CHAPTER 1

INTRODUCTION

1.1. Overview

Agriculture has seen many revolutions, whether the domestication of animals and plants a few thousand years ago, the systematic use of crop rotations and other improvements in farming practice a few hundred years ago, or the "green revolution" with systematic breeding and the widespread use of man-made fertilizers and pesticides a few decades ago.

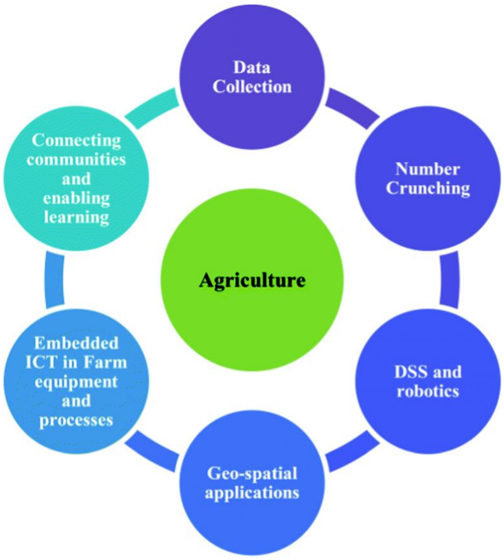


Figure no: 1.1. Agriculture and ICT Innovation

Agriculture is undergoing a fourth revolution triggered by the exponentially increasing use of information and communication technology (ICT) in agriculture. Autonomous, robotic vehicles have been developed for farming purposes, such as mechanical weeding, application of fertilizer, or harvesting of fruits. The development of unmanned aerial vehicles with autonomous flight control, together with the development of lightweight and powerful hyperspectral snapshot cameras that can be used to calculate biomass development and fertilization status of crops, opens the field for sophisticated farm management advice. Moreover, decision-tree models are available now that allow farmers to differentiate between plant diseases based on optical information. Virtual fence technologies allow cattle herd management based on remote-sensing signals and sensors or actuators attached to the livestock.

Taken together, these technical improvements constitute a technical revolution that will generate disruptive changes in agricultural practices. This trend holds for farming not only in developed countries but also in developing countries, where deployments in ICT (e.g., use of mobile phones, access to the Internet) are being adopted at a rapid pace and could become the game-changers in the future (e.g., in the form of seasonal drought forecasts, climate-smart agriculture).

1.2. Smart Farming

Smart farming is a management concept focused on providing the agricultural industry with the infrastructure to leverage advanced technology – including big data, the cloud and the internet of things (IoT) – for tracking, monitoring, automating and analysing operations.

Figure no: 1.2. Smart Farming

Also known as precision agriculture, smart farming is software-managed and sensor-monitored. Smart farming is growing in importance due to the combination of the expanding global population, the increasing demand for higher crop yield, the need to use natural resources efficiently, the rising use and sophistication of information and communication technology and the increasing need for climate-smart agriculture.

1.2.1. Smart farming technologies

The intelligent farm includes the use of technology such as:

- Sensors for soil scanning and water, light, humidity and temperature management.
- Telecommunications technologies such as advanced networking and GPS.

- Hardware and software for specialized applications and for enabling IoT-based solutions, robotics and automation.
- Data analytics tools for decision making and prediction. Data collection is a significant part of smart farming as the quantity of data available from crop yields, soil-mapping, climate change, fertilizer applications, weather data, machinery and animal health continues to escalate.
- Satellites and drones for gathering data around the clock for an entire field. This information is forwarded to IT systems for tracking and analysis to give an "eye in the field" or "eye in the barn" that makes remote monitoring possible.

The combination of these technologies facilitates machine-to-machine (M2M) derived data. This data feeds into a decision support system so that farmers can see what is happening at a more granular level than in the past. For example, by precisely measuring variations within a field and adapting the strategy accordingly, farmers can greatly increase the effectiveness of pesticides and fertilizers and use them more judiciously. Similarly, smart farming techniques, help farmers better monitor the needs of individual animals and adjust their nutrition to prevent disease and enhance herd health.

1.3. Problem Identified

Most farmers have challenges related to crop damage due to wildlife pests. Animal intrusion is a major threat to the productivity of the crops, which affects food security and reduces the profit to the farmers. Organic farmers have additional challenges because they cannot use chemical controls which are sometimes the most effective and efficient options. A need has been identified for alternative pest control appropriate for traditional and organic farmers. Three types of animal intrusion you might find include animal tracks, crop damage and animal scat or faeces. In the case of animal tracks, only one instance of tracks in the field carries a relatively low risk.

On the other hand, sporadic or widespread animal tracks carry a moderate risk, and a no-harvest buffer zone may need to be created around nearby crops. Crop damage, such as bite marks or trampled plants, is riskier than animal tracks. Sporadic evidence, such as a few observations of trampled plants throughout the field, is moderately risky. Widespread crop damage is a high risk and indicates significant evidence of contamination. Marking and avoiding harvest around high-risk areas of crop damage is a good strategy to reduce the potential for contamination. Risks associated with faecal matter in the field are the highest.

For even just one instance of faecal matter, the risk of contamination is moderate. Widespread evidence of faecal contamination is very high risk and would justify marking the contaminated area and creating a no-harvest buffer zone around the area where significant faeces was found. Animal activity on the farm can be a huge risk to food safety when growing fruits and vegetables, which is why prearrest wildlife scouting is so important. Existing methods like fencing can be an effective deterrent, but it may not be practical for larger farms; however, small portions of fencing may direct animals around high value or sensitive crops to other areas and electric fences are no longer efficient in solving such conflicts, to protect their crops from getting damaged because of animal intrusions, farmers have been using electric fences around their fields and areas where the fencing don't prove efficient, farmers prefer to stay up all night and guard their fields from animal intrusions.

Nuisance permits may be another option, but check with local Department of Environmental Conservation (DEC) or the National Resources Conservation Service (NRCS) before choosing this method. Practices like these have done more harm than good for us and in extreme cases; it has even costed lives of both man and animals. So, we decided to come up with a smarter solution that could protect the crops from animal intrusions without causing any harm to the wildlife. To solve this conflict by using technology such as IoT and Deep learning, which is called AIoT (Artificial Intelligence for the Infrastructure of Internet of Things)?

1.3.1. Scope of the Project

The product has been iterated over a period of time. Some of the innovative ways in which this product stands out are:

1. Usage of a combination of Camera Vision to detect intrusion of any animal entering the farm
2. Usage of the electrical signal to trigger an alarm that repels the detected animal away.
3. Solar-powered set-up
4. Since animals tend to get acquainted to the recurring sounds and lights in their surroundings, ordinary alarming systems will work in the beginning but its effectiveness will decrease sharply with time. To overcome this shortcoming, AniRep uses permutations and combinations of Vision and sound patterns those changes every time an intrusion is detected. This delays habituation in animals. As our system is deployed in rural areas, where energy efficiency is essential, we evaluated some HW and SW architectural alternatives for the

design and implementation of the AI embedded Edge computing components to be added to our system.

1.4. Objective

The main aim of the current work is to develop a device to protect crops from damage by wild animals by diverting them from the farms, without harming them physically. The objective of the project is to design, deployment and assessment of an intelligent smart agriculture repelling and monitoring IoT system based on embedded edge AI, to detect and recognize the animals, as well as generate ultrasonic signals tailored to each species of the animals.

Untamed life checking and investigation are in dynamic research fields since last numerous decades. This work mainly concentrates on creature identification from common scenes gained by camera and intimates the farmland owner and also stored the images in the camera trap database. This camera trap database is utilized to track the animals which enter into the farm land and damaging the crops.

CHAPTER 2

LITERATURE SURVEY

1. Workflow and convolutional neural network for automated identification of animal sounds [1]

Authors: Zachary J.Ruff, Damon B.Lesmeister, Cara L.Appel, Christopher M.Sullivan

Objective:

The aim of this project is automated identification of animal sounds using Convolutional Neural Networks.

Methodology:

The use of passive acoustic monitoring in wildlife ecology has increased dramatically in recent years as researchers take advantage of improvements in autonomous recording units and analytical methods. These technologies have allowed researchers to collect large quantities of acoustic data which must then be processed to extract meaningful information, e.g., target species detections. A persistent issue in acoustic monitoring is the challenge of efficiently automating the detection of species of interest, and deep learning has emerged as a powerful approach to accomplish this task. This article reports on the development and application of a deep convolutional neural network for the automated detection of 14 forest-adapted birds and mammals by classifying spectrogram images generated from short audio clips. This article proposes a multi-step workflow that integrates this neural network to efficiently process large volumes of audio data with a combination of automated detection and human review. This workflow reduces the necessary human effort by > 99% compared to full manual review of the data. As an optional component of this workflow, this article proposed a graphical interface for the neural network that can be run through RStudio using the Shiny package, creating a portable and user-friendly way for field biologists and managers to efficiently process audio data and detect these target species close to the point of collection and with minimal delays using consumer-grade computers.

Figure no:2.1. Steps involved in audio processing and target species detection workflow.

Merits:

- Accuracy is high.
- Minimal delays using consumer-grade computers.

- Its Efficiency is high.

Demerits:

- High power to detect these species.
- Combined with a low proportion of false positives.

2.Real-Time Monitoring of Agricultural Land with Crop Prediction and Animal Intrusion Prevention using Internet of Things and Machine Learning at Edge [2]

Authors: R. Nikhil; B.S. Anisha; Ramakanth Kumar P.

Objective:

The aim of this project is crop prediction helps the farmers to grow suitable crops depending on the soil parameters by the use of machine learning techniques and it also helps in prevention of the intruders like wild animals into the field.

Methodology:

The implemented smart agriculture system is cost effective for maximizing agricultural farm water supplies, crop prediction, and wild animal prevention. Depending on the level of soil moisture, the proposed system can be used to turn the water sprinkler on / off, thereby making the process easier to use. The system proposed can be used to predict the crop based on the soil condition which helps the farmer grow the proper crops at proper time. Through this system it can be inferred that use of IOT and Automation there by achieving significant progress in irrigation. The proposed system is thus a solution to the problems facing in current irrigation cycle. The proposed system also helps in the prevention of trespassing wild animals in the agricultural sector. Using ultrasonic sound, the buzzer irritates wild animals and makes them leave the area. Using the alarm tone flooding techniques which requires less energy In addition to that the device is eco-friendly, because there is no harm to the ecosystem and no disruption to humans.

Merits:

- There is no harm to the ecosystem.
- No disruption to humans.
- Using the alarm tone flooding techniques which requires less energy.

Demerits:

- Lacks in monitoring.
- Accuracy is low.

3.Animal Behaviour Prediction with Long Short-Term Memory [3]

Authors: Henry Roberts; Aviv Segev

Objective:

The aim of this project is efficiently converting video of animals at any length into models capable of making accurate behavioural prediction using Long Short-term Memory (LSTM).

Methodology:

A foundational step of any animal is the establishment of an accurate behavioural model. Building a model that is capable of defining and predicting an animal's behaviour is critical to advancing ethological theory and research, however many animal models fail to be sufficiently thorough or often do not exist at all. Great pools of data are available for improving these models through recorded video of animals posted on video hosting sites throughout the internet, however these sources are largely left unused due to their sheer quantity being too much for researchers to manually observe and annotate. This article proposed a pipeline approach for efficiently developing predictive behavioural models using a confluence of machine learning tools. Accuracy in prediction and its significance against a much longer standing time-series analysis statistical model. The results of testing proposed pipeline showed promise in that the LSTM network, trained on the JAABA annotated frame of animal behaviour and classifier function results, was able to outperform the ARIMA model.

Merits:

- Its prediction applied to large datasets and yield useful results.
- Model's accuracy in prediction

Demerits:

- Time-series analysis statistical model is high.
- Its efficiency is low.

4. IoT based animal classification system using convolutional neural network [4]

Authors: L. G. C. Vithakshana; W. G. D. M. Samankula

Objective:

The aim of this project is an IoT based acoustic classification system was designed using Convolutional Neural Networks (CNN), which is beneficial for those who are interested in monitoring ecosystems such as animal scientists, zoologists, and environmentalists.

Methodology:

The kingdom “Animalia” is used to represent all living creatures on the planet earth, which is fallen into six categories. The language is the most common factor to divide humans and animals. Numerous classification techniques can be used for classification purposes, and the classification commonly can be done acoustically and visually. The classification systems are playing a considerable role, and bioacoustics monitoring was a significant field. Visual classification of animals is done by using either satellite images or established camera images. Nevertheless, due to some circumstances, image processing techniques cannot be applied. Then the acoustical classification techniques are taken place to encounter those problems. Even with acoustical methods, a remote observing method is required due to a few issues. Applying an IoT based acoustic classification system was designed using Convolutional Neural Networks (CNN), which is beneficial for those who are interested in monitoring ecosystems such as animal scientists, zoologists, and environmentalists. The hardware implementation was designed to collect the data from the place it was placed. The audio clips were pre-processed using the Mel frequency Cepstral Coefficient (MFCC). A CNN architecture based on Tensor Flow was used for the training process

Merits:

- Accuracy is high.
- Time consumption is low.
- Energy consumption is low.

