VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Jnana Sangama, Belagavi – 590018

Project Work Report

on

"WILD ANIMAL DETECTION IN FARMLAND"

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

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

by

DHANUSH 4MT20IS007

Under the Guidance of
Prof. Manjunath H
Head of the Dept
Dept. of Information Science & Engineering

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
(*Accredited by NBA*)

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)
Autonomous Institute Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Accredited by NAAC with A+ Grade & ISO 9001: 2015 Certified Institution
Badaga Mijar, Moodabidri-574225, Karnataka
2023-24

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)
Autonomous Institute Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Accredited by NAAC with A+ Grade & ISO 9001: 2015 Certified Institution
Badaga Mijar, Moodabidri-574225, Karnataka

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

(*Accredited By NBA*)

CERTIFICATE

Certified that the Project Work entitled **"WILD ANIMAL DETECTION IN FARMLAND"** carried out by **DHANUSH (4MT20IS007)** in partial fulfillment for the award of degree of **Bachelor of Engineering in Information Science & Engineering** of the **Visvesvaraya Technological University, Belagavi during the year 2023-24**. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental Library.

The Project has been approved as it satisfies the academic requirements in respect of Project work prescribed for the above said degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Prof. Manjunath H	**Prof. Manjunath H**	**Dr. Prashanth C M**

External Viva

Name of the Examiner **Signature With Date**

1.

2.

MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)

Autonomous Institute Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Accredited by NAAC with A+ Grade & ISO 9001: 2015 Certified Institution

Badaga Mijar, Moodabidri-574225, Karnataka

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

(*Accredited By NBA*)

DECLARATION

I, **DHANUSH (4MT20IS007)** student of 8th semester BE in Information Science & Engineering, **Mangalore Institute of Technology & Engineering, Moodabidri,** hereby declare that the project work entitled **"WILD ANIMAL DETECTION IN FARMLAND",** submitted to the **Visvesvaraya Technological University, Belagavi** during the academic year **2023-24**, is a record of an original work done by us under the guidance of **Prof. Manjunath H,** Head of the Department, Department of Information Science & Engineering, Mangalore Institute of Technology & Engineering Moodabidri. This project work is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Information Science & Engineering. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree.

Date: **DHANUSH**

Place: MOODABIDRI

ABSTRACT

Crop damage caused by animal attacks is one of the major threats in reducing the crop yield. Due to the expansion of cultivated land into previous wildlife habitat, crop raiding is becoming one of the most antagonizing human-wildlife conflicts. Farmers in India face serious threats from pests, natural calamities & damage by animals resulting in lower yields. Traditional methods followed by farmers are not that effective and it is not feasible to hire guards to keep an eye on crops and prevent wild animals. Since safety of both human and animal is equally vital, it is important to protect the crops from damage caused by animal as well as divert the animal without any harm. Crop damage brought about by animal assaults is one of the significant dangers in lessening the harvest yield. Because of the extension of developed land into past wildlife territory, crop striking is becoming one of the most alienating human-wildlife clashes. Effective and solid checking of the wild animals' right in their natural habitat is fundamental. Thus, in order to overcome above problems and to reach our aim, we use deep learning to detect animals, entering into our farm by using deep neural network concept, a division in computer vision. In this proposed system, system will monitor the entire farm at regular intervals through a camera which will be recording the surrounding throughout the day. With the help of a deep learning model, system detect the entry of animals and system play appropriate sounds to drive the animal away.

ACKNOWLEDGEMENT

The satisfaction and the successful completion of this project would be incomplete without the mention of the people who made it possible, whose constant guidance encouragement crowned our efforts with success.

This project is made under the guidance of **Prof. Manjunath H,** Head of the Department, Department of Information Science & Engineering. We would like to express my sincere gratitude to our guide for all the helping hand and guidance in this project.

We would like to thank our project coordinator **Prof. Narendra U P**, Associate Professor in the Department of Information Science & Engineering for his cordial support, valuable information and guidance, which helped us in completing this project through the various stages.

We would like to thank our Principal **Dr. Prashanth C M**, for encouraging us and giving us an opportunity to accomplish the project.

We also thank our management who helped us directly or indirectly in the completion of this project.

Our special thanks to faculty members and others for their constant help and support.

Above all, we extend our sincere gratitude to our parents and friends for their constant encouragement with moral support.

Dhanush (4MT20IS007)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Agriculture meets food demands of the population and also provides various raw materials for industries. Interference of animals in agricultural lands causes a huge loss of crops. Crop damage due to raiding wild animals has become a major issue of concern these days. Animals like wild buffalo, cow and goat are extremely destructive and have also caused human casualties sometimes. The total losses in crop yield are high for potato and wheat in villages. Small farmers lose up to 40 to 50 percent of their crop to wild animals and they cannot take any harsh measures due to the strict wildlife laws. Human-animal conflict is rising intensely as elephants are a highly conflict prone wildlife species, especially in India. Thus, there is need for a system which can help the farmers to drive away these animals as soon as they learn about their intrusion. Overpopulation causes deforestation, which reduces the availability of food, water, and shelter in forested regions. Animals are therefore increasingly interfering in residential settings According to the real fact in the daily basis every living being needs food so that with no food no living beings cannot live, while the crops are damaged by the animals the demand will be increased and the economy of the food products is increased so that thepoor people are not able to buy the food products.

1.2 SCOPE OF THE PROJECT

Animal attack in the farm area is considered as the major threat, which will reduce the amount of crop. The main reason for this is the expansion of cultivated land. Human-wildlife conflicts occur through crop raiding which is common in these days. The farmers in India face huge loss through natural calamities, animal attacks etc. The age old methods practiced by the farmers are not efficient. It is practically impossible to appoint guards to monitor the farm area. The main aim of the project is to help the farmer to save the crops without harming the animals. The steps performed here is to protect the crops from animal attack by taking appropriate measure to keep the animal away by producing appropriate sound without killing animal.

1.3 OBJECTIVES

The main objectives of proposed system are to develop a deep learning model that process animal images and classifies them. To design a system that identifies the presence different wild animals from a farm camera and to notify the farm owner. To apply the concept of deep learning in determining the wild animal. To enable the farmer to save their crop from being vanished by the wild animals.

1.4 ORGANIZATION OF THE REPORT

- **Chapter-1: Introduction**

  This chapter describes the organization of the report, objectives of the proposed model and brief introduction of our proposed model.

- **Chapter-2: Literature Survey**

  This chapter mainly deals with all the observation, which is conducted as initial study before the actual development of the project. It also describes the details regarding existing system and its disadvantages.

- **Chapter-3:System Requirement Specification**

  This chapter speaks about the product perspective, user characteristics, its assumption and dependencies, specific requirements, functionality along with resource requirements.

- **Chapter-4: System Design**

  This chapter deals with the advance software engineering where the entire flow of the project is represented by data flow diagram and the architecture of the project.

- **Chapter-5: Gantt Chart**

  This chapter deals with the Gantt chart. It is a type of bar chart, developed by Henry Gantt that illustrates project schedule.

- **Chapter-6: Implementation**

  This chapter deals with the steps involved in the creation of the project work. It is defined with the help of code snippets for the ease of reader.

- **Chapter-7: System Testing**

  This chapter deals with the system testing that evaluate the performance, accuracy, and reliability under various conditions.

- **Chapter-8: Result and Discussion**

  This chapter deals with the results obtained from the project, Snapshots or screenshots of the implemented system.

- **Chapter-9: Scope for Future Works**

  These sections also suggest some of the enhancement idea which couldn't be covered up due to constraint of time and resources.

- **Chapter-10: Conclusion**

  These sections are mainly the summary of the entire project development.

CHAPTER 2

LITERATURE SURVEY

LITERATURE REVIEW

A literature survey shows the various analysis and research made in the field of interest and results already published, taking into account the various parameters of the project and the extent of the project. It includes researches made by various analysts- their methodology and the conclusion they have arrived at. It is the most important part of the report as it gives the direction in the area of research.

2.1 BACKGROUND RESEARCH

Wild animal detection in farmland background is a critical area of research aimed at mitigating human-wildlife conflicts and minimizing agricultural losses. With increasing encroachment of human settlements into natural habitats, conflicts between farmers and wild animals have become more prevalent. In response, researchers are exploring innovative methods to detect the presence of wild animals in farmland backgrounds accurately. This involves leveraging advancements in technology such as remote sensing, machine learning, and sensor networks. Remote sensing techniques, including satellite imagery and drones, offer the ability to monitor vast agricultural landscapes efficiently. Machine learning algorithms can then be trained on these data to recognize patterns associated with the presence of wild animals, enabling automated detection and alert systems.

[1] Animals Detection System in The Farm Area Using Iot by Santhiya S, Dhamodharan Y, Kavi Priya NE, Santhosh CS and Surekha M. 'A Smart Farmland Using Raspberry Pi Crop Prevention and Animal Intrusion Detection System '. International Research Journal of Engineering and Technology (IRJET), 2018; 05(03).

The primary objective is to protect the crops from being ravaged many times by wild animals like elephants, wild pigs, monkeys. There will be a large amount of loss to farmers. To avoid the financial losses for the farmer, so that it is very much important to protect agricultural fields or farms from the animals. Farmers cannot block entire fields or remain on the field all day to secure it. to safe-guard the field and intimate the farmer via call alert when any animal is detected. This project has Arduino Uno used to control the Ultrasonic sensor while both are connected to each signal transmitted to the GSM module to transmit a call notification and alarm sound.

[2] Wild Animal Detection in Agriculture Farms Using Deep Convolutional Neural Network by Duhart C, Dublon G, Mayton B, and Paradiso J. 'Deep Learning Locally Trained Wildlife Sensing in Real Acoustic Wetland Environment'. In Thampi SM, Marques O, Krishnan S, Ciuonzo D and Kolekar MH (eds.), Advances in Signal Processing and Intelligent Recognition Systems, 2019: 3–14'

This fosters an algorithm to identify the animals that intrudes into the agriculture land. Since there are enormous number of various animals physically distinguishing them can be a troublesome undertaking. This calculation arranges animals in view of their pictures so we can screen them all the more proficiently. This can be accomplished by applying yolo v3 algorithms which is a powerful real-time object detection algorithms. YOLOv3 detect an object with the help of the features of deep convolutional neural network.

The primary goal of the project is to safe watchman the farming field from wild animals and furthermore to safeguard them by pushing them away as opposed to killing. The project additionally plans to safeguard human lives from creature assaults. We are involving an integrative methodology in the field of Deep Learning to give a checking and repulsing framework for crop insurance against creature assaults.

