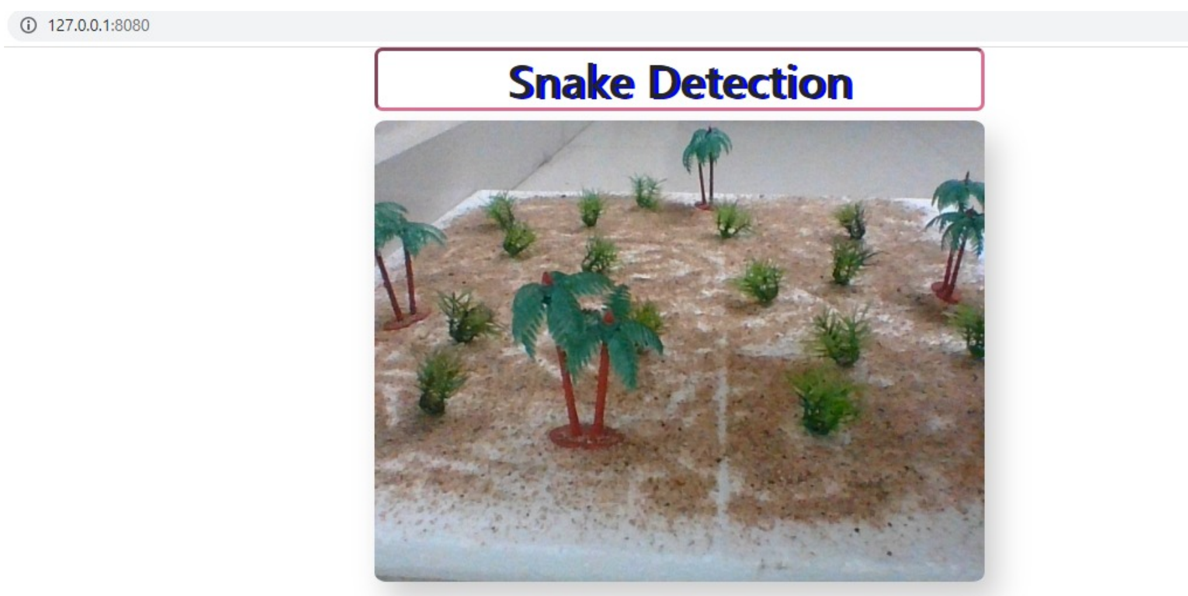

Fig 8.7: In the Website: Webcam Trying to Capture Images in Real-Time

Figure 8.7 depicts the live camera feed interface of the application, accessible through the browser, and ready to capture images.

8.8 and 8.9.