Demerits:

- Cost is high.
- Efficiency is low.

5. Animal Sound Classification Using Dissimilarity Spaces [5]

Authors: Loris Nanni , Sheryl Brahnam , Alessandra Lumini and Gianluca Maguolo

Objective:

The aim of this project is to automated animal audio classification using clustering methods.

Methodology:

This article presents a method for classifying animal vocalizations using four Siamese networks and dissimilarity spaces. Different clustering techniques taking both a supervised and unsupervised approach were used for dissimilarity space generation. A compact descriptor is obtained by projecting each pattern into the dissimilarity spaces generated by the clustering methods using different numbers of centroids and the outputs of the four Siamese networks. The classification step is performed using a set of SVMs trained from such descriptors and combined by sum rule to obtain a highly competitive ensemble as tested on two datasets of animal vocalizations. In addition, experimental results demonstrated the diversity between the proposed approach and other state-of-the-art approaches can be exploited in an ensemble to further improve classification performance. The fusions improved performance on both audio classification problems, outperforming the standalone approaches.

Merits:

- Performance on both audio classification problems.
- It outperforming the standalone approaches.

Demerits:

- Time consumption is high.
- Accuracy for sound classification is low.

CHAPTER 3

SYSTEM ANALYSIS

3.1. Existing System

Wild animals are a special challenge for farmers throughout the world. Animals such as deer, wild boars, rabbits, moles, elephants, monkeys, and many others may cause serious damage to crops. They can damage the plants by feeding on plant parts or simply by running over the field and trampling over the crops. Therefore, wild animals may easily cause significant yield losses and provoke additional financial problems. Another aspect to consider is that wild animal crop protection requires a particularly cautious approach. In other words, while utilizing his crop production, every farmer should be aware and take into consideration the fact that animals are living beings and need to be protected from any potential suffering.

Farmers Traditional Approach

There are different existing approaches to address this problem which can be lethal (e.g., shooting, trapping) and non-lethal (e.g., scarecrow, chemical repellents, organic substances, mesh, or electric fences), firecrackers, bright lights, fire, beating drums, and dogs. Non-chemical control of pocket gophers. 22 rimfire rifle or a shotgun can be used to dispatch woodchucks. Some motion-activated water sprayers have been developed that spray birds when they break the motion-detecting

1. Agricultural fences

Fencing is a popular wild animal protection practice for that can last for many years. Agricultural fences are quite an effective wild animal protection technology. However, utilizing fences as a practice is often regulated. Some local and state entities may restrict or prevent the use of certain types of fences. Therefore, before deciding on a suitable fence, it's important to check local law regulations. The quality of fencing depends on the material and structure. Depending on how it is made and what it is made of, some permanent fences can last up to 30 years. Farmers usually use one of the following types of fences.

Wire fences

Constructed of metal wires woven together forming a physical barrier. The fences are effective, long lasting, and require relatively little maintenance. However, they are expensive and recommended only for the protection of high-value crops.

Plastic fences

Polypropylene fences are generally less expensive and easier to install and repair than other types. Additionally, these fences are widely acceptable and meet various regulations. Their disadvantage includes their short lifespan (up to 10 years) and questionable effectiveness in areas with a higher possibility of wild animal crop damage.

Figure no: 3.1. Wire and Plastic Fence

Electric fences

These are constructed to inflict an electric shock to animals that come in contact with the fence, thus preventing animals from crossing the fence. These fences are long lasting and an effective crop protection measure. Costs vary depending on specific type and size of an area. Before purchasing electric fences, it's very important to make sure they are allowed for use in the specific area, and for protection against endangered animal species. Additionally, it's recommended that electric fences are marked with a warning sign to prevent any possible human contact.

Figure no: 3.2. Electric Fences

2. Natural Repellents

Some farmers prefer using natural protection measures instead of mechanical or chemical protective practices. There are various ways to protect crops from wild animals, including.

Smoke

In some areas farmers burn elephant dung or other materials that smolder and create heavy smoke

Figure no: 3.3. Beehive fence

1. **Chemical repellents.**

Active substances such as Anthraquinonoid, Butane thiol, and Methyl Anthranilate can be used to keep wild animals away from crops

2. **Biophysical barriers**

Fences made of bamboo sticks, coconut tree bunches, or some other available shrubs; low-cost practice but also low efficiency in protecting crops against wild animals

3. **Electronic repellents**

Effective, long lasting, and eco-friendly method for crop protection that repels animals without harming them. Farmers use one of the following two types of electronic repellents.

Figure no: 3.4. Ultrasonic Electric Animal Repellent

Ultrasonic electronic repellent

Silent to humans, high-frequency sound waves repel wild animals

Sonic electronic repellent

Audible noise that scares animals

Hog

Weighted Co- occurrence Histograms of Oriented Gradients (W-CoHOG) feature vector to recognize animal. Histogram equalization is performed to reduce noise, distortions and to enhance the highlighted region of interest. The gradients are calculated in magnitude and direction is represents to convert into eight orientations. Sliding window techniques identify animals in different sizes with zoom level of the camera.

LBP and SIFT

Automated species recognition method using local cell-structured LBP (Local Binary Pattern) feature and global dense SIFT (Scale- invariant Feature Transform) descriptor for feature extraction and improvise (ScSPM) sparse coding spatial pyramid matching to extract dense SIFT descriptor and cell-structured LBP as a local feature. Global features generate max pooling and weighted sparse coding using multi-scale pyramid kernel.

SVM

Animal intrusion detection system based on image processing and machine learning approach. The image of an animal is segmented using a watershed algorithm to extract various objects in the image and to examine that if any threat animal is found in segmentation. This algorithm is to create a barrier which is the contour only when the marked region meets different markers. Gabor filter is extensively used in extracting a region with text to recognize facial expression in various frequencies. Linear SVM is a supervised learning algorithm to train the dataset and to classify text and hypertext.

Disadvantages

- Its disadvantages include the potential for the entire fence to be disabled due to a break in the conducting wire, shorting out if the conducting wire contacts any non-electrified component that may make up the rest of the fence, power failure, or forced disconnection due to the risk of fires starting by dry vegetation touching an electrified

- Other disadvantages can be lack of visibility and the potential to shock an unsuspecting human passer-by who might accidentally touch or brush the fence
- Bee fence disadvantages are that it is only restricted to elephants and humans can also become targets of the bees
- Percentage of all intrusions in the detection area that was detected was relatively low
- Sensor Failure
- Expensive

3.2. Proposed System

AI Computer Vision based DCNN for detecting animal species, and specific ultrasound emission (i.e., different for each species) for repelling them. Design, deployment and assessment of an intelligent smart agriculture repelling and monitoring IoT system based on embedded edge AI, to detect and recognize the different kinds of animal, as well as generate ultrasonic signals tailored to each species of the animal. This combined technology used can help farmers and agronomists in their decision making and management process.
Deep learning in the form of Convolutional Neural Networks (CNNs) to perform the animal recognition.

DCNN

CNNs are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. CNNs are a type of feed-forward neural networks made up of many layers. CNNs consist of filters or kernels or neurons that have learnable weights or parameters and biases. Each filter takes some inputs, performs convolution and optionally follows it with a non-linearity. A typical CNN architecture can be seen as shown in Fig.3.1. The structure of CNN contains Convolutional, pooling, Rectified Linear Unit (ReLU), and Fully Connected layers.

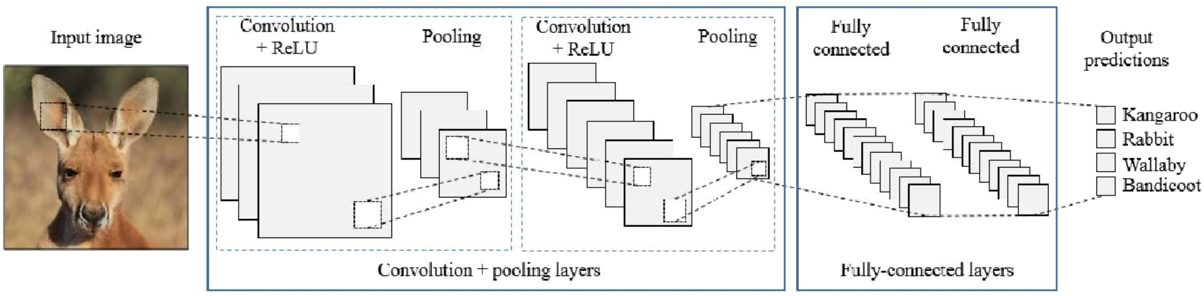


Figure no: 3.5. CNN

A. Convolutional Layer:

Convolutional layer performs the core building block of a Convolutional Network that does most of the computational heavy lifting. The primary purpose of Convolution layer is to extract features from the input data which is an image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input image. The input image is convoluted by employing a set of learnable neurons. This produces a feature map or activation map in the output image and after that the feature maps are fed as input data to the next convolutional layer.

B. Pooling Layer:

Pooling layer reduces the dimensionality of each activation map but continues to have the most important information. The input images are divided into a set of non-overlapping rectangles. Each region is down-sampled by a non-linear operation such as average or maximum. This layer achieves better generalization, faster convergence, robust to translation and distortion and is usually placed between convolutional layers.

C. ReLU Layer:

ReLU is a non-linear operation and includes units employing the rectifier. It is an element wise operation that means it is applied per pixel and reconstitutes all negative values in the feature map by zero. In order to understand how the ReLU operates, we assume that there is a neuron input given as x and from that the rectifier is defined as f(x) = max (0, x) in the literature for neural networks.

D. Fully Connected Layer:

Fully Connected Layer (FCL) term refers to that every filter in the previous layer is connected to every filter in the next layer. The output from the convolutional, pooling, and ReLU layers are embodiments of high-level features of the input image. The goal of employing the FCL is to employ these features for classifying the input image into various classes based on the training dataset. FCL is regarded as final pooling layer feeding the features to a classifier that uses SoftMax activation function. The sum of output probabilities from the Fully Connected Layer is 1. This is ensured by using the SoftMax as the activation function. The SoftMax function takes a vector of arbitrary real-valued scores and squashes it to a vector of values between zero and one that sum to one.

Generation of Repelling Ultrasound

Animals generally have a sound sensitive threshold that is far higher than humans. They can hear sounds having lower frequencies with respect to the human ear. For instance, while the audible range for humans is from 64Hz - 23KHz, the corresponding range of goats, sheep, domestic pigs, dogs and cats is 78Hz - 37KHz, 10Hz - 30KHz, 42Hz - 40.5KHz 67Hz - 45KHz and 45Hz - 64KHz.Generating ultrasounds within the critical perceptible range causes animals to be disturbed, thus making them move away from the sound source. At the same time, these ultrasounds are not problems to the human ear even when the frequency range is beyond the human ear.

Notification System

The detection system recorded the date and time of each detection. In addition, there were cameras and a video recording system that recorded all animal movements within the enclosure. The detection log was compared to the images from the cameras, which also had a date and time stamp, to investigate the reliability of the system. A message alert is sent to the registered mobile number.

Advantages

- Wide area surveillance
- Accurate and Fast prediction
- Cost effectiveness of available Crop protection systems.
- Easy to use and with less maintenance.
- Robust and reliable system.
- Complete security or full proof system.
- Less or no labor requirement.
- Easily adaptable by the farmers
- Remote Monitor
- Low energy consumption
- Warns and tracks
- Fully automated system
- Integra table with third-party cameras

CHAPTER 4

SYSTEM SPECIFICATION

4.1 Hardware Specification

- Processors: Intel® Core™ i5 processor 4300M at 2.60 GHz or 2.59 GHz
  (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM
- Disk space: 320 GB
- Operating systems: Windows® 10, macOS*, and Linux*

4.2 Software Specification

- Server Side : Python 3.7.4(64-bit) or (32-bit)
- Client Side : HTML, CSS, Bootstrap
- IDE : Flask 1.1.1
- Back end : MySQL 5.
- Server : Wampserver 2i
- OS : Windows 10 64 –bit or Ubuntu 18.04 LTS “Bionic Beaver”

CHAPTER 5

SOFTWARE DESCRIPTION

5.1. FRONT END

PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, and Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following.

TENSOR FLOW

Tensor Flow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and gives developers the ability to easily build and deploy ML-powered applications.

Tensor Flow provides a collection of workflows with intuitive, high-level APIs for both beginners and experts to create machine learning models in numerous languages. Developers have the option to deploy models on a number of platforms such as on servers, in the cloud, on mobile and edge devices, in browsers, and on many other JavaScript platforms. This enables developers to go from model building and training to deployment much more easily.