[3] Animal Detection in Farm Area by Govind S.K., Jayson E.A (2018) Crop Damage by Wild Animals in Thrissur District, Kerala, India. In: Sivaperuman C., Venkataraman K. (eds) Indian Hotspots. Springer, Singapore.

The main aim of the project is to help the farmer to save the crops without harming the animals. The steps performed here is to protect the crops from animal attack by taking appropriate measure to keep the animal away by producing appropriate sound without killing or harming the animals. Thus, to reach our goal and solve the problem, we make use of machine learning technique to detect the animal entering into the farm area using convolutional neural network. Here in this project, the entire farm area is monitored at regular interval of time through the camera, which helps to record the entire surrounding of the farm. Machine learning model is designed to detect the animal entering the farm and plays the appropriate sound to shoo an animal away from the farm such that the crops are prevented from damage. Different types of packages and concepts of the convolutional neural network is used to design the model to achieve the desired aim in the project.

[4] WildDect -YOLO by Arunabha M. Roy, Jayabrata Bhaduri, Teerath Kumar, Kislay Raj, An efficient and robust computer vision -base d accurate object localization model for automated endangered wildlife detection.

With climate instability, diverse ecological disturbances, and human actions affecting the survival of many endangered animal species, an up-to-date, accurate, and thorough detection procedure is critical for biodiversity protection, conservation, and ecosystem management. Current cutting-edge wildlife identification models, however, frequently lack better feature extraction capabilities in complicated situations, restricting the creation of accurate and detection models. To solve this issue, they introduced WilDetect-YOLO, a deep learning (DL)-based automated high-performance detection model for real-time endangered wildlife identification.

[5] A Case Study by Sujatha Kamepalli, Venkata Krishna Kishore Kolli, Srinivasa Rao Bandaru, Animal Breed Classification and Prediction Using Convolutional Neural Network Primates.

Primates have crucial roles in human evolution, civilizations, and religions. It is our obligation to protect primate breeds by reintroducing them into their native habitat. This article uses a deep Convolutional Neural Network to categorise and forecast primate races from test images. We utilised a dataset of 10 monkey species from Kaggle data science community. This dataset contains ten different monkey breeds labelled n0–n9. The model trained with many epochs achieved an accuracy of 0.8050 on the training set and 0.7353 on the validation set at epoch 20. The trained model correctly predicted the primate breeds. Predictions can help discover and protect primate breeds from extinction. Future study might automate primate breed identification with IoT integration.

2.2 EXISTING SYSTEM

The current systems essentially give the observation usefulness. Additionally, these systems don't give security from wild animals, particularly in such an application region. They additionally need to make moves in light of the on the kind of animal that attempts to enter the region, as various techniques are taken on to keep various animals from entering such confined regions. Likewise the ranchers resort to different techniques by raising human mankind and likenesses in their homesteads, which is ineffectual in warding off the wild animals, however is valuable somewhat to avert birds. The other usually involved strategies by the ranchers to forestall the harvest vandalization by animals incorporate structure actual obstructions, utilization of electric wall and manual reconnaissance and different such thorough and risky techniques.

2.3 PROPOSED SYSTEM

System contains Four main modules called Pre-processing, Model creation, System Training and Classification. Pre-processing module is used to pre- process the dataset images by resizing them to required dimension. Model Creation module is used to construct a machine language model using desired number of layers. In the Classification module, once the input test image is received, it undergoes a series of processing steps within the trained machine learning model. These steps involve feature extraction and pattern recognition, where the model analyzes the unique characteristics of the image to make an accurate classification decision. System training phase is used to train the system with dataset images and store the model weight.

Classification module is used to classify the input test image to determine the animal. If itis captured as wild animal system sends a notification mail to farm owner and as well as the nearest forest officers.

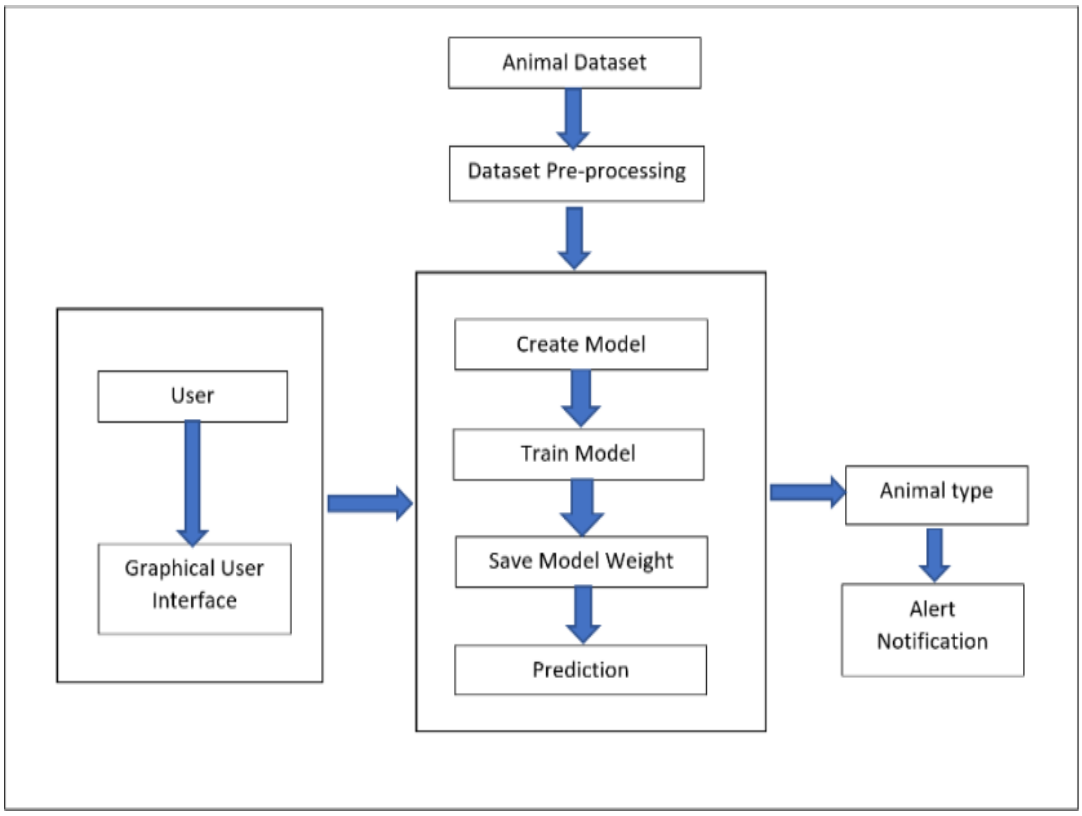

Figure 2.3 Proposed System

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

The basic goal of the requirements phase is to produce the software requirements specification, which describes the complete external behavior of the proposed software. The complete description of the behavior of a system developed is explained below. It includes functional requirements of the system. In addition to functional requirements, the non-functional requirements and user-interface requirements are also explained. Requirement analysis is done in order to understand the problem the software system is to solve. The problem could be automating an existing the manual process, developing a new automated system, or a combination of two.

3.1 FUNCTIONAL REQUIREMENTS

System has following functional modules:

- Registration Module: This module is used to do the registration. User need to provide name, mobile no, email Id, address etc during registration.

- Login Module: this module is used by the user to login into the system. They need to provide username and password to login.

- Upload Module: This module is used to upload animal image for classification.

- Train Model: This model is used to build a machine learning model that predict the food quality.

- Pre-process Module: This module is used to preprocess the dataset images using image augmentation and resizing.

- Notification Module: This module allows to send notification to owner whenever wild animal is detected.

- Classification Module: This module is used to classify the test image either into normal animal and wild animal.

3.2 NON-FUNCTIONAL REQUIREMENTS

Quality attribute of a system that are used to judge the operation of a system and measure the performance of the system.

- Availability – The defined system should be available all the time to monitor the animals.

- Robustness-The system should be able to provide result accurately to the end user.

- Reliable-The system designed should be a reliable one providing safety to the end user.

- Security-The designed system should be able to provide a secure line for data communication without any interrogation from the third party

3.3 SOFTWARE REQUIREMENTS

Following are the software requirements for development of the project.

Operating System : Windows 10

Front End : React native

Back End : .NET

IDE : Visual studio

Database : MySQL

3.4 HARDWARE REQUIREMENTS

Following shows the Hardware requirements for development of the project.

Processor	: Intel(R) Core™ i3-6006UCPU@2.00GHz
Hard Disk	: 500MB
Memory	: 8GB
Display	: 2.8 inches or larger

3.5 SUMMARY OF THE CHAPTER

In this chapter, we have discussed the Functional requirements to identify the use cases involved and sketch UML diagrams for the use cases identified which includes Activity diagrams and Sequence diagrams. Simultaneously, we discuss about the Non-Functional requirement which provides quality attributes of a system used to judge operation of a system and also ensure the performance of the system. Finally, we conclude the chapter with hardware and software requirements which lists out the software used to build the application and the tools involved to incorporate the software which are used for implementing.

CHAPTER 4

SYSTEM DESIGN

The software application design is a difficult entity. The improvement of the application or a system follows the Software Development Life Cycle (SDLC). The next step after analysis of the entire requirement is design. The main purpose of system design is to attain the entire design of the system, also to catch the idea about the different modules present in the system, relation between the different modules, to know about the purpose of each module, and how all the modules merge together to form a complete system. The design gives the overview of the system flow, the architecture and to moreover provides the common solution for the application.

4.1 ARCHITECTURAL DESIGN

An architectural explanation is a formal description and illustration of a system, organized in a manner that supports reason in relation to the structure ofthe system which comprises system components, the externally detectable properties of individual components, the interaction among them, and provides a plan from which products can be procured, and systems developed, that will work mutually to implement the on the whole as a system.

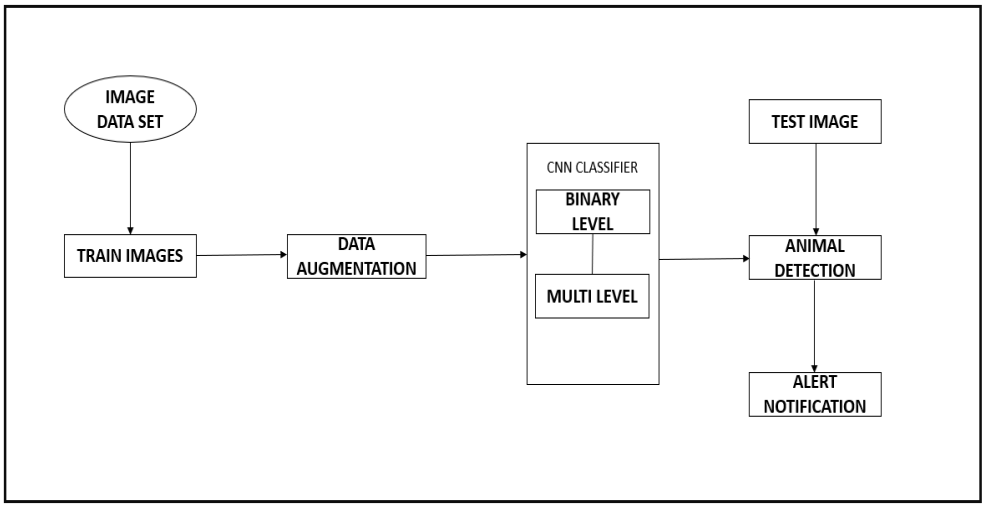

Figure 4.1 Architectural Design

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic- related data. This could be MySQL, MySQL, or PostgreSQL, etc.