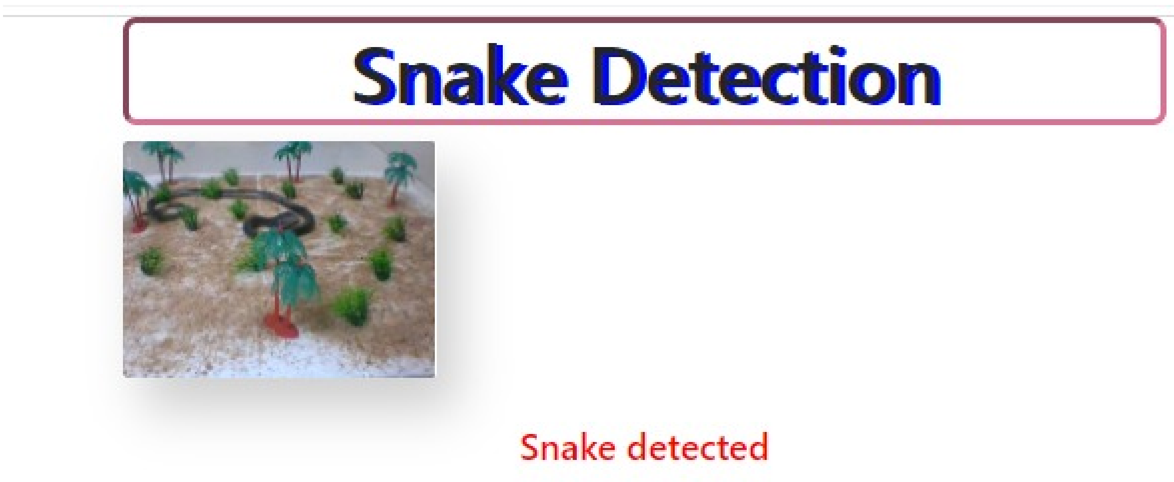

Fig 8.8: Result Prediction for the Image Captured: Snake Detected

Fig 8.9: Result Prediction for the Image Captured: No Snake Detected

Upon detecting unusual activity, the application captures an image and displays the result along with the captured image on the website for a few seconds. This can be seen in Figures 8.8 and 8.9.

Fig 8.10: Result Prediction for the Image Captured: Failure Cases (MobileNetV2)

The MobileNetV2 model showed some limitations in detecting snakes when only the tail was visible in the image, as illustrated in Figure 8.10. However, the EfficientNetB7 model was able to correctly classify such images as snakes, highlighting its superior performance in this regard.

Fig 8.11: Result Prediction for the Image Captured: Failure Cases (EfficientNetB7)

Figure 8.11 depicts a misclassification by the EfficientNetB7 model, where it incorrectly identified an ID card image as having a snake present. This was one instance out of the total 72 different images in the test dataset.

```
1/1 [==============================] - 1s 927ms/step
[[0.6252549  0.37474507]]
```

Snake present

Fig 8.12: Prediction result with Probabilities (MobileNetV2)

Figure 8.12 shows that the probability of the image being a snake is 63% and that it is not a snake is 37%. The number 0 is assigned to snakes being present in the picture and the number 1 to snakes being absent in the picture. As the probability of 0 is greater than 1 here, it is predicted that the snake is present in the image, which is correct.

CHAPTER 9: CONCLUSION AND FUTURE SCOPE

9.1: Conclusion

The main goal is to find snakes coming into the farm fields and warn people who are nearby. It was implemented using different IoT sensors and deep learning models. Farmers' lives can be saved as long as the system works with minimal delay and as long as the model is pretrained. Training was done using Resnet, MobileNet and EfficientNet models, and accuracy and loss curves, and other metrics like precision, recall, etc. were used to evaluate the models and choose what's best among them. MobilenetV2 and EfficientNetB7 were selected for their higher accuracy of over 95% compared to ResNet50's 88%. Taking all the technical constraints into consideration, the MobileNetV2 model emerges as the clear winner for this project, offering an unbeatable combination of speed and efficiency. It occupies 20 times less storage than EfficientNetB7 and has a noticeably faster training speed. Faster training speeds means that you will be up and running in no time. These benefits will be more pronounced when working with a large and diverse dataset. Hyperparameter tuning was done to increase the accuracy and make the model better. The accuracy and loss curves generated were satisfactory, and when tested, they produced only a few errors. A more effective, complex image detection model can find snakes even when the land is covered. This model sometimes produces wrong results, which can be avoided by training on a more diverse dataset with computers having higher computational power.

9.2: Future scope

There are a lot of possibilities that can be explored with the main idea of this project. Many features can be added to this project, and several other applications can be developed using the implementation and core idea of this project. Features like sending alerts to the nearby people surrounding the farm field, making predictions of snake species, sending SMS notifications to the user (the farmer), sending emergency alerts to the nearby clinics and hospitals, etc., can be integrated with this project for greater usability and better functionality.

Advanced sensors like Lidar can also be used for more range and better detection. It can create a 3D map of the environment it is placed in, which can be used to detect any unusual activity along with cameras. The 3D map is created by measuring distances using laser lights.