KERAS

Keras is a deep learning API written in Python, running on top of the machine learning platform Tensor Flow. It was developed with a focus on enabling fast experimentation.

- Allows the same code to run on CPU or on GPU, seamlessly.
- User-friendly API which makes it easy to quickly prototype deep learning models.
- Built-in support for convolutional networks (for computer vision), recurrent networks (for sequence processing), and any combination of both.
- Supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, etc. This means that Keras is appropriate for building essentially any deep learning model, from a memory network to a neural Turing machine.

PANDAS

Pandas are a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. Pandas are a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

NUMPY

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

MATPLOTLIB

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

.

5.2. BACK END

MySQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals. MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries.

MySQL DATABASE.

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Esso Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

WAMPSERVER

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.

WAMPServer is a reliable web development software program that lets you create web apps with MYSQL database and PHP Apache2. With an intuitive interface, the application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn't require a payment or subscription.

BOOTSTRAP

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, and a wiki or go as big as a web-based calendar application or a commercial website.

CHAPTER 6

PROJECT DESCRIPTION

6.1. SYSTEM ARCHITECTURE

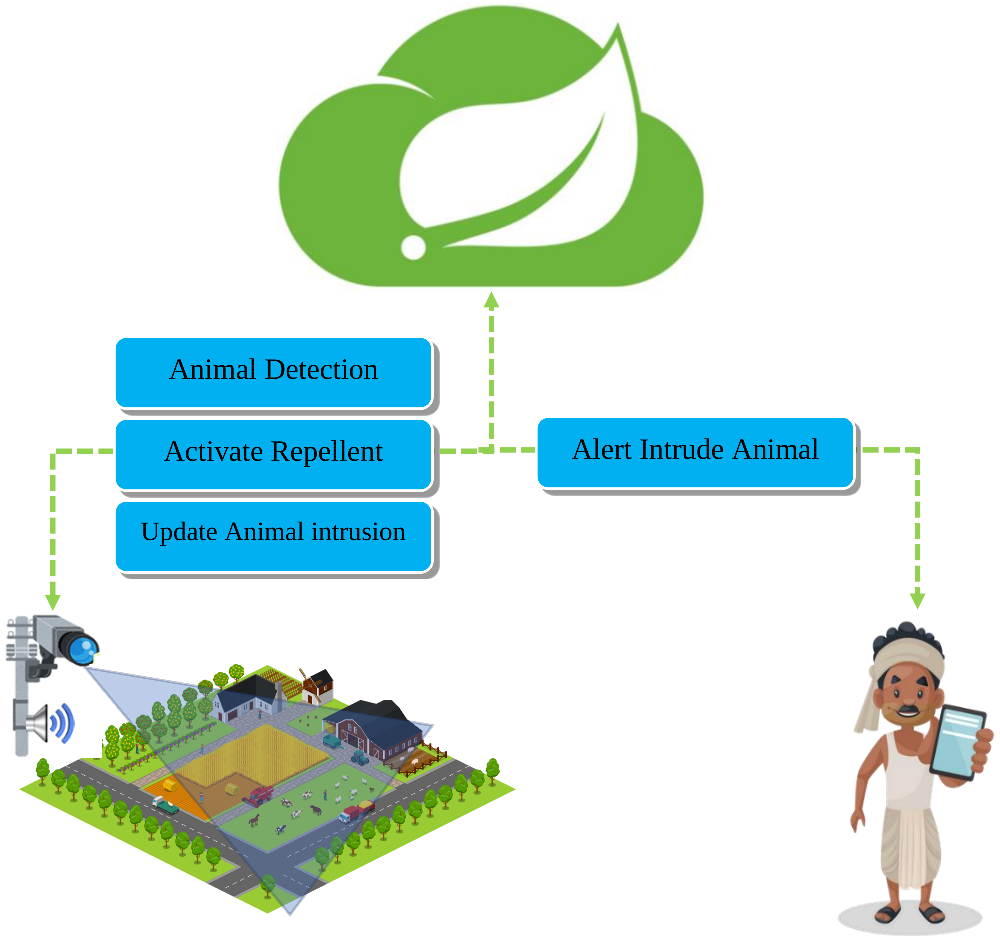


Figure no: 6.1. System Architecture

6.2. DCNN Model

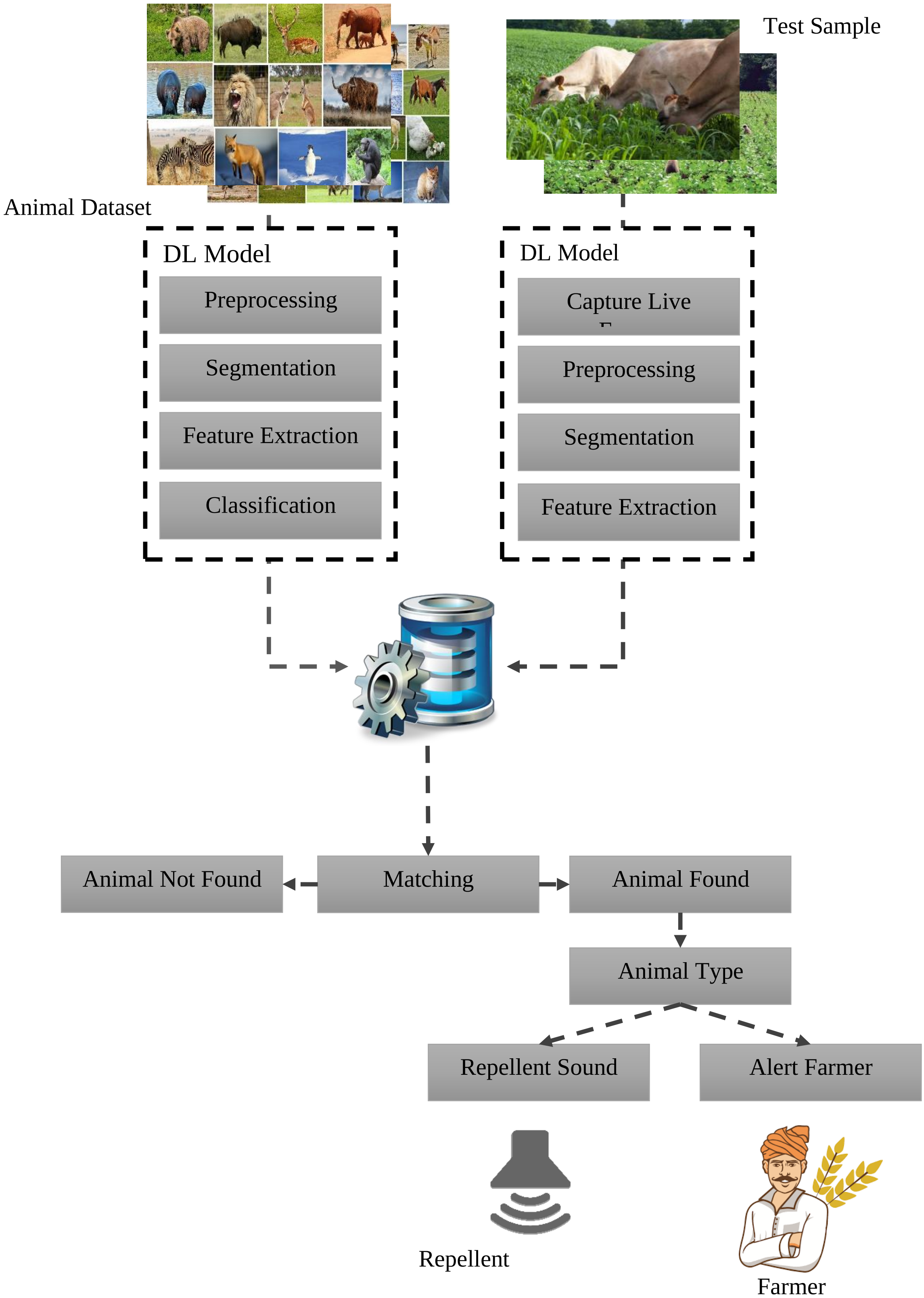


Figure no: 6.2. DCNN Model

6.3. DFD

Level 0

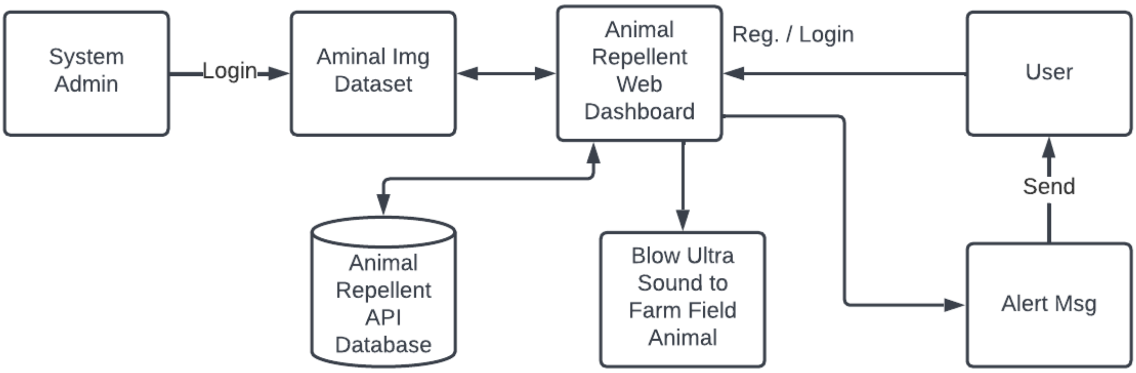


Figure no:6.3. Level 0 DFD

Level 1

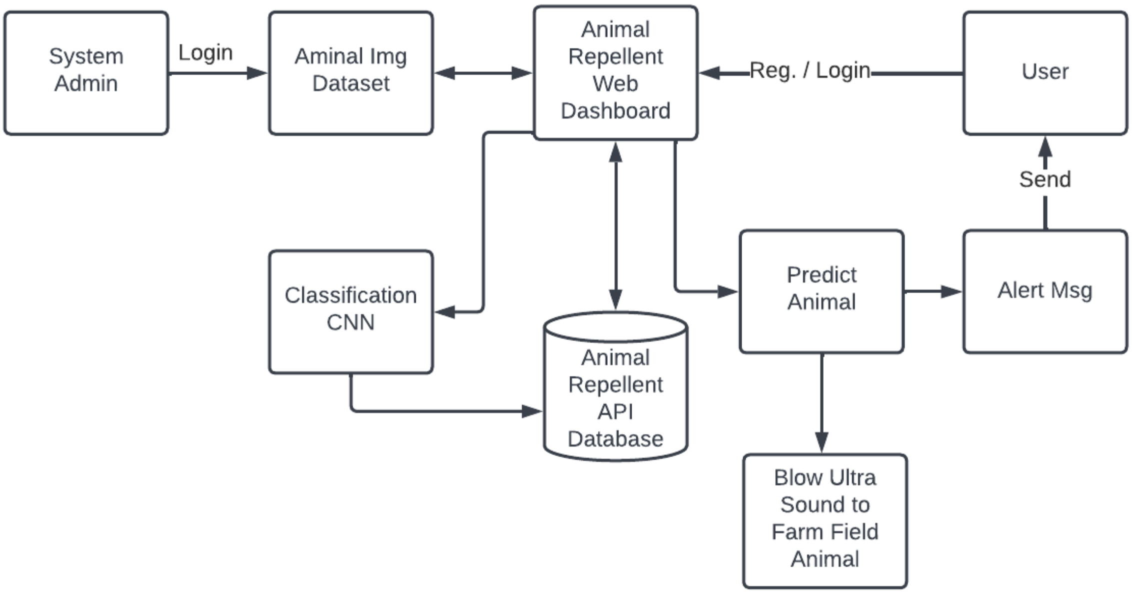


Figure no:6.4. Level 1 DFD

Level 2

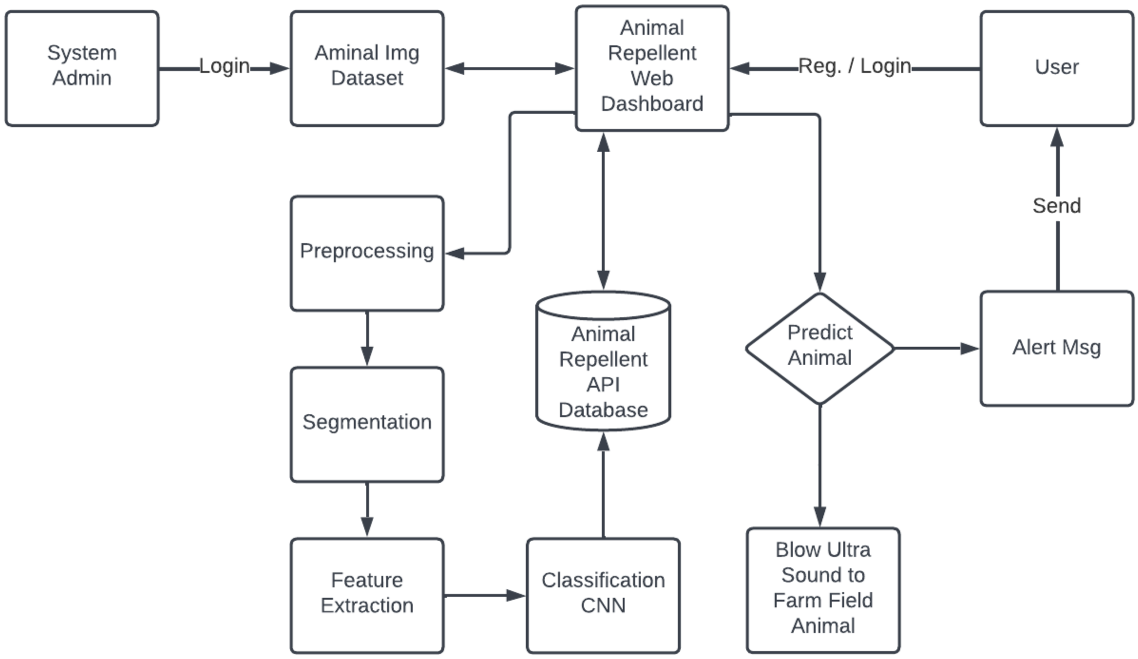


Figure no:6.5. Level 2 DFD

6.4. UML DIAGRAMS

Class Diagram

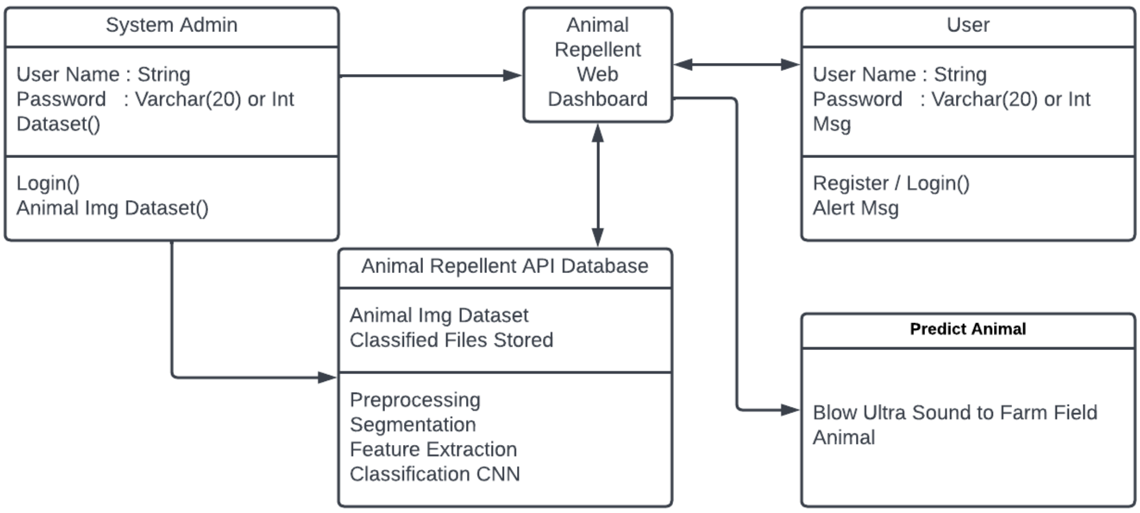


Figure no:6.6. Class Diagram