View

The View component is used for all the UI logic of the application. It can be written inHTML, CSS, JS. It also supports react, angular etc.

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Modelcomponent and interact with the Views to render the final output. The controller can be written in Java, Python, C#, C++ etc.

4.2 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical illustration of the flow of data throughout an information system, modelling its procedural aspects. A DFD is often used as a beginning step to create an outline of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing. DFDs help in understanding how data is processed, stored, and transformed within a system, aiding in system analysis, design, and communication of system functionalities to stakeholders in a clear and concise manner.

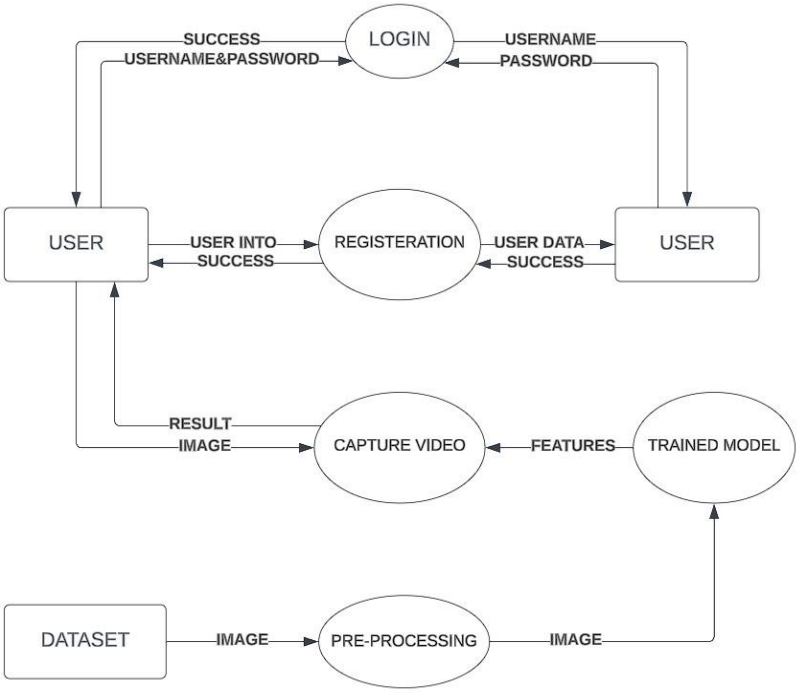

Figure 4.2 Data Flow Diagram

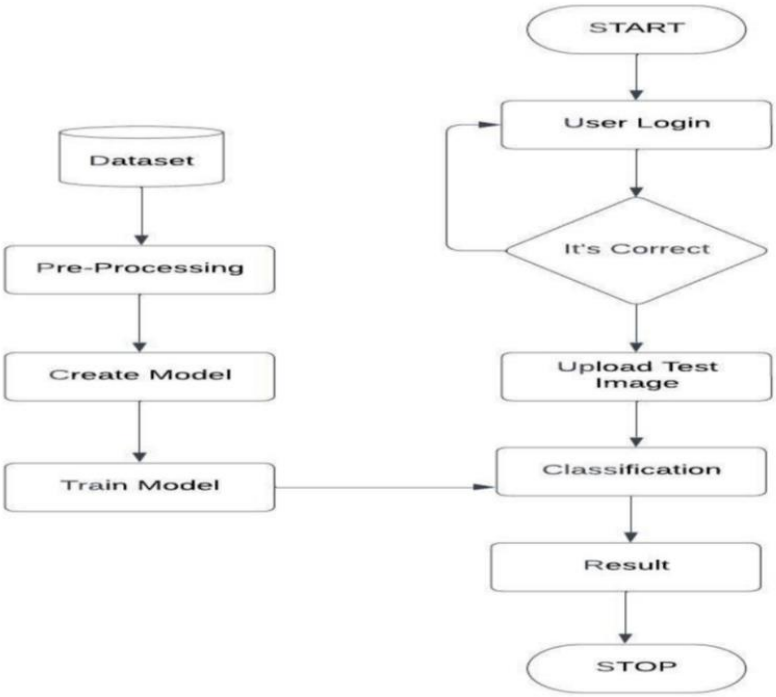

Figure 4.3 Work Flow Diagram

4.4 USE CASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a source and a destination. The two main components of a use case diagram are use cases and actors. It displays the relationship among them. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Use case diagram of this system is as shown below.

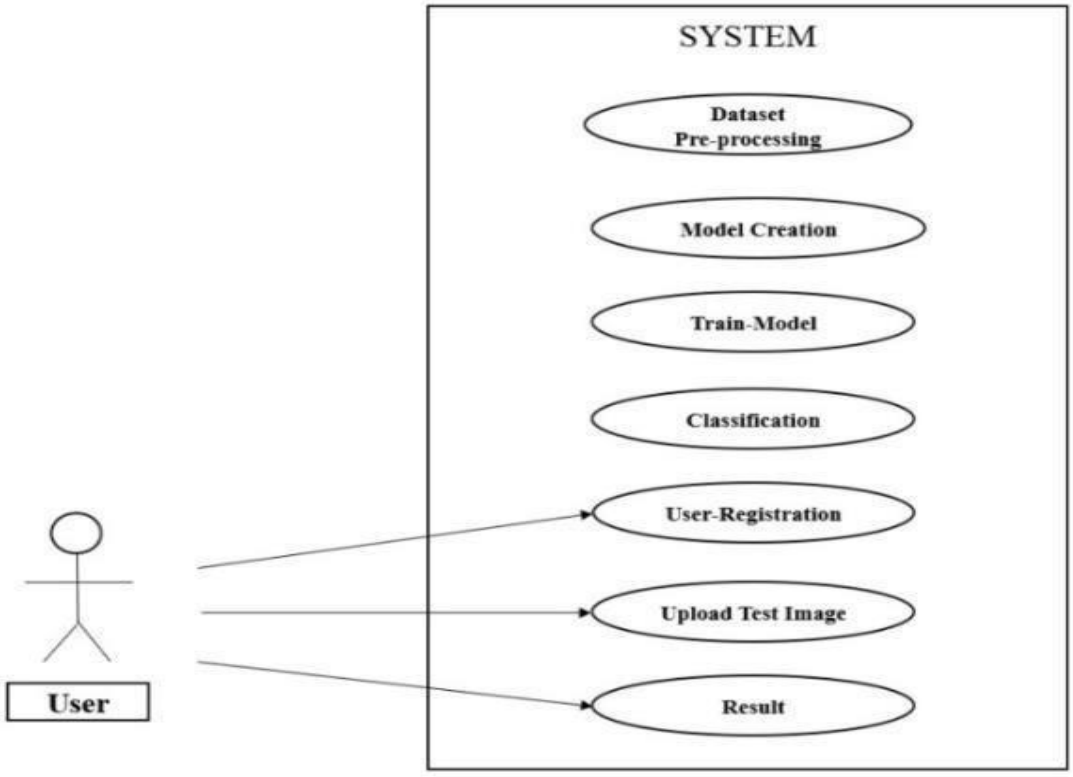

Figure 4.4 Use Case Diagram

4.5 SEQUENCE DIAGRAM

A sequence diagram for system User shows object interactions arranged in time sequence. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called as Event Diagrams or Event Scenarios.

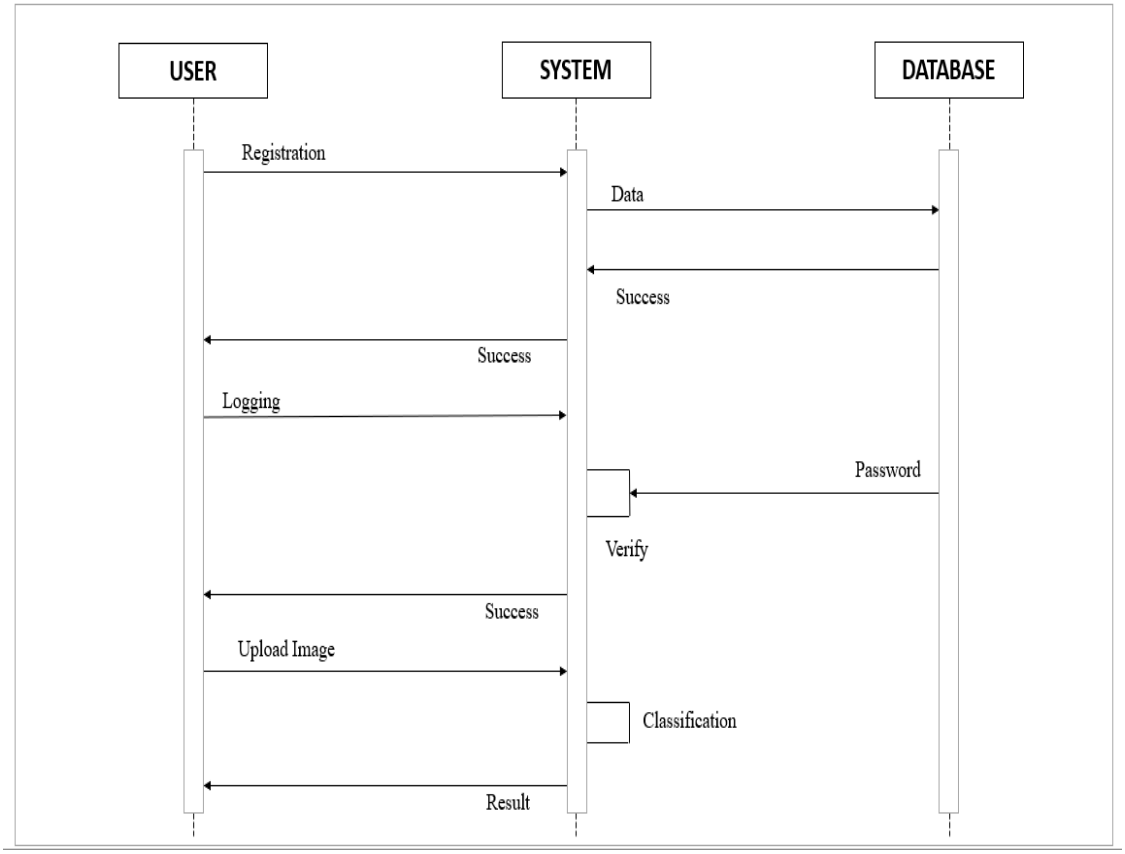

Figure 4.5 Sequence Diagram

4.6 CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network is a representative algorithm in deep learning. It is essentially a multi-layer perceptron that simulates local perception to achieve an input-to-output mapping. It extracts the characteristics of the data at different scales through multiple convolutions and pooling. What is unique in the CNN network is the way used in local connections and shared weights. On the one hand, it reduces the number of weights which makes the network easy to optimize, and on the other hand, it reduces the risk of overfitting. CNNs are generally composed of three mutually supported levels, namely convolutional layer, pooling layer, fully connected and Softmax layer. In the convolution process, we get local features. Since one of the convolution layers is composed of multiple convolution units, in the calculation process, in order to extract more features about the input parameters, it is necessary to obtain more complex feature correlation values from low-level convolutional layers through multi-level cascading.