There are various benefits of using Lidar: it can improve accuracy drastically in real world situations, it can be very good in low light scenarios and cover extensive range with just one sensor. While Lidar has been expensive since its inception into the market, the cost has been seeing a downward trend in recent years. This Lidar technology is continuously evolving and is becoming more accessible. In future, Lidar sensors may become as affordable as any other sensor and incorporating them into the system can bring great value for money to farmers who are looking to improve and optimise their snake detection systems. Human life is invaluable and above everything in this world. With these advanced techniques, saving even one life is also a significant accomplishment. So, innovative technologies like these should be continuously explored and invested in to improve safety and protect people working in farm fields.

The deep learning models which have been trained can be deployed on the cloud and can be easily used without concern for storage or computational power. Usually, cloud systems like AWS use servers with high computational power making it a good choice for deploying DL models. So, deploying on the cloud may increase the overall performance of the system and also makes it easy to get connected from any location. This not only makes the system more efficient but also enhances its accessibility and scalability, making it a viable option for farmers of all sizes.

The core idea of this project is to detect something that can cause harm to the people in the surrounding area. We have detected snakes in this case and notified the farmer. In a generalised context, this detection system can be used for any animal, not just snakes. We can use it for tigers, monkeys, etc., which are ruining the crops in villages that are near the forests. This idea can be extended to detect intruders in the farm field too. Another use case can be in prisons, where the system can detect prisoners who try to escape the prison illegally. This is a problem in rural India, where security is not that tight. By leveraging the core idea of this project, many pressing problems could be addressed with urgency.

project

ORIGINALITY REPORT

8%	6%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Kumoh National Institute of Technology Graduate School Student Paper	2%
2	www.coursehero.com Internet Source	1%
3	Submitted to London School of Commerce Student Paper	<1%
4	dr.lib.sjp.ac.lk Internet Source	<1%
5	Submitted to VNR Vignana Jyothi Institute of Engineering and Technology Student Paper	<1%
6	"Soft Computing for Security Applications", Springer Science and Business Media LLC, 2022 Publication	<1%
7	www.jetir.org Internet Source	<1%
8	Submitted to University of Westminster Student Paper	<1%

Fig 9.1: Plagiarism Report 1

9	Submitted to CSU, San Jose State University Student Paper	<1 %
10	dspace.dtu.ac.in:8080 Internet Source	<1 %
11	Mrugendra Vasmatkar, Ishwari Zare, Prachi Kumbla, Shantanu Pimpalkar, Aditya Sharma. "Snake Species Identification and Recognition", 2020 IEEE Bombay Section Signature Conference (IBSSC), 2020 Publication	<1 %
12	link.springer.com Internet Source	<1 %
13	www.scitepress.org Internet Source	<1 %
14	www.cghr.org Internet Source	<1 %
15	Submitted to PES University Student Paper	<1 %
16	www.slideshare.net Internet Source	<1 %
17	ncet.co.in Internet Source	<1 %
18	repository.usfca.edu Internet Source	<1 %

Fig 9.2: Plagiarism Report 2

19	"Next Generation Information Processing System", Springer Science and Business Media LLC, 2021 Publication	<1 %
20	dspace.lib.cranfield.ac.uk Internet Source	<1 %
21	ceur-ws.org Internet Source	<1 %
22	www.researchgate.net Internet Source	<1 %
23	Submitted to Southampton Solent University Student Paper	<1 %
24	aclanthology.org Internet Source	<1 %
25	fdocuments.in Internet Source	<1 %
26	issuu.com Internet Source	<1 %
27	"Developments in Information and Knowledge Management Systems for Business Applications", Springer Science and Business Media LLC, 2023 Publication	<1 %
28	D.D.K.R.W. Dandeniya, B.C.T. Wickramasinghe, C. Dasanayaka. "A Web-	<1 %

Fig 9.3: Plagiarism Report 3

based Application for Snake Species Identification using Vision Transformer and CNN-based Ensemble Meta Classifier", 2022 IEEE Pune Section International Conference (PuneCon), 2022
Publication