6.5 USECASE DIAGRAM

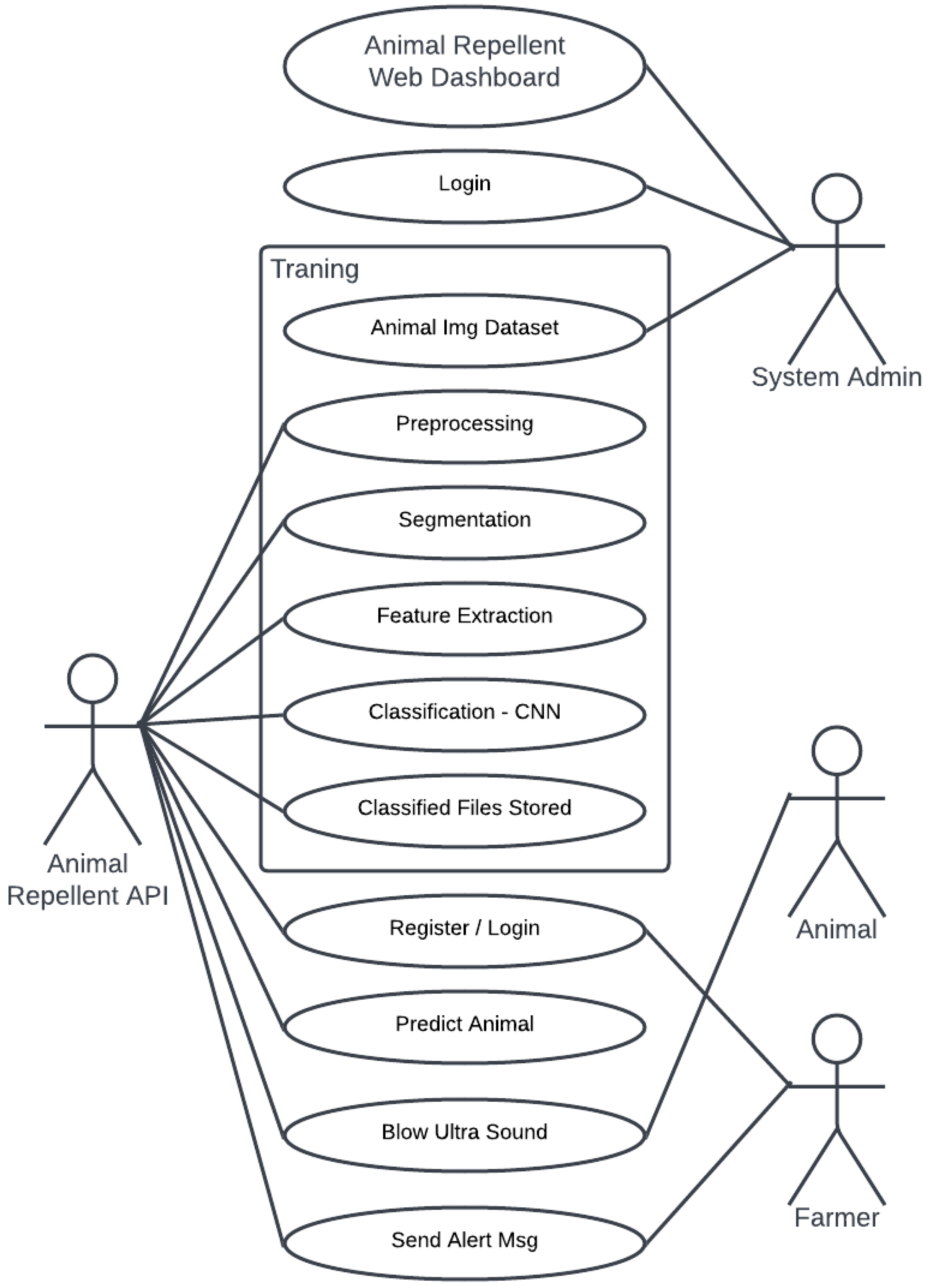


Figure no:6.7. Usecase Diagram

6.6. ACTIVITY DIAGRAM

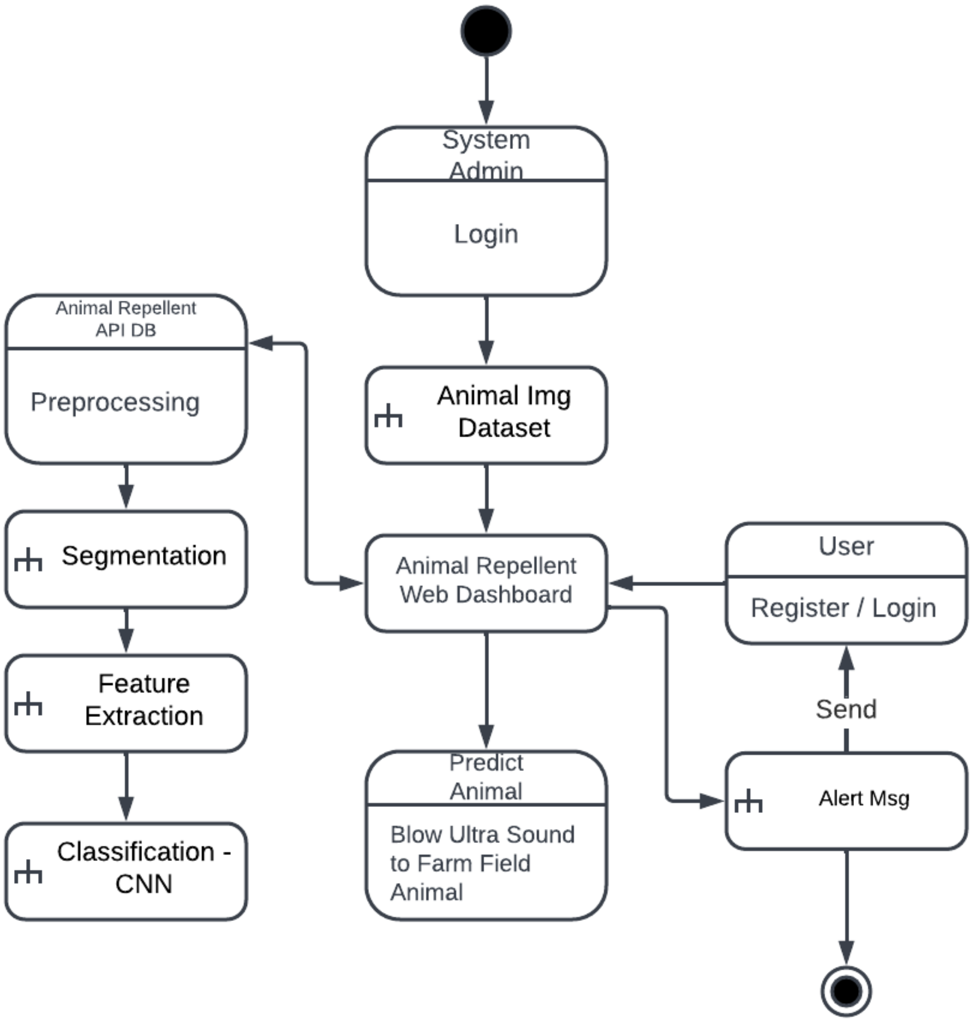


Figure no:6.8. Activity Diagram

6.7. SEQUENCE DIAGRAM

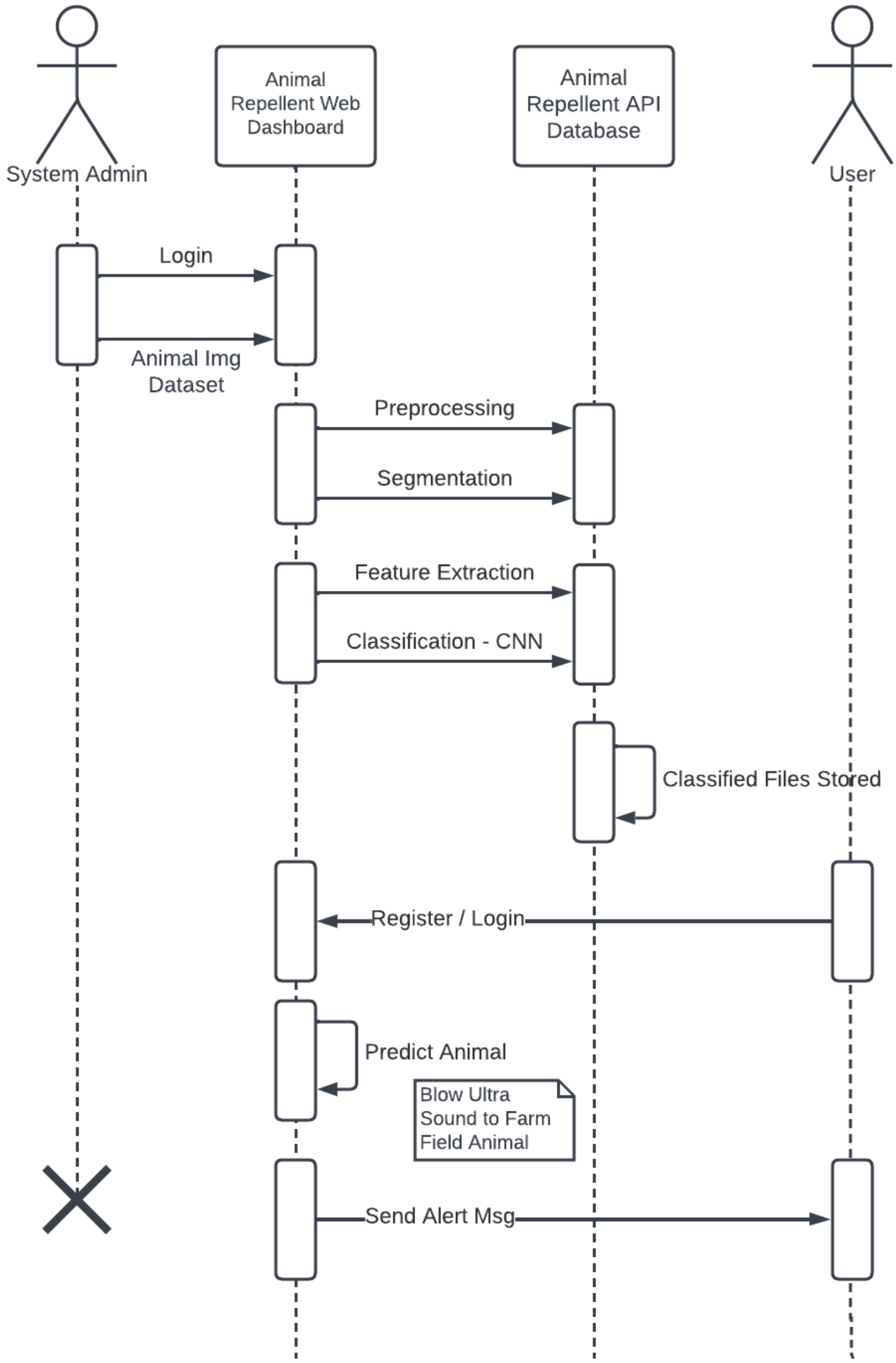


Figure no:6.9. Sequence Diagram

6.8. DEPLOYMENT DIAGRAM

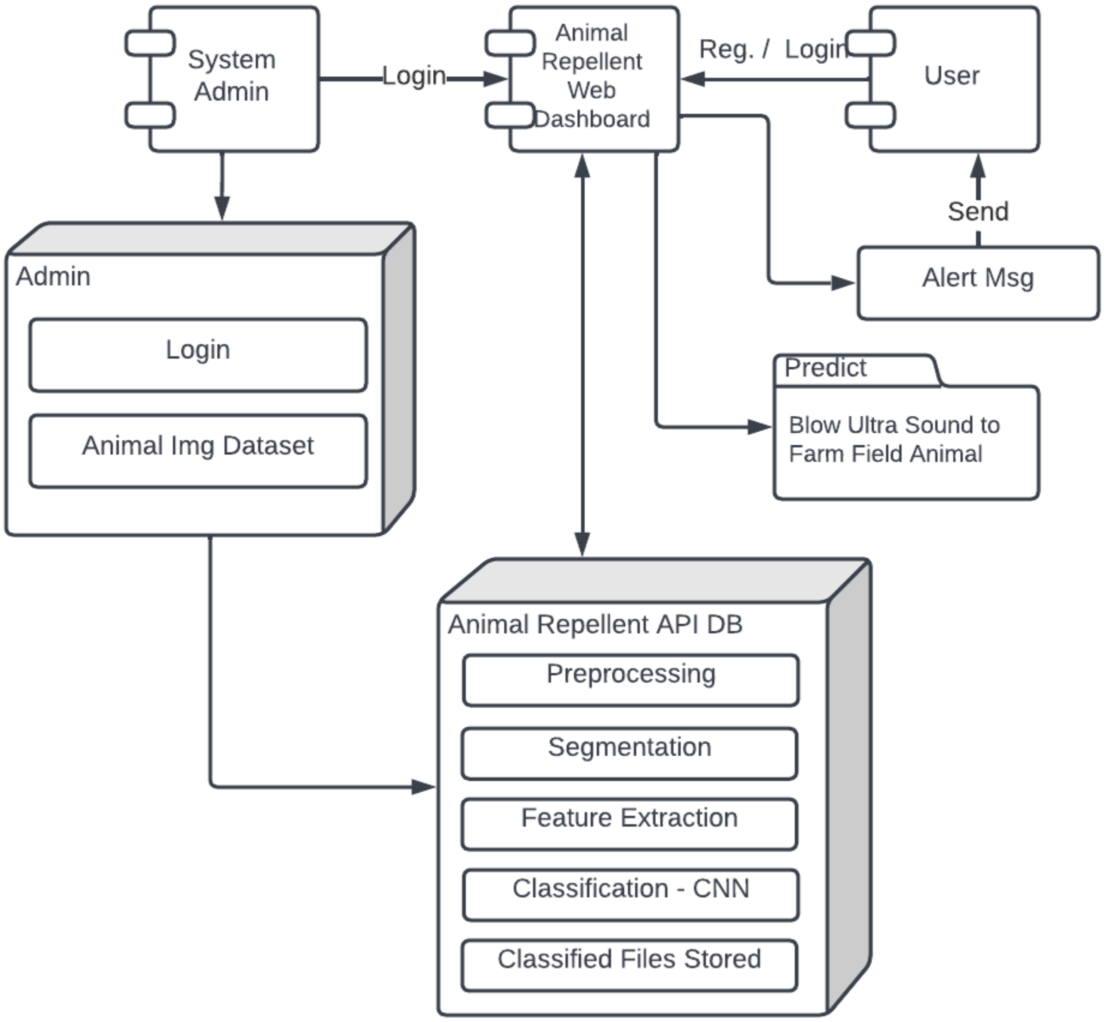


Figure no:6.10. Deployment Diagram

6.9. PROBLEM DESCRIPTION

In this project, deep convolution neural network-based classification algorithm is devised to detect animals both in video and images. Proposed approach is a classification model based on different features and classifiers. The different features like colour, Gabor and LBP are extracted from the segmented animal images. Possibilities of fusing the features for improving the performance of the classification have also been explored. Classification of animals is accomplished using CNN and symbolic classifiers. Initially, features are extracted from images/frames using blink app pre-trained convolution neural network. Later the extracted features are fed into multi-class CNN classifier for the purpose of classification. CNN is constructed using sequence of layers like Convolutional, subsampling and fully connected Layer.

Overall procedure for animal detection is given below:

1. The image is fetched using the monitoring panel.
2. The fetched image is processed using the python coding.
3. The fetched image is checked for various features of objects that match with any animal of trained data set.
4. Then it detects and classifies the animal which has been captured by the monitoring panel.
5. Algorithm calculates the accuracy in percentage based on number of matched objects.
6. If the accuracy of detected animal is above 45% the alert signal will be sent to the registered user through the SMS Service Provider.

CHAPTER 7

PROJECT DESCRIPTION

7.1. MODULES LIST

1. Animal Repellent Web Dashboard
2. Animal Recognition
3. Repellent
4. Monitoring and Visualizing
5. Notification
6. Performance Analysis

7.1.1. MODULES DESCRIPTION

1.Animal Repellent Web Dashboard