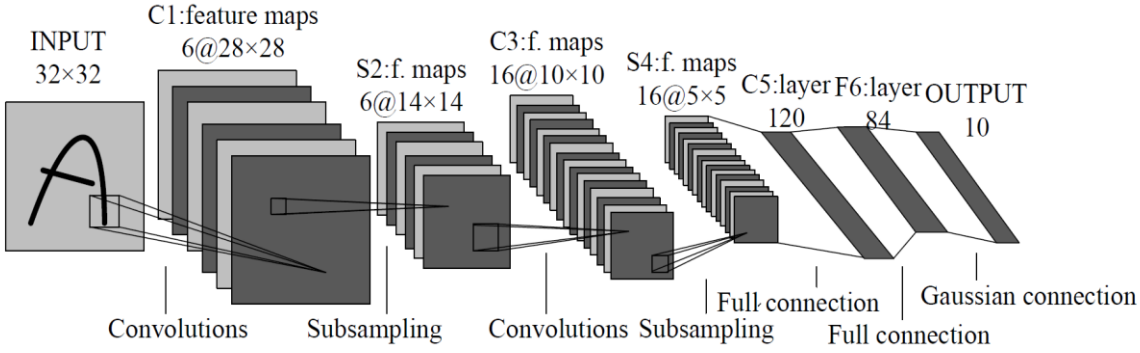

Figure 4.6 CNN Model

Workflow

The following steps are carried out to implement the proposed system.

Pre-processing

Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Dataset is pre-processed to resize the dataset images.

Train Test Split

The process of organizing data into groups and classes on the basis of certain characteristics is known as the classification of data. Classification helps in making comparisons among the categories of observations. It can be either according to numerical characteristics or according to attributes. So here we need to visualize the prepared data to find whether the training data contains the correct label, which is known as a target or target attribute. In this project the dataset is split into 90 samples for training and 10 for testing.

Training

The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The term ML model refers to the model artifact that is created by the training process. The training data must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

Picking the Model

Pickling is a process in which model is stored in a file for future use. Before pickling model must be trained well and optimized to maximum extent possible because after picking model, data will not be trained. By pickling there is no need of training the model each time the user makes a classification.

CHAPTER 5

GANTT CHART

A Gantt chart is a type of bar chart, developed by Henry Gantt that illustrates a project schedule. Gantt charts illustrate the start and finish of the terminal elements and summary elements of the project. Terminal elements and summary elements comprise the work breakdown structure of the project.

The following is the Gantt chart of the project "Wild Animal Detection In Farmlands"

Tasks: (Months)	2023				2024			
	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Selection of Topic								
Literature Review								
Synopsis Report and PPT Preparation								
Software Requirements Specification								
Dataset Collection								
Project Designing								
Project Phase-1 Presentation								
Implementation								
Testing								
Documentation								
Project Phase-2 Presentation								

Figure 5.1 Gantt Chart

CHAPTER 6

SYSTEM IMPLEMENTATION

A crucial phase in the system development life cycle is successful implementation of new system design. Implementations simply mean converting new system design into operation. The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of computer system. Implementation is used here to mean the process of converting a new or revised system design into an operational one.

6.1 Technologies Used

OpenCV: OpenCV is a Python library that allows you to perform image processing and computer vision tasks. It provides a wide range of features, including identifying and recognizing faces, identifying objects, classifying human actions in videos, tracking camera movements, tracking moving objects, extracting 3D object models, generating 3D point clouds from stereo cameras, stitch images together to generate an entire scene with a high resolution image and many more. In this OpenCV Tutorial in Python, we'll be learning more about the library. OpenCV is an open-source software library for computer vision and machine learning. The OpenCV full form is Open Source Computer Vision Library. It was created to provide a shared infrastructure for applications for computer vision and to speed up the use of machine perception in consumer products. OpenCV, as a BSD-licensed software, makes it simple for companies to use and change the code. There are some predefined packages and libraries that make our life simple and OpenCV is one of them.

Matplotlib: Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of

the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information.

Django – Python Web Framework: Django is a Python-based free and open-source web framework that follows the model–template–views (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established in the US as a non-profit. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Keras: Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet is also the author of the XCeption deep neural network model. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page.

Numpy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures. NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ND array, it provides a lot of supporting functions that make working with ND array very easy. Arrays are very frequently used in data science, where speed and resources are very important.

TensorFlow: is an open-source machine learning framework developed by Google Brain, designed to provide a flexible platform for building and deploying machine learning models. With its widespread adoption across various industries and applications, TensorFlow has become one of the most popular and powerful tools in the field of artificial intelligence and data science. At its core, TensorFlow enables developers to build and train machine learning models efficiently by providing a comprehensive set of libraries and tools for numerical computation, optimization, and distributed computing. One of its key features is its flexibility, allowing users to create models ranging from simple linear regressions to complex deep neural networks. TensorFlow operates on the concept of computational graphs, where mathematical operations are represented as nodes, and the data flowing between these operations are represented as edges. This graph-based approach allows for efficient execution of operations and enables TensorFlow to leverage hardware acceleration techniques such as GPU and TPU to speed up computation.

6.2 Pseudo Code Of System :

Step 1: Collect dataset

Step 2: Pre-process dataset

Import libraries/modules

For every image in dataset:

Read a image from dataset using imread()

Resize the image using resize()

Perform Image augmentation using ImageDataGenerator()

Append images into images[]

Append labels into labels[]

End for

Step 3: Build Model

Create a CNN Model

Add an input layer of size (150,150)

Add convolution, pooling, dense layer

Step 4: Train Model

Split the dataset into training set and test set in 90:10 proportion

Configure training parameters (batch size, number of epoch)

Step 5: For I in range(epoch)

For image in training set(X_test):

Extract feature

Add corresponding labels (Y_test)

Evaluate the training accuracy and validation accuracy

If model accuracy < 90

Adjust model parameter

Repeat Step 5

Else:

Break;

Step 6: Save model weight in .h5 file

6.3 Code Snippets:

6.3.1 predict.py:

```
import numpy as np
from keras.preprocessing import image
from tensorflow.keras.models import load_model
from os import getcwd
import cv2 as cv
import imutils


  animal = ['Bear', 'Chinkara', 'Elephant', 'Lion', 'Tiger']
  def process():
  imagetest = cv.imread(getcwd() + '/media/test.png')
  # test_image = image.img_to_array(test_image)
  # test_image = np.expand_dims(test_image, axis=0)
  classifier = load_model(getcwd() + '\\trained_model_wild.h5')
  gray = cv.cvtColor(imagetest, cv.COLOR_BGR2GRAY)
  gray = cv.GaussianBlur(gray, (5, 5), 0)
  thresh = cv.threshold(gray, 45, 255, cv.THRESH_BINARY)[1]
  thresh = cv.erode(thresh, None, iterations=2)
  thresh = cv.dilate(thresh, None, iterations=2)
  cnts = cv.findContours(thresh.copy(), cv.RETR_EXTERNAL,
cv.CHAIN_APPROX_SIMPLE)
            cnts = imutils.grab_contours(cnts)
            c = max(cnts, key=cv.contourArea)
            extLeft = tuple(c[c[:, :, 0].argmin()][0])
            extRight = tuple(c[c[:, :, 0].argmax()][0])
            extTop = tuple(c[c[:, :, 1].argmin()][0])
            extBot = tuple(c[c[:, :, 1].argmax()][0])
            new_image = imagetest[extTop[1]:extBot[1], extLeft[0]:extRight[0]]
            image = cv.resize(new_image, dsize=(150,150),
            interpolation=cv.INTER_CUBIC)
            image = image / 255.
            image = image.reshape((1, 150, 150, 3))
```

```
    result = classifier.predict(image)
    print(result[0][0])
    print(result[0][1])
    print(result[0][2])
    index=result[0].tolist().index(max(result[0]))
    print(index)
    print(animal index)
    print(animal[index])
    return animal[index]
```

6.3.2 wildtrain.py:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import cv2
from tqdm import tqdm
import random as random
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from keras.utils import to_categorical
from keras.layers import
Dense,Dropout,Flatten,Conv2D,MaxPool2D,BatchNormalization
from keras.models import Sequential
from keras.losses import categorical_crossentropy
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
import warnings
warnings.filterwarnings('always')
warnings.filterwarnings('ignore')
bear_png='DS/bear_png'
chinkara='DS/chinkara'
elephant='DS/elephant'
```

```
lion='DS/lion'
tiger='DS/tiger'
IMG_SIZE= 150
predicters = []
target = []
def read_image(label,DIR):
    for img in tqdm(os.listdir(DIR)):
    #for img in DIR:


        path = os.path.join(DIR,img)
        # _, ftype = os.path.splitext(path)
        # if ftype == ".jpg":
        image = cv2.imread(path,cv2.IMREAD_COLOR)


        image = cv2.resize(image, (IMG_SIZE, IMG_SIZE),interpolation =
cv2.INTER_AREA)
        predicters.append(np.array(image))
        target.append(str(label))
         read_image("bear_png",bear_png)
         read_image("chinkara",chinkara)
         read_image("elephant",elephant)
         read_image("lion",lion)
         read_image("tiger",tiger)
         encoder = LabelEncoder()
         X = np.array(predicters)
         X = X/255
         y = encoder.fit_transform(target)
         y = to_categorical(y,8)
         print(X.shape)
         print(y)
         X_train,X_valid,y_train,y_valid =
         train_test_split(X,y,test_size=0.1,random_state=42)
         batch_size = 64
```