29	elifesciences.org Internet Source	<1 %
30	prezi.com Internet Source	<1 %
31	Submitted to Liverpool John Moores University Student Paper	<1 %
32	utpedia.utp.edu.my Internet Source	<1 %
33	www.diplomarbeiten24.de Internet Source	<1 %
34	"Artificial Intelligence Systems and the Internet of Things in the Digital Era", Springer Science and Business Media LLC, 2021 Publication	<1 %
35	"Proceedings of Second International Conference on Advances in Computer Engineering and Communication Systems", Springer Science and Business Media LLC, 2022 Publication	<1 %

Fig 9.4: Plagiarism Report 4

36 Ashis K. Mukherjee. "The 'Big Four' Snakes of India", Springer Science and Business Media LLC, 2021 <1%
Publication

37 Isabelle Bolon, Lukáš Picek, Andrew M. Durso, Gabriel Alcoba, François Chappuis, Rafael Ruiz de Castañeda. "An artificial intelligence model to identify snakes from across the world: Opportunities and challenges for global health and herpetology", PLOS Neglected Tropical Diseases, 2022 <1%
Publication

38 Md Alimul Haque, Shameemul Haque, Deepa Sonal, Kailash Kumar, Ejaz Shakeb. "Security Enhancement for IoT Enabled Agriculture", Materials Today: Proceedings, 2021 <1%
Publication

39 www.mdpi.com <1%
Internet Source

40 "Computer Vision and Image Processing", Springer Science and Business Media LLC, 2021 <1%
Publication

41 Singh, Aditya, Scott SK Lee, Marguerite Butler, and Victor Lubecke. "Activity monitoring and motion classification of the lizard Chamaeleo jacksonii using multiple Doppler radars", 2012 Annual International Conference of the IEEE <1%

Fig 9.5: Plagiarism Report 5

	Engineering in Medicine and Biology Society, 2012. Publication	
42	Y A Roopashree, M Bhoomika, R Priyanka, K Nisarga, Sagarika Behera. "Monitoring the Movements of Wild Animals and Alert System using Deep Learning Algorithm", 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2021 Publication	<1 %
43	bbrc.in Internet Source	<1 %
44	ebin.pub Internet Source	<1 %
45	papers.ssrn.com Internet Source	<1 %
46	pdfslide.net Internet Source	<1 %
47	www.ijartet.com Internet Source	<1 %
48	www.ijert.org Internet Source	<1 %
49	www.ijirset.com Internet Source	<1 %
	www.readkong.com	

Fig 9.6: Plagiarism Report 6

50	Internet Source	<1%
51	www.vnrvjiet.ac.in Internet Source	<1%
52	Ali Cheloee Cheloee Darabi, Shima Rastgordani, Mohammadreza Khoshbin, Vinzenz Guski, Siegfried Schmauder. "Hybrid Data-Driven Deep Learning Framework for Material Mechanical Properties Prediction with the Focus on Dual-Phase Steel Microstructures", Materials, 2023 Publication	<1%
53	K Prabu, P Sudhakar. "Design and Implementation of an Automated Control System for Anomaly Detection Using an Enhanced Intrusion Detection System", 2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2022 Publication	<1%
54	doi.org Internet Source	<1%
55	Prasanna Venkatesan Theerthagiri, Menakadevi Thangavelu. "Elephant Intrusion Warning System Using IoT and 6LoWPAN", International Journal of Sensors, Wireless Communications and Control, 2020 Publication	<1%

Fig 9.7: Plagiarism Report 7

BIBLIOGRAPHY

1) Detection and Prevention Mechanism of Snakes and Insects Biting from Farmers using IOT Monitoring System by J.Suganthi, Mrs.V.Suganthi, Mr.S.Giridharan.

2) Identification of Snake Species in Sri Lanka Using Convolutional Neural Networks by S. B Abayaratne, W. M. K. S. Ilmini and T. G. I. Fernando.