This system works in real time to detect the animals in the fields. The system enables the farmer to have a real time view of his fields from any place via internet and even provides manual buzzer controls if the need arises to use them. Thus, the farmer is in effective control of the system and can manually sound the buzzer if needed. This system is economical as compared to many of the existing solutions like electric fences, brick walls and manual supervision of the fields. This system is very effective in driving off the animals from the fields and keeping them away. It accurately determines the presence of animals in the fields and sounds the buzzer. It does not sound the buzzer due to the presence of a human being or due to some random motion. The ultrasonic buzzer is very effective against animals and causes no noise pollution. This system is totally harmless and doesn't injure animals in any way. It also doesn't cause any harm to humans. Also, this system has a very low power requirement thus reducing the hazards of electric shocks.

2.Animal Recognition

Training Phase

This module begins by annotation of animal dataset. These templates then become the reference for evaluating and registering the templates for the other poses: tilting up/down, moving closer/further, and turning left/right.

Animal Image Acquisition

ANIMAL-10N dataset contains 5 pairs of confusing animals with a total of 55,000 images. The 5 pairs are as following: (cat, lynx), (jaguar, cheetah), (wolf, coyote), (chimpanzee, orangutan), (hamster, guinea pig). The images are crawled from several online search engines including Bing and Google using the predefined labels as the search keyword.

Pre-processing

Animal Image pre-processing are the steps taken to format images before they are used by model training and inference. The steps to be taken are:

- Read image
- RGB to Grey Scale conversion
- Resize image

Original size (360, 480, 3) — (width, height, no. RGB channels)

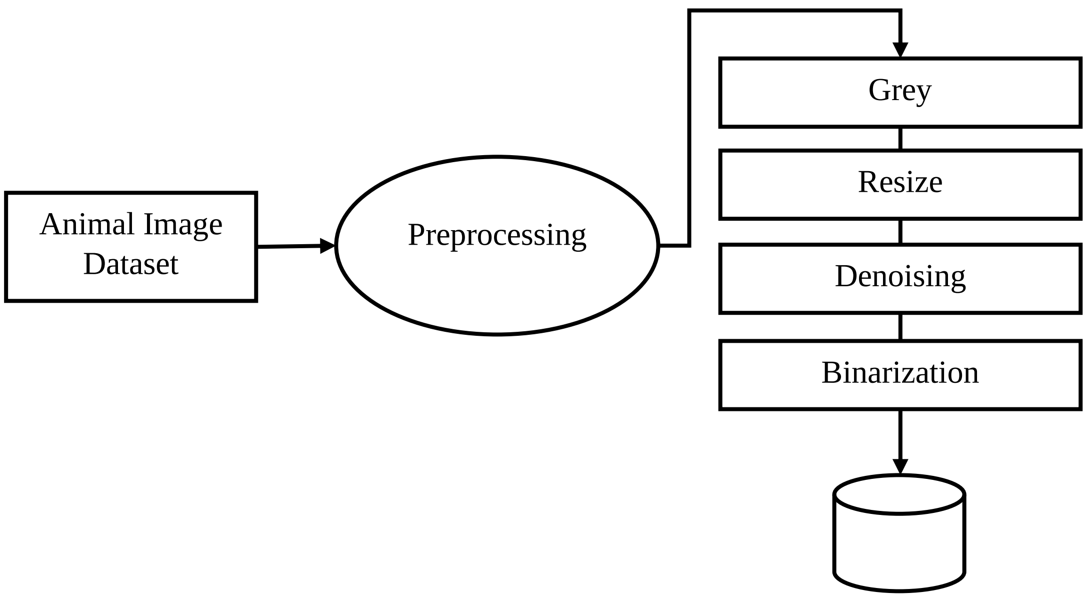
Resized (220, 220, 3)

- Remove noise (Denise)

Smooth our image to remove unwanted noise. We do this using Gaussian blur.

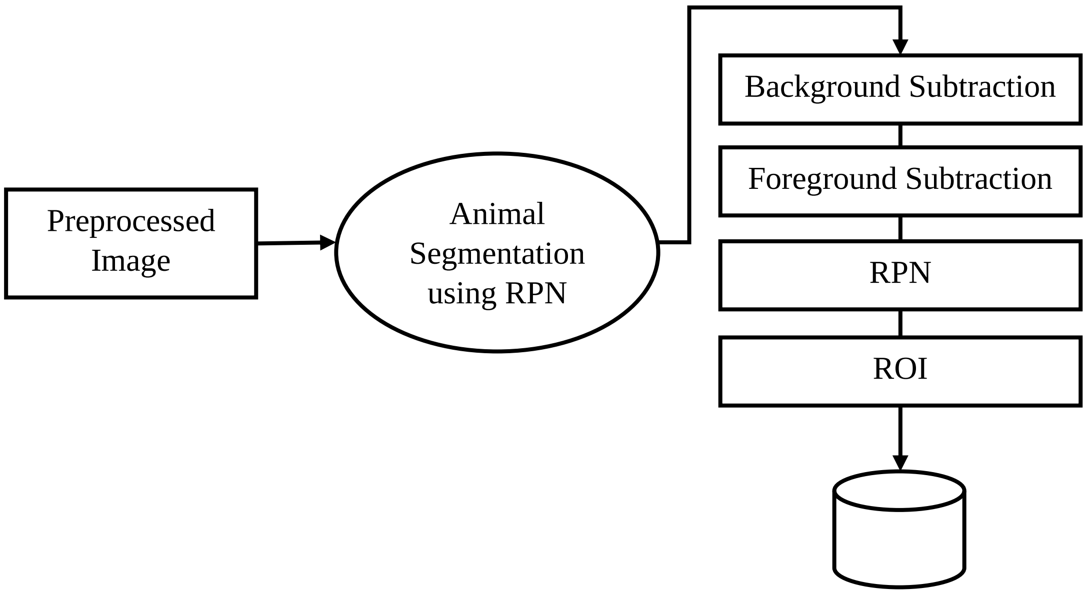
- Binarization

Image binarization is the process of taking a grayscale image and converting it to black-and-white, essentially reducing the information contained within the image from 256 shades of grey to 2: black and white, a binary image.