```
epochs = 5

num_classes = y.shape[1]

model = Sequential()

model.add(Conv2D(64, kernel_size=(3,
3),activation='relu',padding='same',input_shape=(IMG_SIZE,IMG_SIZ
E,3)))

model.add(MaxPool2D((2, 2),padding='same'))

model.add(BatchNormalization())

model.add(Dropout(0.2))

model.add(Conv2D(128,kernel_size= (3, 3),
activation='relu',padding='same'))

model.add(MaxPool2D(pool_size=(2, 2),padding='same'))

model.add(BatchNormalization())

model.add(Dropout(0.2))

model.add(Conv2D(128, kernel_size=(3, 3),
activation='relu',padding='same'))

model.add(MaxPool2D(pool_size=(2, 2),padding='same'))

model.add(BatchNormalization())

model.add(Dropout(0.3))

model.add(Conv2D(256, kernel_size=(3, 3),
activation='relu',padding='same'))

model.add(MaxPool2D(pool_size=(2, 2),padding='same'))

model.add(BatchNormalization())

model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3, 3),
activation='relu',padding='same'))

model.add(MaxPool2D(pool_size=(2, 2),padding='same'))

model.add(BatchNormalization())

model.add(Dropout(0.4))

model.add(Flatten())

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.3))

model.add(BatchNormalization())

model.add(Dense(num_classes, activation='softmax'))
```

```
model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0
01),metrics=['accuracy'])
imagegen = ImageDataGenerator(featurewise_center=False,
                              samplewise_center=False,
                              featurewise_std_normalization=False,
                              samplewise_std_normalization=False,
                             rotation_range=60,

                              zoom_range=0.1,
                              width_shift_range=0.1,
                              height_shift_range=0.1,
                              shear_range=0.1,
                              fill_mode='reflect')
imagegen.fit(X_train)
     model_dropout = model.fit_generator(imagegen.flow(X_train,y_train,
batch_size=batch_size),  epochs=epochs,verbose=1,
    validation_data=(X_valid,
y_valid),steps_per_epoch=X_train.shape[0] // batch_size
                         )
model.save('trained_model_wild.h5')
```

6.3.3 view.py:

```
from django.shortcuts import render
from WildAnimalDetectionApp.models import User
from WildAnimalDetectionApp import predict
from WildAnimalDetectionApp import helpers
from django.db.models import Q, Sum
from django.core.files.storage import default_storage
from django.core.files.base import ContentFile
from django.conf import settings


import shutil
import os
import cv2
import time
def index(request):
```

```
 return render(request,'index.html')

def user(request):
    return render(request,'user/index.html')

def registration(request):
    return render(request,'user/registration.html')

def saveUser(request):
    if request.method == 'POST':
        farmername = request.POST['uname']
        contactNo = request.POST['contactNo']

        emailId = request.POST['emailId']
        address = request.POST['address']
        username = request.POST['username']
        password = request.POST['password']

        user = User.objects.filter(
        Q(email=emailId) | Q(contact=contactNo) | Q(user_name=username)
        ).first()

        has_error = False
        error = ''

        if user != None and user.user_name == username:
            has_error = True
            error = 'Duplicate user name'
        if user != None and user.email == emailId:
            has_error = True
            error = 'Duplicate email'
        if user != None and user.contact == contactNo:


            has_error = True
            error = 'Duplicate contact number'

        if has_error:
            return render(request, "user/registration.html", {'error': error})



 user = User(name=farmername, contact=contactNo, email=emailId,
                    address=address, user_name=username, password=password)
        user.save()
```

```
      return render(request, "user/registration.html", {'success': 'User Added
Successfully'})
    else:
        return render(request, 'user/registration.html')


def userlogin(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = User.objects.values_list('password', 'id', 'name').\
            filter(user_name=request.POST['username'])
        user = User.objects.filter(
            user_name=username, password=password).first()

        if user == None:


     return render(request, 'user/index.html', {'error': 'Invalid login
credentials'})
        request.session['userid'] = user.id
        request.session['userName'] = user.name
        return render(request, 'user/userHome.html')

    else:
        return render(request, 'user/index.html')

def uploadImage(request):
    return render(request,'user/upload.html')

def homepage(request):
    return render(request,'user/userHome.html')

def home(request):
    if request.method == "GET":
        return render(request, 'home.html')

    if request.method == "POST":
        image = request.FILES['test1']

        user_id = request.session['userid']
        shutil.rmtree(os.getcwd() + '\\media')


        path = default_storage.save(
```

```
os.getcwd() + '\\media\\test.png', ContentFile(image.read()))
        result = predict.process()
        res_values = 1
        if result == "Chinkara":
            res_values = 0
        if result != "Chinkara":
            helpers.send_email_to_user(user_id)
            helpers.send_email_to_officer(user_id)

      return render(request, "user/result.html", {'result': result,'path':path,
'res': res_values})

def testagain(request):
    return render(request, "user/upload.html")

def camera(request):
    face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
    cap = cv2.VideoCapture(0)


      while True:
        ret, frame = cap.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
        minNeighbors=5, minSize=(30, 30))