3) IoT Based Animal Harm Detection using Sensors by Creating an Alert Indushree S Nandini sridevi H S Navya V, Nikitha B M.

4) Snake Species Recognition using Tensor Flow Machine Learning Algorithm & Effective Convey System by Pranalini Joshi, Dhanesh Sarpale, Rohan Sapkal and Apeksha Rajput.

5) A Survey on Snake Species Identification using Image Processing Technique by Pranalini Joshi, Dhanesh Sarpale, Rohan Sapkal and Apeksha Rajput.

6) Prevention of Monkey Trespassing in Agricultural field Using Application Agricultural Specific Flooding Approach in Wireless Sensor Network by R.Radha, K.Kathiravan, V.Vineeth, J.Sanjay and S.Venkatesh

7) Embedded Home Surveillance System with Pyroelectric Infrared Sensor Using GSM by Rupali R. Ragade.

8) IOT based wild animal intrusion detection system by Prajna. P, Soujanya B.S and Mrs. Divya.

9) Solar Fencing Unit and Alarm for Animal Entry Prevention by Krishnamurthy B, Divya M, Abhishek S and Shashank H A.

10) Suraweera, W., Warrell, D. A., Whitaker, R., Menon, G. R., Rodrigues, R., Fu, S. H., Begum, R., Sati, P., Piyasena, K., Bhatia, M., Brown, P. O., & Jha, P. (2020, July 7). Trends in snakebite deaths in India from 2000 to 2019 in a nationally representative mortality study. ELife, 9. https://doi.org/10.7554/elife.54076

11) Ortho Snake B Gon Snake Repellent Granules, 2-Pound (Not Sold in AK) : Amazon.in: Garden & Outdoors, n.d.

12) O. (n.d.). EcoBugBye Natural Ultrasonic Pest Repeller Indoor ? 4 Pack Pest Control, Electronic Plug-in Repellent for Insects, Rodents, Mice, Rats, Roaches, Spiders, Flies, Ants, Eco-Friendly, Humans&Pets Safe [B071CJ5HH6] $6.60. EcoBugBye Natural

Ultrasonic Pest Repeller Indoor ? 4 Pack Pest Control, Electronic Plug-in Repellent for Insects, Rodents, Mice, Rats, Roaches, Spiders, Flies, Ants, Eco-Friendly, Humans&Pets Safe [B071CJ5HH6] $6.60. https://www.outdootp.shop/ecobugbye-natural-ultrasonic-pest-repeller-indoor-4-pack-pest-control-electronic-plugin-repellent-for-insects-rodents-mice-rats-roaches-spiders-flies-ants-ecofriendly-humanspets-safe-p-263358.html

13) Maxwell, C. (2022, March 4). Snake Repellent: How To Keep Snakes Away. AZ Animals. https://a-z-animals.com/blog/snake-repellent-how-to-keep-snakes-away/

14) S. A. Halim et al., "A development of snake bite identification system (N'viteR) using Neuro-GA," 2012 International Symposium on Information Technologies in Medicine and Education, 2012, pp. 490-494, doi: 10.1109/ITiME.2012.6291349.

15) Dr. Mahesh K Kaluti, Naveen kumar GS, Vinaya B, “IoT Based Wireless Sensor Network for Earlier Detection and Prevention of Wild Animals Attack on Forming Lands”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03 | Mar-2018.

16) Lekshmi Kalinathan, Prabavathy Balasundaram, Pradeep Ganesh, Sandeep Sekhar Bathala and Rahul Kumar Mukesh, “Automatic Snake Classification using Deep Learning Algorithm”, CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021.

17) A. Yousefi, A. A. Dibazar and T. W. Berger, "Intelligent fence intrusion detection system: detection of intentional fence breaching and recognition of fence climbing," 2008 IEEE Conference on Technologies for Homeland Security, 2008, pp. 620-625, doi: 10.1109/THS.2008.4635057.