Animal Detection

Therefore, in this module, Region Proposal Network (RPN) generates RoIs by sliding windows on the feature map through anchors with different scales and different aspect ratios. Animal detection and segmentation method based on improved RPN. RPN is used to generate RoIs, and RoI Align faithfully preserves the exact spatial locations. These are responsible for providing a predefined set of bounding boxes of different sizes and ratios that are going to be used for reference when first predicting object locations for the RPN.



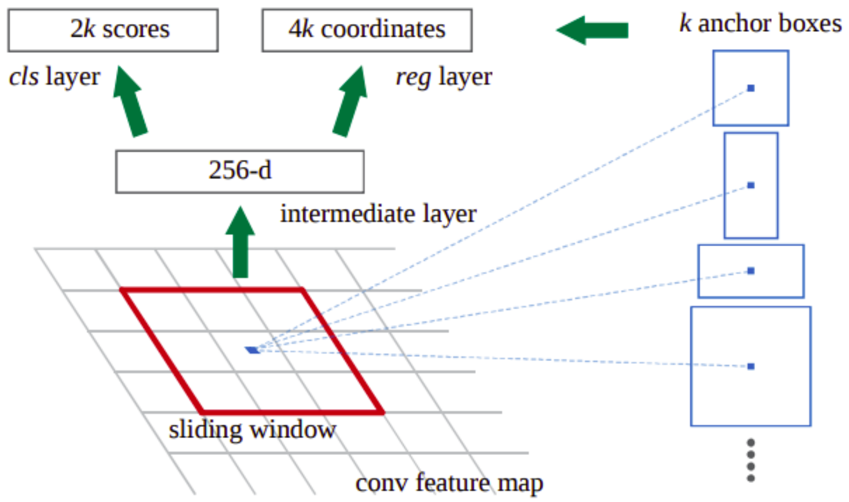
Animal Image segmentation using region growing (RG) method

The region growing methodology and recent related work of region growing are described here.

RG is a simple image segmentation method based on the seeds of region. It is also classified as a pixel-based image segmentation method since it involves the selection of initial seed points. This approach to segmentation examines the neighbouring pixels of initial “seed points” and determines whether the pixel neighbours should be added to the region or not based on certain conditions. In a normal region growing technique, the neighbour pixels are examined by using only the “intensity” constraint. A threshold level for intensity value is set and those neighbour pixels that satisfy this threshold is selected for the region growing.

RPN

A **Region Proposal Network**, or **RPN**, is a fully convolutional network that simultaneously predicts object bounds and objectless scores at each position. The RPN is trained end-to-end to generate high-quality region proposals. It works on the feature map (output of CNN), and each feature (point) of this map is called Anchor Point. For each anchor point, we place 9 anchor boxes (the combinations of different sizes and ratios) over the image. These anchor boxes are cantered at the point in the image which is corresponding to the anchor point of the feature map.



Training of RPN.

To know that for each location of the feature map we have 9 anchor boxes, so the total number is very big, but not all of them are relevant. If an anchor box having an object or part of the object within it then can refer it as a **foreground**, and if the anchor box doesn't have an object within it then we can refer it as **background**.

So, for training, assign a label to each anchor box, based on its Intersection over Union (IoT) with given ground truth. We basically assign either of the three (1, -1, 0) labels to each anchor box.

Label = 1 (Foreground): An anchor can have label 1 in following conditions,

If the anchor has the highest IoT with ground truth.

If the IoT with ground truth is greater than 0.7. (IoT >0.7).

Label = -1 (Background): An anchor is assigned with -1 if IoT < 0.3.

Label = 0: If it doesn't fall under either of the above conditions, these types of anchors don't contribute to the training, they are ignored.

After assigning the labels, it creates the mini-batch of 256 randomly picked anchor boxes, all of these anchor boxes are picked from the same image.

The ratio of the number of positive and negative anchor boxes should be 1:1 in the mini-batch, but if there are less than 128 positive anchor boxes then we pad the mini-batch with negative anchor boxes.

Now the RPN can be trained end-to-end by backpropagation and stochastic gradient descent (SGD).

The processing steps are

- Select the initial seed point
- Append the neighbouring pixels—intensity threshold
- Check threshold of the neighbouring pixel
- Thresholds satisfy-selected for growing the region.
- Process is iterated to end of all regions.

Feature Extraction

In feature extraction process, the useful information or characteristics of the image are extracted in the form of statistical, shape, colour and texture features. The Transformation of the input image into features is called feature extraction. Features are extracted by using feature extraction techniques. Features are extracted based on texture, boundary, spatial, edge, transform, colour and shape features. Shape-based features are divided into the boundary and region-based features. Boundary features are also called contour-based which uses boundary segments. Boundary based features are geometrical descriptors (diameter, major axis, minor axis, perimeter, eccentricity and curvature), Fourier descriptors and statistical descriptors (mean, variance, standard deviation, skew, energy and entropy). Region based features are texture features as GLCM.

Gary Level Co-occurrence Matrix

GLCM is a second-order statistical texture analysis method. It examines the spatial relationship among pixels and defines how frequently a combination of pixels are present in an image in a given direction Θ and distance d. Each image is quantized into 16 gravy levels (0–15) and 4 GLCMs (M) each for Θ = 0, 45, 90, and 135 degrees with d = 1 are obtained. From each GLCM, five features (Eq. 13.30–13.34) are extracted. Thus, there are 20 features for each image. Each feature is normalized to range between 0 to 1 before passing to the classifiers, and each classifier receives the same set of features.

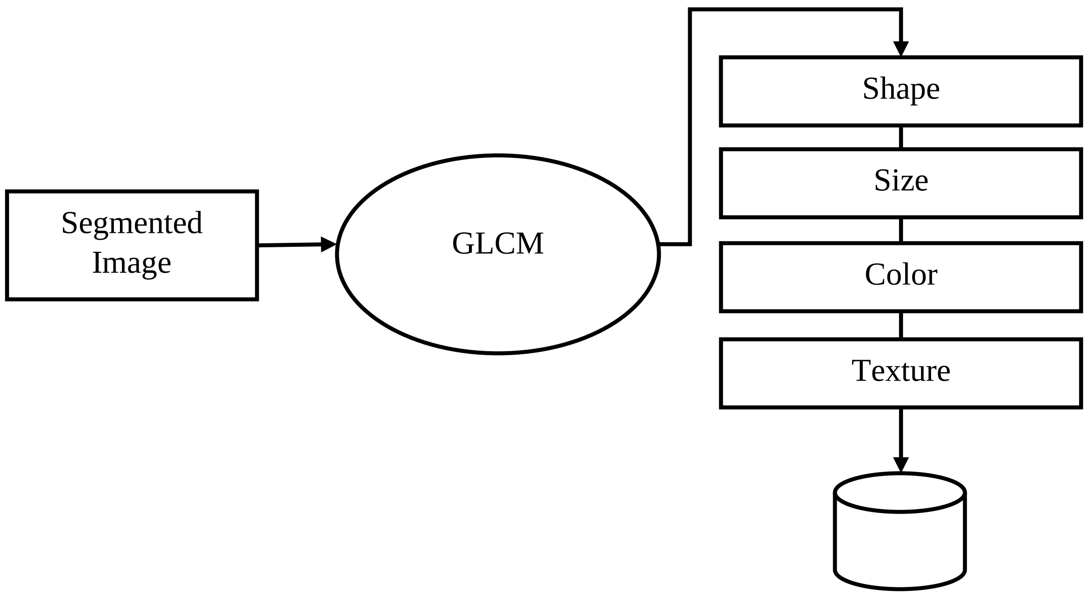
The features we extracted can be grouped into three categories. The first category is the first order statistics, which includes maximum intensity, minimum intensity, mean, median, 10th percentile, 90th percentile, standard deviation, variance of intensity value, energy, entropy, and others. These features characterize the Gray level intensity of the tumour region.

SI.No	**GLCMFeature**	**Formula**
1.	*Contrast*	$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2$
2.	*Correlation*	$\sum_{i,j=0}^{N-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}\right]$
3.	*Dissimilarity*	$\sum_{i,j=0}^{N-1} P_{i,j}\lvert i-j\rvert$
4.	*Energy*	$\sum_{i,j=0}^{N-1} P_{i,j}^2$
5.	*Entropy*	$\sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j})$
6.	*Homogeneity*	$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$
7.	*Mean*	$\mu_i = \sum_{i,j=0}^{N-1} i(P_{i,j})$, $\mu_j = \sum_{i,j=0}^{N-1} j(P_{i,j})$
8.	*Variance*	$\sigma_i^2 = \sum_{i,j=0}^{N-1} P_{i,j}(i-\mu_i)^2$, $\sigma_j^2 = \sum_{i,j=0}^{N-1} P_{i,j}(j-\mu_j)^2$
9.	*Standard Deviation*	$\sigma_i = \sqrt{\sigma_i^2}$, $\sigma_j = \sqrt{\sigma_j^2}$

Table No:7.1 Formulas to Calculate Texture Features From GLCM

The second category is shape features, which include volume, surface area, surface area to volume ratio, maximum 3D diameter, maximum 2D diameter for axial, coronal and sagittal plane respectively, major axis length, minor axis length and least axis length, sphericity, elongation, and other features. These features characterize the shape of the tumour region.

The third category is texture features, which include 22 Gray level co-occurrence matrix (GLCM) features, 16 Gray level run length matrix (GLRLM) features, 16 Gray level size zone matrix (GLSZM) features, five neighbouring Gray tone difference matrix (NGTDM) features and 14 Gray level dependence matrix (GLDM) Features. These features characterize the texture of the tumour region.



DCNN algorithms were implemented to automatically detect and reject improper animal images during the classification process. This will ensure proper training and therefore the best possible performance.

The CNN creates feature maps by summing up the convolved grid of a vector-valued input to the kernel with a bank of filters to a given layer. Then a non-linear rectified linear unit (ReLU) is used for computing the activations of the convolved feature maps. The new feature map obtained from the ReLU is normalized using local response normalization (LRN). The output from the normalization is further computed with the use of a spatial pooling strategy (maximum or average pooling). Then, the use of dropout regularization scheme is used to initialize some unused weights to zero and this activity most often takes place within the fully connected layers before the classification layer. Finally, the use of softmax activation function is used for classifying image labels within the fully connected layer.

3.Animal Identification

After capturing the animal image from the Farm Camera, the image is given to animal detection module. This module detects the image regions which are likely to be human. After the animal detection using Region Proposal Network (RPN), animal image is given as input to the feature extraction module to find the key features that will be used for classification. The module composes a very short feature vector that is well enough to represent the animal image.

Repellent

Monitoring window detecting the presence of animals then it enables repelled module to repelling them through the generation of ultrasounds, which has recently been proven as an alternative, effective method for protecting crops against predicted animals. Animals generally have a sound sensitive threshold that is far higher than humans. They can hear sounds having lower frequencies with respect to the human ear. For instance, while the audible range for humans is from 64Hz - 23 KHz, the corresponding range of goats, sheep, domestic pigs, dogs and cats is 78Hz - 37 KHz, 10 Hz - 30 KHz, 42 Hz - 40.5 KHz 67 Hz - 45 KHz and 45 Hz - 64 KHz respectively.

Monitoring and Visualizing

The system works in real time detect the animals in the field; in addition the farmers can access the view of their fields remotely. Type of animal and also the count can be given. The animal recognition module will share the data over the cloud regularly through a Wi-Fi connection. The cloud setup will consist of a private cloud instance running on a machine. The data shared will be used to analyse the patterns and responses of wild animals. The farmer can visualize the errors if any, resolve them, and achieve better results.

Notification

The email and sms notification consisting of captured image is notified to the user regarding the detected motion in this phase. The email is sent to registered email id and sms is sent to the Telegram account of the user to the registered number.

Performance Analysis

In this module we able to find the performance of our system using SENSITIVITY, SPECIFICITY AND ACCURACY of Data in the datasets are divided into two classes not animal (the negative class) and animal and type (the positive class). Sensitivity, specificity, and accuracy are calculated using the True positive (TP), true negative (TN), false negative (FN), and false positive (FP). TP is the number of positive cases that are classified as positive. FP is the number of negative cases that are classified as positive. TN is the number of negative cases classified as negative and FN is the number of positive cases classified as negative.

The important points involved with the performance metrics are discussed based on the context of this project:

True Positive (TP): There is an Animal and the algorithms detect Animal name.

False Positive (FP): There is no Animal, but the algorithms detect as Animal and display Animal name.

False Negative (FN): There is an Animal, but the algorithms do not detect Animal and name.

True Negative (TN): There is no Animal, and nothing is being detected.

Accuracy

Accuracy is a measure that tells whether a model/algorithm is being trained correctly and how it performs. In the context of this thesis, accuracy tells how well it is performing in detecting humans in underwater environment. Accuracy is calculated using the following formula.