        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

        cv2.imshow('Animal Detection', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
    return render(request, "user/userHome.html")
```

CHAPTER 7

SYSTEM TESTING

Testing is a process of executing a program with the explicit intention of finding error. It is a process used to identify correctness, completeness and quality of developed computer software.

7.1 White Box Testing:

White box testing strategy deals with the internal logic and structure of the code. It is also called as glass, structural, open and clear box testing. The test that are written based on the white box testing strategy incorporate coverage of the code written, branches, statements and internal logic of the code etc. In order to implement white box testing the tester has to deal with the code and hence it is required possess knowledge of the coding and logic i.e. Internal working of the code.

Advantages:

- As the knowledge of the internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively
- It helps in optimizing the code
- It helps in removing the extra line of code, which introduce defect in the code

Disadvantage

- As the knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, and this, in turn, increase the cost of the software.
- It is nearly impossible into every bit of code to find out the hidden errors, which may create problems, resulting in failure of application.

7.2 Black Box Testing

Black box testing takes the internal perspective of the test object to derived test cases. These tests can be functional or non-functional though usually

functional. The test designer selects valid and invalid inputs and determines the correct input. There is no knowledge of the test object's internal structure. This method of test design is applicable to all levels of software testing: unit, internal, functional and system and acceptance.

Advantages:

- Black box test are reproducible
- The environment in which the program is running is also tested
- The invested effort can be used multiple times

Disadvantages:

- The results are often over estimated
- Not all properties of the software can be tested
- The reason for failure is not found

7.3 Test Cases:

7.3.1 Login:

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
1	If user clicks on login button without entering username and password.	Inputs are not given	Please enter username and password	Please enter username and password	Pass
2	Enter correct username & Password and hit login button	username test@xyz.com Password: ******	Login success	Login success	Pass
3	If username is blank but password is entered.	Password: ******	Please enter username	Please enter username	Pass

4	If password is blank but username is entered.	Username: abcd	Please enter password	Please enter password	Pass
5	If the username or password is incorrect.	Username: abcd Password: *****	Invalid login id or Password	Invalid login id or Password	Pass

7.3.2 Registration:

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
1	When user click on the Create new account	Inputs are not given	Displays registration form	Displays registration form	Pass
2	Enters Name, Mobile No, email, Address, Gender, DOB, and clicks on signup.	Abcd 8989898989 abcd@gmail.com Mumbai Male 15/05/1998 avi	Successfully registered	Successfully registered	Pass
3	If any fields are blank	Xyz Shetty Password: *******	Please enter Mobile no	Please enter Mobile no	Pass

CHAPTER 8

RESULTS AND DISCUSSION

8.1 Login Page:

A login page is where users enter their credentials—usually a username or email and a password—to access a platform or service they've previously registered for. It's the gateway for users to gain authorized entry to their accounts. The login page typically consists of fields for username/email and password, sometimes accompanied by options like "Remember Me" or "Forgot Password."

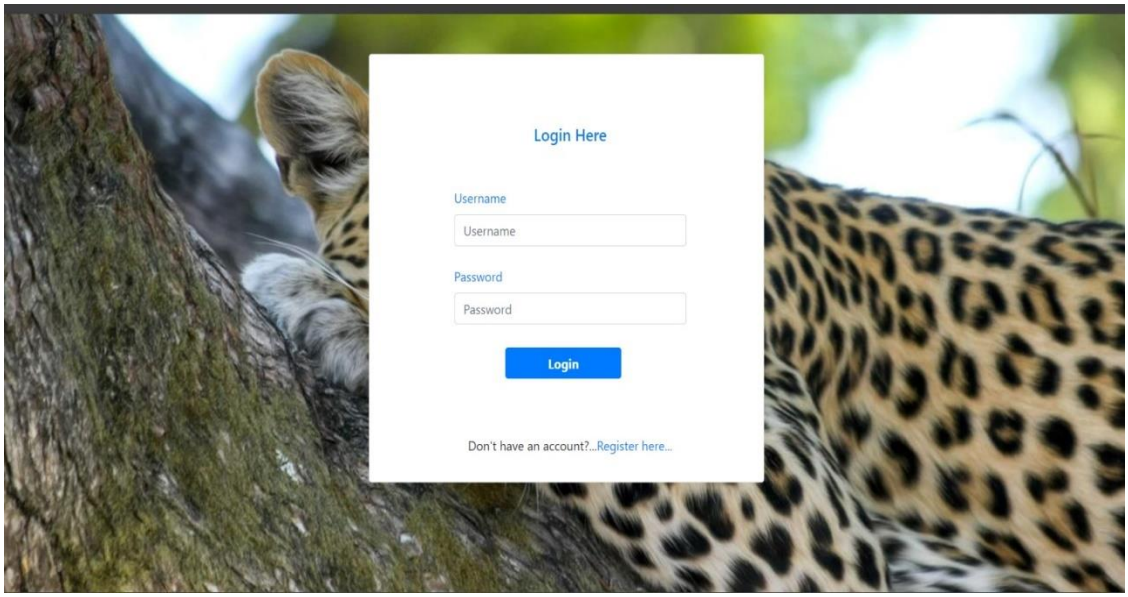

Figure 8.1 Login Page

8.2 Signup Page:

A signup page is a web page where users register or create an account to access a service, platform, or application. It typically includes a form where users input their information, such as name, contact number, email address, password, and sometimes additional details.

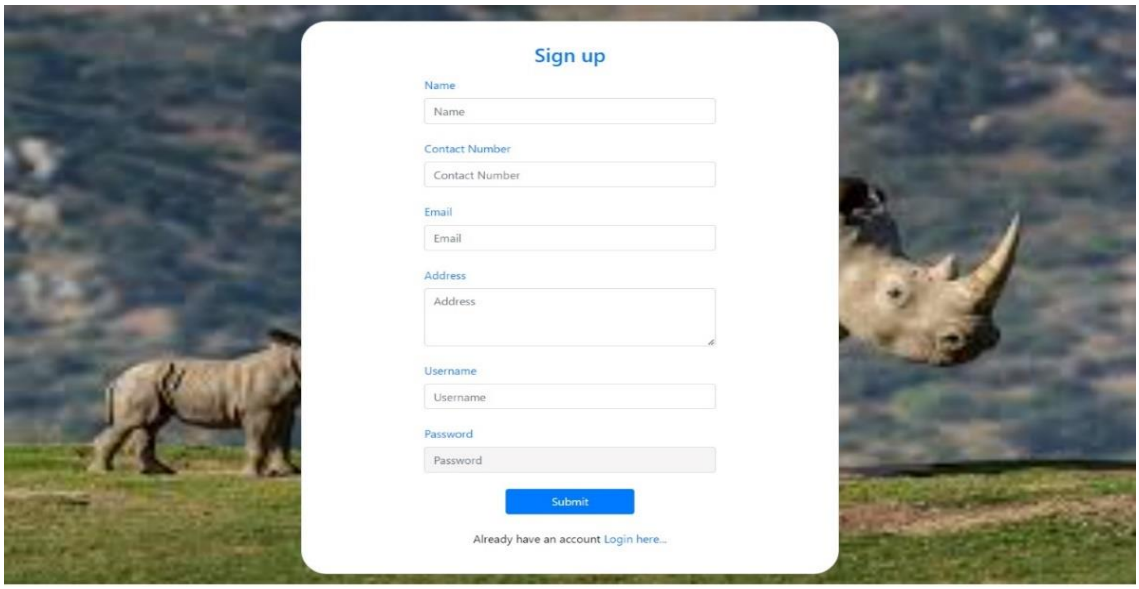

Figure 8.2 Signup Page

8.3 Upload Image Page:

User can Upload the images of the animals from the dataset and the image must be in the .jpg format. Once the image has been inserted click to submit button.

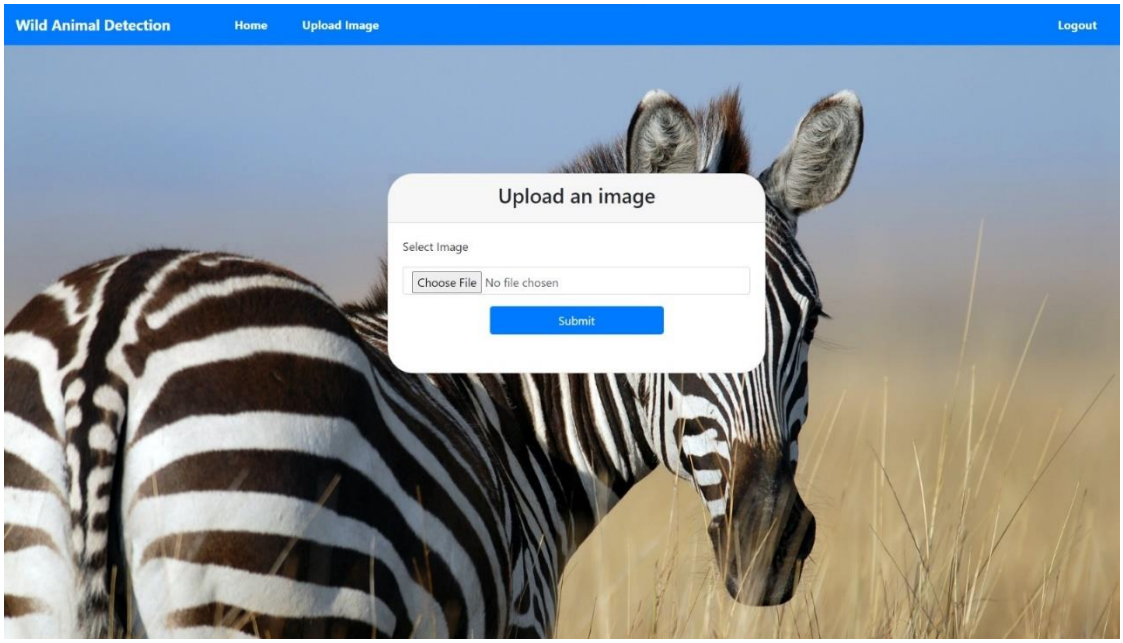

Figure 8.3 Upload Image Page

8.4 Result Page:

After uploading an image, the application will analyze it. Once identified, the application will display the name of the animal.

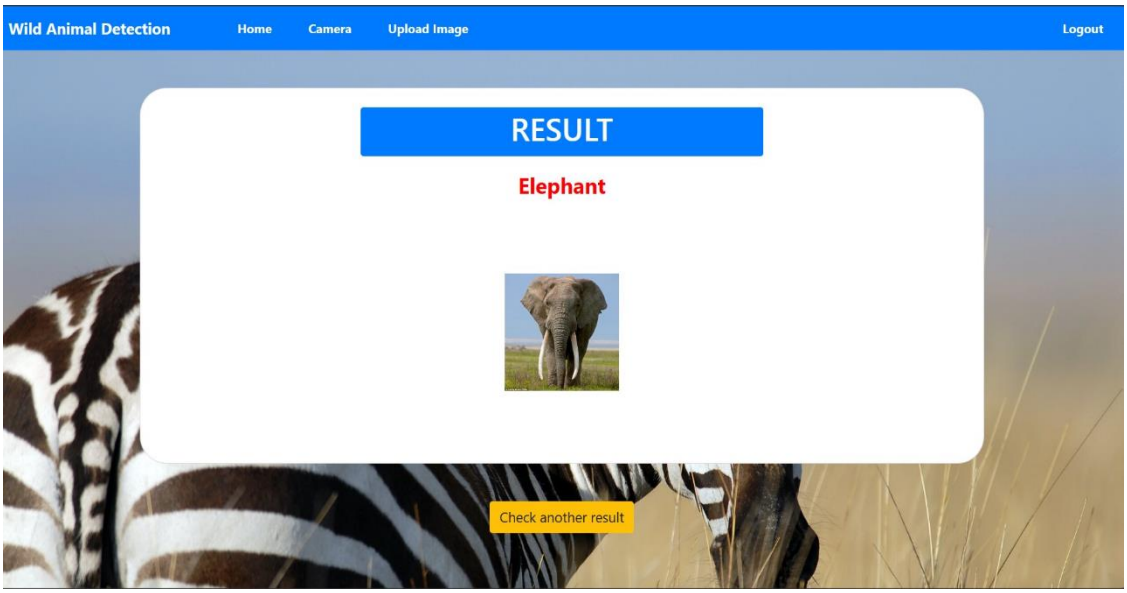

Figure 8.4 Result Page

8.5 Email to Farmer:

Once the wild animal is detected, an email is sent to the respective Farmer notifying them of the identification.

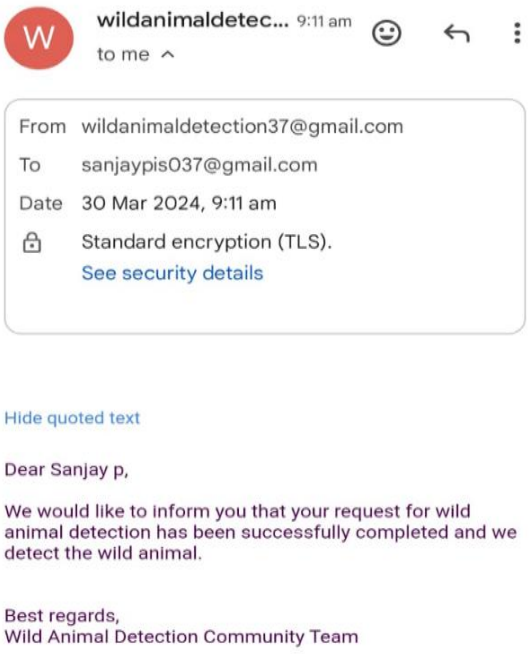

Figure 8.5 Email to Farmer

8.6 Email to Forest Officer:

Once the wild animal is detected, an email is sent to the respective forest officer notifying them of the identification.

Figure 8.6 Email to Forrest Officer

CHAPTER 9

SCOPE FOR FUTURE WORKS

- Utilizing advanced sensor technologies such as drones equipped with high-resolution cameras and LiDAR systems holds promise for enhancing wild animal detection in farmland.
- Integration of artificial intelligence and machine learning algorithms enables real-time analysis of sensor data, leading to early detection of pest infestations or crop damage caused by wildlife.
- Collaborative efforts between farmers, conservationists, and technology developers are essential for developing customized solutions tailored to the specific challenges of protecting crops while promoting biodiversity conservation.
- The convergence of technology-driven solutions and conservation objectives characterizes the future scope of wild animal detection in farmland.
- Application of remote sensing and satellite technology facilitates large-scale monitoring of wildlife populations and habitat use patterns.
- Advanced sensor technologies enable precise monitoring and management of wildlife interactions with crops, promoting efficient agricultural practices.
- Continuous innovation in sensor technology and data analytics holds the potential to revolutionize wildlife detection and management in farmland.
- Real-time analysis of wildlife behavior using advanced sensor data allows for proactive measures to prevent crop damage.
- Adoption of technology-driven solutions for wildlife detection contributes to sustainable agricultural practices by minimizing environmental impact.
- The interdisciplinary approach involving stakeholders from agriculture, conservation, and technology sectors fosters holistic solutions for balancing agricultural productivity with wildlife conservation efforts.

- Enhanced monitoring capabilities provided by advanced sensor technologies contribute to improved decision-making in farm management and land use planning.
- Continuous research and development efforts in wildlife detection technologies drive innovation and expand the possibilities for effective wildlife management in agricultural landscapes.
- The future of wild animal detection in farmland is characterized by the integration of cutting-edge technologies with conservation principles, fostering harmony between agricultural activities and wildlife conservation goals.

CHAPTER 10

CONCLUSION

Implementing a wild animal detection system in farmlands offers numerous advantages for both farmers and wildlife conservation efforts. Firstly, such a system can significantly reduce crop damage caused by wildlife. By providing early warnings when animals are detected in or near farmlands, farmers can take immediate action to protect their crops, such as using scare tactics or installing physical barriers. This can lead to a decrease in financial losses for farmers and a more sustainable agricultural practice. Moreover, these systems can help mitigate human-wildlife conflicts by reducing the need for lethal control measures. When farmers are alerted to the presence of wildlife, they can take non-lethal actions to deter animals, such as using noise-making devices or implementing temporary fencing. This can help prevent retaliatory killings of wildlife, which are often carried out when animals are perceived as threats to crops. Overall, implementing wild animal detection systems in farmlands represents a proactive and humane approach to managing human-wildlife conflicts. By reducing crop damage, preventing retaliatory killings, and providing valuable data for research, these systems benefit both farmers and wildlife, ultimately contributing to a more sustainable coexistence between humans and wildlife.