18) J. de Vries, "A low-cost fence impact classification system with neural networks," 2004 IEEE Africon. 7th Africon Conference in Africa (IEEE Cat. No.04CH37590), 2004, pp. 131-136 Vol.1, doi: 10.1109/AFRICON.2004.1406647.

19) L. Wijesinghe, P. Siriwardena, S. Dahanayake, D. Kasthuriratne, R. Corea and D. Dias, "Electric Fence Intrusion Alert System (eleAlert)," 2011 IEEE Global Humanitarian Technology Conference, 2011, pp. 46-50, doi: 10.1109/GHTC.2011.16.

20) Singh A, Lee SS, Butler M, Lubecke V. “Activity monitoring and motion classification of the lizard Chamaeleo jacksonii using multiple Doppler radars”. Annu Int Conf IEEE Eng Med Biol Soc. 2012;2012:4525-8. doi: 10.1109/EMBC.2012.6346973. PMID: 23366934; PMCID: PMC4891065.

21) S. Giordano, I. Seitanidis, M. Ojo, D. Adami and F. Vignoli, "IoT solutions for crop protection against wild animal attacks," 2018 IEEE International Conference on Environmental Engineering (EE), 2018, pp. 1-5, doi: 10.1109/EE1.2018.8385275.

22) Cody H. Jensen,Nancy G. Caine, "Preferential snake detection in a simulated ecological experiment", Wiley Online Library, 08 January 2021,https://doi.org/10.1002/ajpa.24224

23) Olszewska, J. I. (2022). "Snakes in trees: an explainable artificial intelligence approach for automatic object detection and recognition". In Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 3: ICAART (pp. 996-1002). SciTePress. https://doi.org/10.5220/0010993000003116

24) T. Burghardt and J. C´ alic, "Analysing animal behaviour in wildlife videos using face detection and tracking", IEE Proc.-Vis. Image Signal Process., Vol. 153, No. 3, June 2006.

25) S. U. Sharma and D. J. Shah, "A Practical Animal Detection and Collision Avoidance System Using Computer Vision Technique," in IEEE Access, vol. 5, pp. 347-358, 2017, doi: 10.1109/ACCESS.2016.2642981.

26) Sheela.S, Shivaram. K. R, Chaitra. U, Kshama. P, Sneha. K.G, Supriya. K.S," Low-Cost Alert System for Monitoring the Wildlife from Entering the Human Populated Areas Using IOT Devices", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, Special Issue 10, May 2016.

27) Anika Patel, Lisa Cheung, Nandini Khatod, Irina Matijosaitiene, Alejandro Arteaga and Joseph W. Gilkey Jr. "Revealing the Unknown: Real-Time Recognition of Galápagos Snake Species Using Deep Learning", Animals 2020, 10, 806; doi:10.3390/ani10050806

28) James Sabatier, Thyagaraju Damarla and Asif Mehmood, "Detection of people and animals using non-imaging sensors", 14th International Conference on Information Fusion, Chicago, Illinois, USA, July 5-8, 2011

29) James A. 2017. "Snake classification from images". PeerJ Preprints 5: e2867v1 https://doi.org/10.7287/peerj.preprints.2867v1

30) Kunal Jain, NishantKabra, Shriya Khatri, Surekha Dholay, "Automated Snake Bite Prevention System", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075 (Online), Volume-10 Issue-3, January 2021.

31) Mr.Srinivasa Reddy Gudibandi, Mr. M. Amarnath, "Design of Smart Surveillance

System using PIR and Ultrasonic Sensor", International Journal & Magazine of Engineering, Technology, Management and Research, Volume No: 2 (2015), Issue No: 11 (November).

32) "Study estimates more than one million Indians died from snakebite envenoming over past two decades" Who, 10 Jul. 2020, https://www.who.int/news/item/10-07-2020-study-estimates-more-than-one-million-indians-died-from-snakebite-envenoming-over-past-two-decades.