Accuracy = (T P + T N)/ (T P + T N + F P + F N)

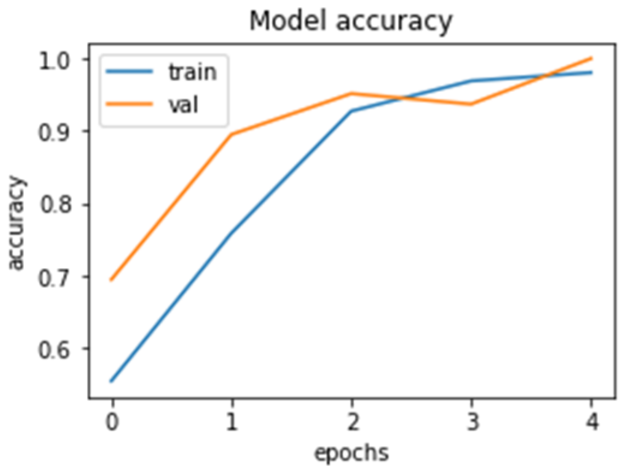


Figure no:7.1. Model Accuracy

Precision

It denotes the ratio of positively predicted cases that are actually positive. In the context of this thesis, precision measures the fraction of objects that are predicted to be animals and are actually animals present in farm environment. Precision is calculated using the following formula.

Precision = T P/ (T P + F P)

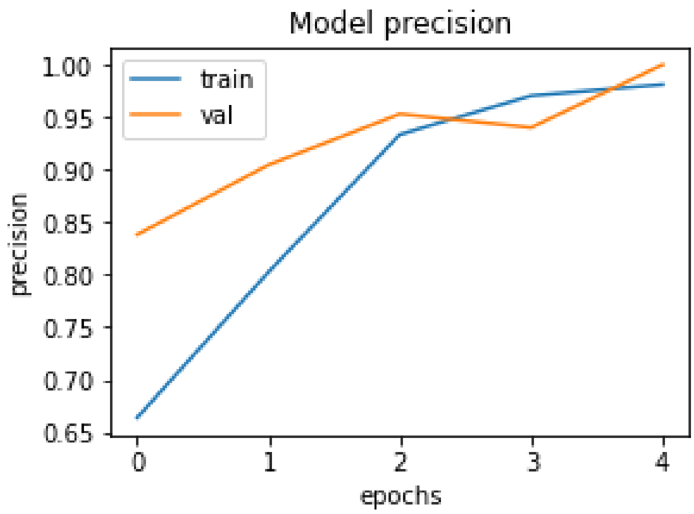


Figure no: 7.2 Model Precision

Recall

It is the ratio between actual positive cases that are predicted to be positive. In the context of this thesis, recall measures the fraction of animals that are predicted as animals. Recall is calculated using the following formula.

Recall =T P/ (T P + F N)

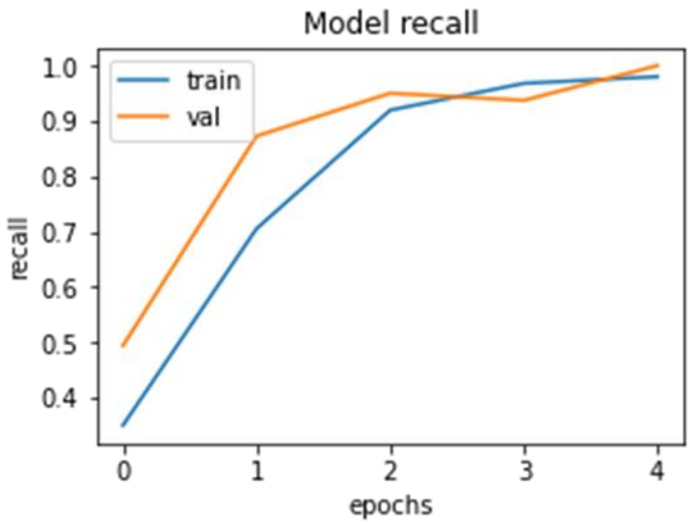


Figure no:7.3. Model Recall

- **F1 Score**

It is also known as balanced F-score or F-measure. F1 score is a measure of accuracy of a model combining precision and recall. In the context of this thesis, a good F1 score shows that there are less false positives and false negatives. This shows that the model is correctly identifying animals in farm environment.

A model/algorithm is considered perfect if F1 score is 1. It is calculated using the following formula.

F1 = 2 × (Precision × Recall /Precision + Recall)

Confusion Matrix – Training

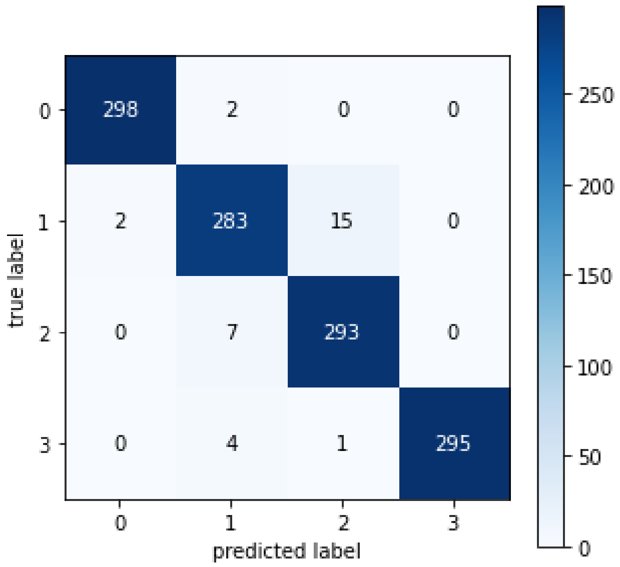


Figure no:7.4. Confusion Matrix – Training

Confusion Matrix – Testing

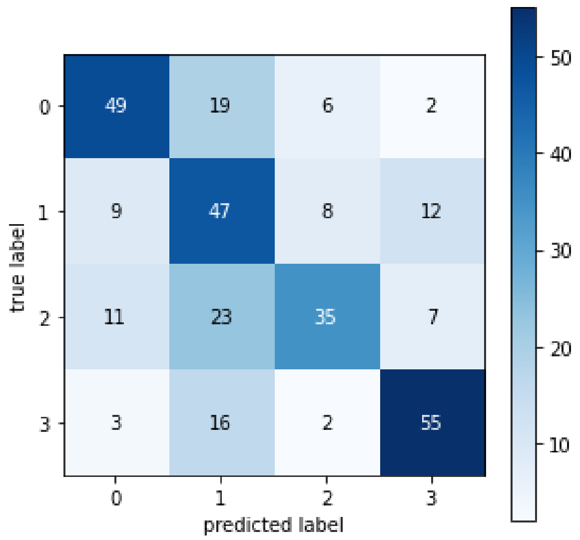


Figure no:7.5. Confusion Matrix – Testing

Accuracy: 0.9984025559105432

Precision: 0.9990234375

Recall: 0.9964285714285714

F1_score: 0.9977122020583142

Training time

Training time is metric used in this thesis to measure the time taken to train the selected machine learning algorithms on the dataset.

Prediction Speed

Speed is a metric used in this thesis to measure the time taken for the algorithms to process and detect obstacle.

Loss Function

Loss function, to perform feature matching between the ground truth and the output of segmentation network, optimizing also the network weights on features extracted at multiple resolutions rather than focusing just on the pixel level.

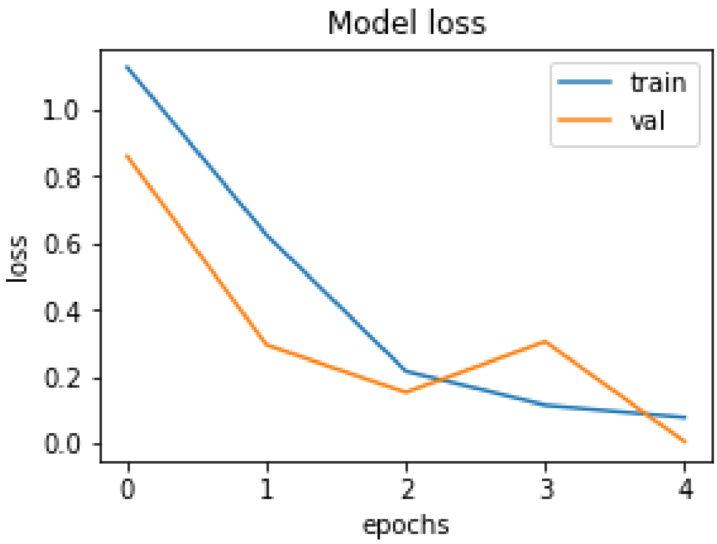


Figure no:7.6. Loss Function

CHAPTER 8

TESTING

8.1. SYSTEM TESTING

In this phase of methodology, testing was carried out on the several application modules. Different kind of testing was done on the modules which are described in the following sections. Generally, tests were done against functional and non-functional requirements of the application following the test cases. Testing the application again and again helped it to become a reliable and stable system.

8.2. UNIT TESTING

Before you can test an entire software program, make sure the individual parts work properly on their own. Unit testing validates the function of a unit, ensuring that the inputs (one to a few) result in the lone desired output. This testing type provides the foundation for more complex integrated software. When done right, unit testing drives higher quality application code and speeds up the development process. Developers often execute unit tests through test automation.

CHAPTER 9

SYSTEM IMPLEMENTATION

9.1. RESULT AND DISCUSSION

This module graphs the training and validation accuracy and loss for each epoch. During an epoch, the loss function is calculated across every data item and it is guaranteed to give the quantitative loss measure at the given epoch and plotting curve across each iteration only gives the loss on a subset of the entire dataset.

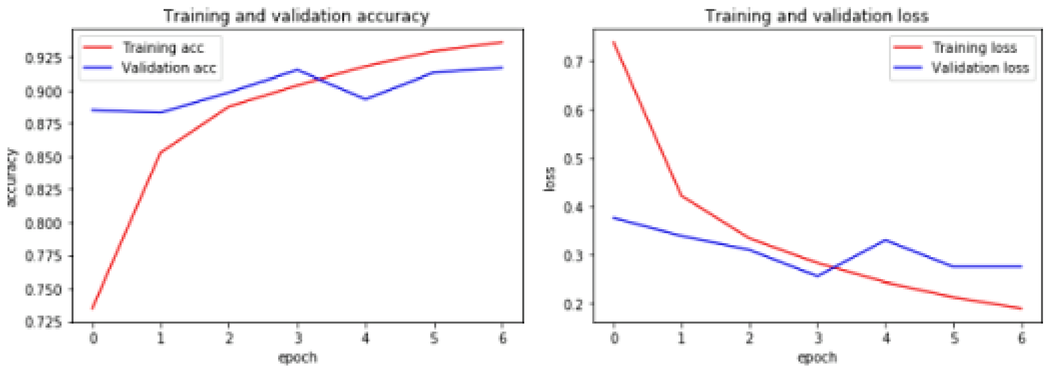


Figure no:9.1. Training and Validation accuracy and loss graph

Experiments with the dataset E2 show better accuracy results respect to the unbalanced dataset E1 reaching inaccuracy of 80.6% Top-1 and 94.1% Top-5, respectively (Fig. 7.2a). The plots of training and testing accuracy of the joint CNN (Top-5) depending of number of epochs are depicted in Fig.7.2b.

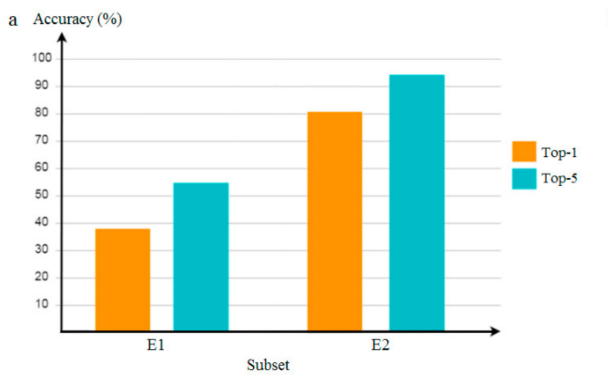


Figure no:9.2. Accuracy of the joint CNN (Top-1 and Top-5) during training

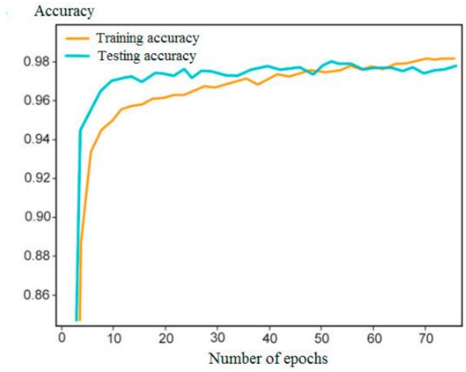


Figure no:9.3. Training and testing accuracy of the joint CNN (Top-5).