REFERENCES

[1] Santhiya S, Dhamodharan Y, Kavi Priya N E. Santhosh CS and Surekha M. 'A Smart Farmland Using Raspberry Pi Crop Prevention and Animal Intrusion Detection System'. International Research Journal of Engineering and Technology (IRJET), 2018; 05(03).

[2] Duhart C, Dublon G, Mayton B, and Paradiso J. 'Deep Learning Locally Trained Wildlife Sensing in Real Acoustic Wetland Environment'. In Thampi SM, Marques O, Krishnan S, Ciuonzo D and Kolekar MH (eds.), Advances in Signal Processing and Intelligent Recognition Systems, 2019: 3–14.

[3] Govind S.K., Jayson E.A. (2018) Crop Damage by Wild Animals in Thrissur District, Kerala, India. In: Sivaperuman C., Venkataraman K. (eds) Indian Hotspots. Springer, Singapore.

[4] A Case Study by Sujatha Kamepalli, Venkata Krishna Kishore Kolli, Srinivasa Rao Bandaru, Animal Breed Classification and Prediction Using Convolutional Neural Network Primates, 2021.

[5] WildDect -YOLO by Arunabha M. Roy, Jayabrata Bhaduri, Teerath Kumar, Kislay Raj, An efficient and robust computer vision -base d accurate object localization model for automated endangered wildlife detection, 2020.

APPENDIX A

PAPER PUBLICATION

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

Certificate of Publication

DHANUSH

Student, Dept. of Information Science & Engineering,

Mangalore Institute of Technology & Engineering, Moodabidre, India

Published a paper entitled

WILD ANIMAL DETECTION IN FARMLAND

Volume 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13392

Certificate# IJARCCE/2024/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

Tejass Publisheers
ORGANIZATION

Editor-in-Chief
IJARCCE

SERTIFIKAAT 証明書 شهادة Sertifikat CERTYFIKAT CERTIFICADO 증명서 СЕРТИФИКАТ 证书 POTVRDA SERTİFİKA

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

A monthly Peer-reviewed journal

Indexed by Google Scholar, Crossref and Mendeley

DOI IJARCCE/10.17148

www.ijarcce.com

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13392

WILD ANIMAL DETECTION IN FARMLAND

Manjunath Hebbagilu[1], Dhanush[2], Ramnath Nayak[3], Sahil Faraz[4], Sanjay P[5]

Professor, Dept. of Information Science & Engineering, Mangalore Institute of Technology & Engineering, Moodabidre, India[1]

Student, Dept. of Information Science & Engineering, Mangalore Institute of Technology & Engineering, Moodabidre, India[2-5]

Abstract: Animal assaults that cause crop damage are one of the main factors lowering agricultural yields. Crop raiding is turning into one of the most vexing human-wildlife conflicts as a result of the extension of farmed land into former animal habitat. India's farmers face significant risks from pests, natural disasters, and animal damage, which lowers production. In order to monitor crops and deter wild animals, farmers cannot afford to pay guards and their traditional tactics are not very efficient. Given the equal importance of ensuring the safety of humans and animals, it is crucial to safeguard crops from animal damage and safely redirect animals away from crops. Crop striking is turning into one of the most acrimonious human-wildlife conflicts due to the expansion of cultivated land into former animal habitat. It is essential to thoroughly and effectively verify that wild animals are allowed to remain in their natural habitat. Therefore, we employ deep learning to identify animals visiting our farm by applying the deep neural network idea, a branch of computer vision, in order to overcome the aforementioned issues and achieve our goal. This suggested system would use a camera to capture the surrounding area all day long and monitor the entire farm at predictable periods. When an animal enters the area, the system uses a deep learning model to recognise it and plays the proper noises to scare it away.

Keywords: Convolutional Neural network, Deep learning, Remote monitoring, Alert system.

I. INTRODUCTION

The various problems that arise when agriculture and wildlife coexist call for creative solutions. In addition to providing raw materials for other businesses, agriculture is the backbone of food production. On the other hand, crop productivity and public safety are seriously threatened by the existence of wild animals on agricultural land.

The harm that strays wild animals inflict to crops has become a major worry in many areas. Goats, cows, and wild buffalo are among the species infamous for their destructive foraging practices, which can occasionally cause human mortality. This puts farmers' lives, who depend on these crops for survival, at jeopardy in addition to causing financial losses. The entire agricultural productivity is severely harmed by animal intervention, especially for basic crops like potatoes and wheat.

Strict wildlife laws exacerbate the problem by frequently preventing small farmers from adopting practical precautions to safeguard their crops. Farmers may find themselves unable to fully handle the issue in spite of experiencing significant losses—up to 40–50% of their produce. Farmers feel vulnerable and helpless as a result of this regulatory restriction, which makes an already difficult situation worse.

Proactive steps to discourage wild animals are especially necessary in areas such as India where there is a high rate of conflict between humans and elephants. Elephants in particular have a reputation for seriously damaging property and crops, which exacerbates conflicts between local people and animals. To reduce these conflicts and protect lives and livelihoods, farmers need technology that makes it possible for them to quickly identify and react to animal invasions.

In addition, the situation is made worse by the deforestation caused by human growth, which leaves animals in their natural habitats with less access to food, water, and shelter. Conflicts between humans and animals increase when wild animals invade homes in quest of food, creating more difficulties for both communities and law enforcement.

Animal interference in agriculture has effects that go beyond short-term financial setbacks. Food insecurity is made worse by the ensuing rise in food costs, which disproportionately affects the poor and those who cannot purchase basic food items. Therefore, it is imperative to address the problem of wild animal detection on farms in order to maintain agricultural output as well as to guarantee that everyone in society has fair access to food.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13392

II. LITERATURE SURVEY

In [1] G. S. Gomez-Gomez et. al (2021) the research "Image-Based Animal Recognition based on Transfer Learning" shows how transfer learning may be used to effectively recognise animals, producing competitive results even with little data. It presents a new cow vs. non-cow database in addition to CIFAR, demonstrating a variety of architectures such as VGG-16, Google Net, and ResNet, demonstrating the effectiveness of the method in picture classification.

In [2] Tushar Atkare et. al (2021) the paper, highlights the move towards computer vision-based systems for quicker and more accurate detection while discussing several approaches for animal detection in agricultural settings. Techniques like FPGA implementations, complicated algorithms, and Python with TensorFlow and Google Vision API are mentioned. Sequence diagrams improve implementation efficiency by assisting with documentation and system comprehension.

In [3] Manasa Kommineni1 et. al (2022) the study suggests computer vision and machine learning system for agricultural animal detection is a viable way to protect crops from wildlife damage. It claims to reduce crop losses and environmental damage while being affordable and easily obtainable, thanks to its effective detection and alarm generation. It is consistent with the trend of using AI to improve agriculture.

In [4] R.S. Sabeenian et. al (2020) the review, addresses the drawbacks of conventional approaches to wildlife encroachment in agriculture and suggests AI-based alternatives that are more economical. It describes methods that have the potential to improve accuracy, but note additional hardware requirements. Examples of these methods are animal sound identification utilising signal processing and deep learning, and WCoHOG feature vectors using LIBLINEAR classifier.

In [5] T. Sandeep et. al (2022) this comprehensive review addresses the disastrous effects of agricultural depredation and emphasises the need of surveillance in farmlands to stop both human and animal incursion. It offers practical, non-lethal ways to protect crops while supporting farmers' decision-making processes by utilising edge computing, image processing platforms, and intelligent repelling systems.

In [6] Aniket Gat et. al (2022) this review various animal detection technologies, such as crop protection systems based on Raspberry Pi and distributed looking devices from ASFAR, are shown. These systems show good accuracy in identifying and tracking animals, promoting wildlife safety and conservation in various settings through the use of visual descriptors, gait analysis, and algorithms such as AdaBoost.

In [7] Abhineet Singh et. al (2020) this study emphasises transfer learning issues while examining deep learning for animal detection in human situations. It assesses the generalisation capabilities of eight detectors and suggests creating fake data for domain-specific training. In order to meet the constraints of real-time video sequences in animal recognition, the RETINA and YOLO models emerge as competitive options for multi-camera mobile deployment.

In [8] Kamepalli S et. al (2021) this study uses camera trap photos and deep learning, specifically CNNs, to present a bidirectional convolutional neural network for effective object detection, with a special emphasis on primate breed categorization. It demonstrates successful wildlife conservation applications with a trained model that achieves 80.5% accuracy on the training set and 73.53% on validation, backed by loss measures and sample photos.

In [9] Reddy B et. al (2021) this study creates a YOLOV3 model that detects animals by utilising a pre-trained coco dataset and the darknet method. Even with a few errors, it points to potential future gains with bespoke dataset training. It uses the Google API and the recognizer deep learning package with the Darknet-53 feature extractor to perform recognition. Matplotlib is used to visually represent the results.

In [10] Faseeha M et. al (2022) this research uses YOLOv3 to detect wild animals in agriculture in real-time with the goal of preventing dangers to rural areas and harm to crops. It proposes camera-based detection with farmer warnings to solve the limitations of infrared sensors and farming practices that are favourable to wildlife. The object grouping frameworks and loss function of YOLOv3 improve the precision of animal identification and categorization.

III. SCOPE AND METHODOLOGY

Aim of the project

The Aim of farmland wild animal detection is to create and put into place efficient technologies or systems that can quickly detect the presence of wild animals in agricultural regions. With the use of this technology, farmers should be able to identify incursions with greater speed and accuracy, enabling them to take prompt action to prevent or lessen any

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13392

wildlife harm. The objective is to safeguard human life, reduce crop losses, and advance the sustainable coexistence of agriculture and wildlife by accomplishing this goal. Furthermore, the objective includes mitigating human-animal conflicts, improving food security, and safeguarding agricultural landscape ecosystems.

Existing system

In essence, the observation is made helpful by the existing systems. Furthermore, especially in an application area like this, these systems do not provide security against wild animals. Additionally, as different strategies are used to prevent different creatures from accessing such limited areas, they must act based on the type of animal that tries to enter the area. In a similar vein, the ranchers use other methods, such as keeping human replicas and sorts on their homesteads, which work slightly to deter birds but are ineffective at keeping out wild animals. In order to prevent animals from damaging their harvest, ranchers often employ many dangerous and comprehensive measures such as building physical barriers, using electric walls, conducting manual reconnaissance, and more.

Proposed system

Model Creation, Pre-processing, System Training, and Classification are the four primary components that make up the system. By scaling the dataset photographs to the necessary dimensions, the pre-processing module is utilised to pre-process them.

To create a machine language model with the necessary number of layers, utilise the Model Creation module. To train the system using dataset photos and store the model weight, utilise the system training phase. To identify the animal from the supplied test picture, a classification module is employed. The technology notifies the farm owner via email if it is caught as a wild animal.

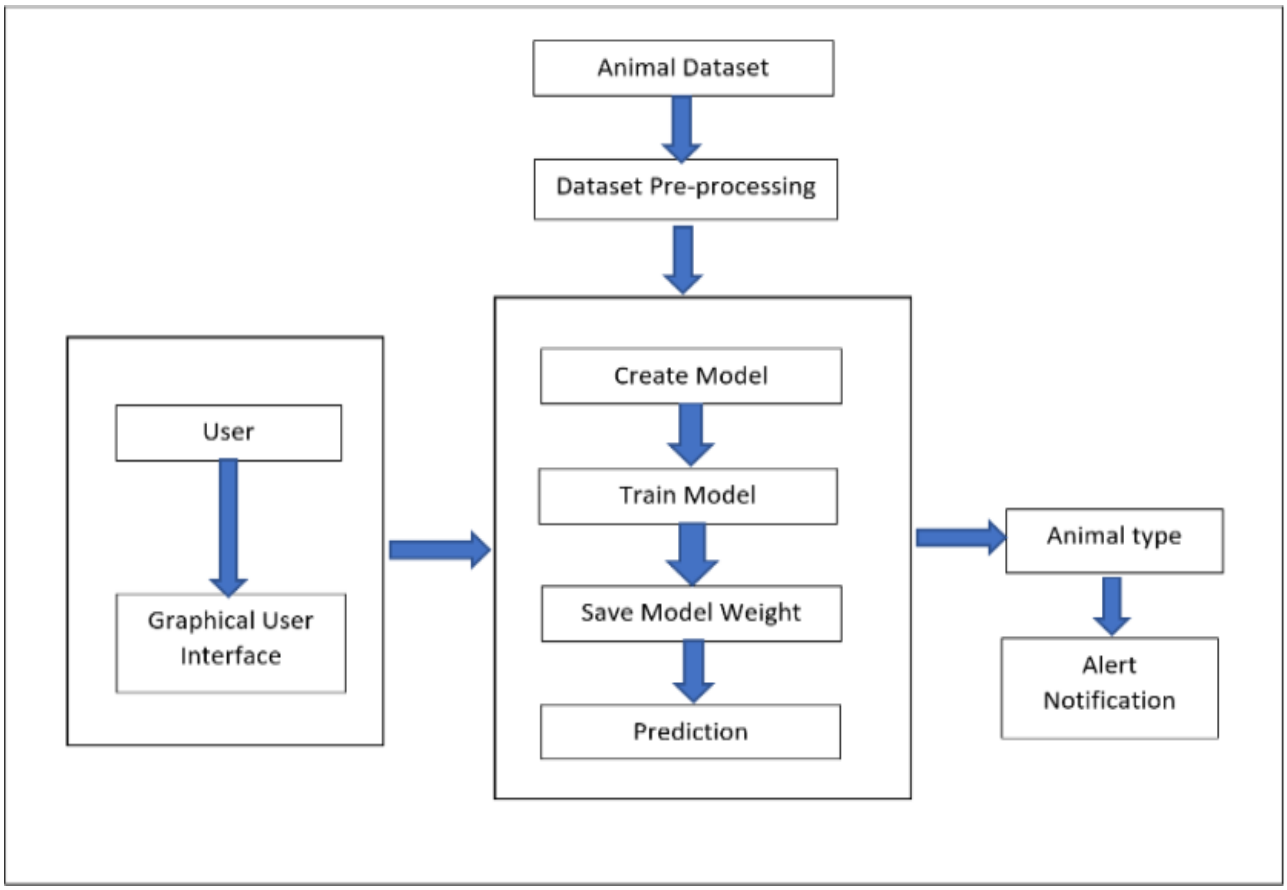

Fig 1. Proposed system

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13392

System Architecture

A formal explanation of a system is provided by an architectural explanation, which is arranged to support reasoning regarding the system's structure, individual components' properties that can be observed from the outside, and the interactions between them. It also offers a framework from which systems can be developed and products acquired that will cooperate to implement the system as a whole.

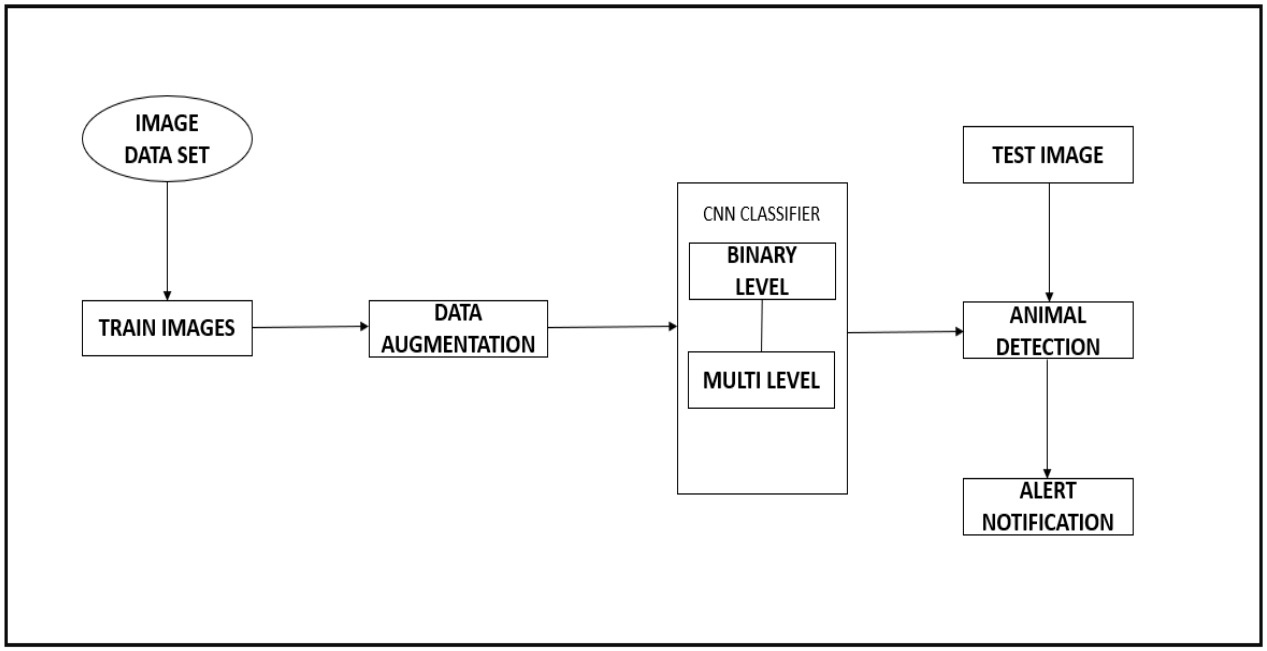

Fig 2. System Architecture

The wild animal detection system for agriculture is made up of multiple interrelated components that detect, identify, and respond to animal invasions in a timely and effective manner. The system achieves its goals by integrating alert technologies, data processing algorithms, and reaction mechanisms. The system then combines a variety of reaction mechanisms to efficiently prevent or reduce animal invasions. These can include automated deterrents like sound alarms. In more extreme circumstances, the system can send alarms to human operators or wildlife authorities requiring rapid action.

The architectural description provided above describes the structure, components, and interactions of a system for detecting wild animals in agriculture. The system attempts to efficiently identify and reduce animal invasions while minimising disturbance to farm activities by utilising alert technologies, data processing algorithms, and reaction mechanisms. Continuous development and adaptability are critical ideas driving the growth of the system architecture to handle emerging difficulties and increase its efficacy in protecting farmland and encouraging coexistence between agriculture and wildlife.

IV. CONCLUSION

In conclusion, there are a variety of issues that arise from the coexistence of wild animals on agricultural land, from possible safety risks to farmers suffering financial losses. Particularly in areas where human-animal conflicts are common, it is obvious that effective technology is needed to identify and prevent these creatures. By preventing food prices from rising and safeguarding food supplies for disadvantaged groups, such technology would safeguard livelihoods and agricultural productivity.

Long-term solutions further depend on addressing the underlying causes of wildlife infiltration, such as habitat degradation from deforestation. We may work to ensure cohabitation between people and animals while maintaining food security and ecological balance by encouraging creative thinking and fusing conservation initiatives with agricultural methods.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2024

DOI: 10.17148/IJARCCE.2024.13392

REFERENCES

[1] G. S. Gomez-Gomez, D. Collazos-Huertas, A. M. Álvarez-Meza, "Image-Based Animal Recognition based on Transfer Learning" (2021).

[2] Tushar Atkare, Snehal Khalkar, Pallavi Mandekar, Ganesh Kolhe, Anita Gawali "Wild Animal Detection in Farm using Image Processing and Machine Learning" (2021).

[3] Manasa Kommineni1, M. Lavanya1, V. Harsha Vardhan1, Jagadeesh Kumar, G Shamsheer Shaik V, T. Gouri, Sankar "Agricultural Farms Utilizing Computer Vision (AI) And Machine Learning Techniques For Animal Detection And Alarm Systems" (2022).

[4] DR.R.S. Sabeenian ,N. Deivanai, B.Mythil "Wild Animals Intrusion Detection using Deep Learning Techniques" (2020).

[5] T. Sandeep, B. Manushree, S. Rahul, T. Bharath "Crop Protection from Animals using CNN" (2022).

[6] Aniket Gat, Hrishikesh Gaikwad, Rahul Giri, Dr. Mohini P Sardey, Milind P Gajare "Animal Detector System for Forest Monitoring Using OpenCV and Raspberry-pi" (2022).

[7] Abhineet Singh, Marcin Pietrasik, Gabriell Natha, Nehla Ghouaiel, Ken Brizel, Nilanjan Ray "Animal Detection in Man-made Environments" (2020).

[8] Kamepalli S, Kolli V, Bandaru S "Animal Breed Classification and Prediction Using Convolutional Neural Network Primates as a Case Study" (2021).

[9] Reddy B, Bano S, Reddy G "Convolutional Network based Animal Recognition using YOLO and Darknet" (2021).

[10] Faseeha M, Jacob A, Scholar M "Wild Animal Detection in Agricultural Farms using Deep Convolutional Neural Network" (2022).