First experiment was conducted using a single branch SVM without taking in consideration the features of muzzles and shapes. Second experiment employed the proposed joint CNN according to decision marking rules. The obtained results in the terms of Average Precision (AP), Miss Rate (MR), and False Positives (FP) are grouped in Tables 1 and 2 for the dataset E2.

Animals	AP, %	MR,%	FP,%	Animals	AP, %	MR,%	FP,%
Goat	78.83	12.2	11.3	Goat	76.52	8.4	16.3
Cow	80.64	9.4	14.8	Cow	81.3	7.8	12.5
Elephant	73.88	15.1	18.6	Elephant	75.54	14.2	17.9
Deer	76.74	13.7	17.4	Deer	69.88	19.6	22.4
Horse	80.91	8.9	16.1	Horse	63.85	24.2	26.9
Pig	79.72	8.1	18.2	Pig	79.21	16.3	18.7

Table No 9.1. Animal recognition results using a single branch SVM.

Animals	AP, %	MR,%	FP,%	Animals	AP, %	MR,%	FP,%
Goat	83.37	10.7	9.7	Goat	81.93	6.9	14.4
Cow	84.29	9.1	13.2	Cow	85.69	7.1	10.8
Elephant	79.13	12.4	16.3	Elephant	79.14	12.3	14.7
Deer	80.21	11.8	16.5	Deer	75.35	18.6	19.8
Horse	84.9	7.9	12.9	Horse	69.52	21.6	24.8
Pig	83.07	8	15.6	Pig	81.23	14.9	17.2

Table 9.2. Animal recognition results using the joint CNN.

In this project, we proposed classifiers for endangered animal species. The models extract the features of the animal appearances at the convolutional layer, which has been retrained on a large amount of data, and then we classify the images at the last layer. Our proposed models require a relatively short time to conduct their job. They are more accurate than the models trained from scratch, especially for the species that have a similar color. This is because the retrained models can already extract the low-level features of a new image. Another advantage of the models trained by transfer learning is that the model does not need to draw a bounding box to train the last layer. This approach will greatly reduce the inconvenience for humans by eliminating manual processes. We expect that the accuracy will be increased if fine tuning is applied. Finally, Tf.keras-based model can be easily deployed on an Android mobile device using the FlatBuffer file converter provided by TensorFlow Lite. To clarify the key points of this study, we suggest the following highlights:

(i) CNN models with transfer learning can be trained without any special difficulty

(ii) The designed advanced CNN models do not require any manual preprocessing (such as labeling or drawing bounding boxes on the images)

(iii) The CNN models can be easily converted into a file for deploying in a mobile application using TensorFlow Lite framework

(iv) The web application can classify endangered quasi-species of animals having a high color similarity in real time

CHAPTER 10

CONCLUSION & FUTURE ENHANCEMENT

10.1. CONCLUSION

Agricultural farm security is widely needed technology nowadays. In order to accomplish this, a vision-based system is proposed and implemented using Python and Open CV and developed an Animal Repellent System to blow out the animals. The implementation of the application required the design and development of a complex system for intelligent animal repulsion, which integrates newly developed software components and allow stop recognize the presence and species of animals in real-time and also to avoid crop damages caused by the animals. Based on the category of the animal detected, the edge computing device executes its DCNN Animal Recognition model to identify the target, and if an animal is detected, it sends back a message to the Animal Repelling Module including the type of ultrasound to be generated according to the category of the animal. The proposed CNN was evaluated on the created animal database. The overall performances were obtained using different number of training images and test images.

The obtained experimental results of the performed experiments show that the proposed CNN gives the best recognition rate for a greater number of input training images (accuracy of about 98 %). This project presented a real-time monitoring solution based on AI technology to address the problems of crop damages against animals. This technology used can help farmers and agronomists in their decision making and management process.

10.2. FUTURE ENHANCEMENT

Further in the proposed architecture, some image compression techniques can be developed to reduce the time taken for notification to reach user as described above.

CHAPTER 11

APPENDICES AND SCREEN SHOTS

APPENDIX-A

11.1. SOURCE CODE

```
# main.py
Import os
Import base64
from flask import Flask, render_template, Response, redirect, request, session, abort, url_for
from camera import VideoCamera
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC
from cryptography.fernet import Fernet
import argparse
import cv2
import shutil
import random
from random import seed
from random import randint
import time
import PIL.Image
from PIL import Image, ImageChops
import numpy as np
import pandas as pd
import random
import seaborn as sns
import matplotlib.pyplot as plt
```

```
import math
import imagehash
import mysql.connector
import urllib.request
import urllib.parse
from werkzeug.utils import secure_filename

from urllib.request import urlopen
import webbrowser

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="",
  charset="utf8",
  database="animal_repellent"

)

UPLOAD_FOLDER = 'static/trained'
ALLOWED_EXTENSIONS = { 'png', 'jpg', 'jpeg', 'gif'}

app = Flask(__name__)
app.secret_key = 'abcdef'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

#@app.route('/')
```

```
#def index():
#    return render_template('index.html')



@app.route('/', methods=['GET', 'POST'])
def index():
    msg=""

    ff3=open("ulog.txt","w")
    ff3.write("")
    ff3.close()

    act=request.args.get("act")
    act2=request.args.get("act2")
    act3=request.args.get("act3")

    return render_template('index.html',msg=msg,act=act,act2=act2,act3=act3)

@app.route('/upload', methods=['GET', 'POST'])
def upload():
            hash0 = imagehash.average_hash(Image.open("static/dataset/"+fname))
            hash1 = imagehash.average_hash(Image.open("static/test/m1.jpg"))
            cc1=hash0 - hash1
            print("cc="+str(cc1))
            if cc1<=cutoff:
                fn=fname
                ss="ok"
                break
        if ss=="ok":
            act3="yes"
```

```
            else:
                act3="no"



    if gf1[0]==fn:
                gid=int(gf1[1])-1
                fn2="c_"+fn
                animal=cname[gid]
                afile="a"+gf1[1]+".mp3"
                break
        print(fn2)
        print(animal)
        print(afile)

        ff3=open("ulog.txt","r")
        user=ff3.read()
        ff3.close()

        ff4=open("sms.txt","r")
        sms=ff4.read()
        ff4.close()

        if user=="":
            aa=1
        else:

            if sms=="":
                aa=1
            else:
                mycursor.execute("SELECT * FROM farmer where uname=%s",(user, ))
                row1 = mycursor.fetchone()
```

```
            mobile=row1[2]
            name=row1[1]

            mess=animal+" detected"

url="http://iotcloud.co.in/testsms/sms.php?sms=msg&name="+name+"&mess="+mess+"&mobile="
+str(mobile)
            webbrowser.open_new(url)

            ff41=open("sms.txt","w")
            ff41.write("")
            ff41.close()

        mycursor = mydb.cursor()
        mycursor.execute("SELECT max(id)+1 FROM animal_detect")
        maxid = mycursor.fetchone()[0]
        if maxid is None:
            maxid=1
        sql = "INSERT INTO animal_detect(id,user,animal,image_name) VALUES (%s, %s,%s,%s)"
        val = (maxid,user,animal,fn2)
        mycursor.execute(sql, val)
        mydb.commit()

    elif act3=="no":
        g=2
        msg="No Result"
    return
render_template('upload.html',msg=msg,act=act,act2=act2,act3=act3,fn=fn,animal=animal,fn2=fn2,
afile=afile,page=page)

@app.route('/login', methods=['GET', 'POST'])
def login():
```

```
    msg=""

    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM admin WHERE username = %s AND password = %s', (uname,
pwd))
        account = cursor.fetchone()
        if account:
            session['username'] = uname
            return redirect(url_for('train_data'))
        else:
            # Account doesnt exist or username/password incorrect
            msg = 'Incorrect username/password!'



    return render_template('login.html',msg=msg)


@app.route('/login_farmer', methods=['GET', 'POST'])
def login_farmer():
    msg=""
    msg1=""
    act = request.args.get('act')
    if act=="success":
        msg1="New Farmer Register Success"
    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
```

```
        cursor.execute('SELECT * FROM farmer WHERE uname = %s AND pass = %s', (uname, pwd))
        account = cursor.fetchone()
        if account:
            session['username'] = uname

            ff3=open("ulog.txt","w")
            ff3.write(uname)
            ff3.close()

            ff3=open("sms.txt","w")
            ff3.write("yes")
            ff3.close()

            return redirect(url_for('userhome'))
        else:
            # Account doesnt exist or username/password incorrect
            msg = 'Incorrect username/password!'



    return render_template('login_farmer.html',msg=msg,msg1=msg1)


@app.route('/userhome', methods=['GET', 'POST'])
def userhome():
    msg=""

    msg=""
    act=request.args.get("act")
    act2=request.args.get("act2")
    act3=request.args.get("act3")
```

```
    return render_template('userhome.html',msg=msg,act=act,act2=act2,act3=act3)


@app.route('/register', methods=['GET', 'POST'])
def register():
    msg=""


    if request.method=='POST':
        name=request.form['name']
        mobile=request.form['mobile']
        email=request.form['email']
        location=request.form['location']
        uname=request.form['uname']
        pwd=request.form['pass']



        mycursor = mydb.cursor()
        mycursor.execute("SELECT max(id)+1 FROM farmer")
        maxid = mycursor.fetchone()[0]
        if maxid is None:
            maxid=1
        sql = "INSERT INTO farmer(id,name,mobile,email,location,uname,pass) VALUES (%s, %s, %s, %s, %s, %s, %s)"
        val = (maxid,name,mobile,email,location,uname,pwd)
        mycursor.execute(sql, val)
        mydb.commit()
        print(mycursor.rowcount, "Added Success")
        act='success'
        return redirect(url_for('login_farmer',act=act))
except:
        print("excep")
```

```
    msg1=""
    msg2=""
    mess=""
    ff=open("get_value.txt","r")
    get_value=ff.read()
    ff.close()
    s=""
    if get_value=="":
        s="1"
    else:

        msg1=get_value+" detected, "

    ff1=open("person.txt","r")
    pp=ff1.read()
    ff1.close()
    sc=""
    if pp=="":
        sc="1"
    else:
        msg2=""+pp+""

    mess=msg1+" "+msg2
    '''
    return render_template('process.html',mess=mess,act=act)

def object_detect(fname):
    # construct the argument parse
    parser = argparse.ArgumentParser(
```

```
    description='Script to run MobileNet-SSD object detection network ')
  parser.add_argument("--video", help="path to video file. If empty, camera's stream will be used")
  parser.add_argument("--prototxt", default="MobileNetSSD_deploy.prototxt",
                    help='Path to text network file: '
                      'MobileNetSSD_deploy.prototxt for Caffe model or '
                      )
  parser.add_argument("--weights", default="MobileNetSSD_deploy.caffemodel",
                   help='Path to weights: '
                     'MobileNetSSD_deploy.caffemodel for Caffe model or '
                     )
  parser.add_argument("--thr", default=0.2, type=float, help="confidence threshold to filter out
weak detections")
  args = parser.parse_args()

  # Labels of Network.
  classNames = { 0: 'background',
        1: 'Bear', 2: 'Cow', 3: 'Elephant', 4: 'Goat',
        5: 'Horse', 6: 'Pig', 7: 'Sheep' }

  # Open video file or capture device.
  '''if args.video:
    cap = cv2.VideoCapture(args.video)
  else:
    cap = cv2.VideoCapture(0)'''

  #Load the Caffe model
  net = cv2.dnn.readNetFromCaffe(args.prototxt, args.weights)

  #while True:
  # Capture frame-by-frame
  #ret, frame = cap.read()
```

```
    frame = cv2.imread("static/test/"+fname)
    frame_resized = cv2.resize(frame,(300,300)) # resize frame for prediction

@app.route('/login_farmer', methods=['GET', 'POST'])
def login_farmer():
    msg=""
    msg1=""
    act = request.args.get('act')
    if act=="success":
        msg1="New Farmer Register Success"
    if request.method=='POST':
        uname=request.form['uname']
        pwd=request.form['pass']
        cursor = mydb.cursor()
        cursor.execute('SELECT * FROM farmer WHERE uname = %s AND pass = %s', (uname, pwd))
        account = cursor.fetchone()
        if account:
            session['username'] = uname

            ff3=open("ulog.txt","w")
            ff3.write(uname)
            ff3.close()

            ff3=open("sms.txt","w")
            ff3.write("yes")
            ff3.close()

            return redirect(url_for('userhome'))
        else:
            # Account doesnt exist or username/password incorrect
            msg = 'Incorrect username/password!'
```

```
    return render_template('login_farmer.html',msg=msg,msg1=msg1)


@app.route('/userhome', methods=['GET', 'POST'])
def userhome():
    msg=""

    msg=""
    act=request.args.get("act")
    act2=request.args.get("act2")
    act3=request.args.get("act3")

    return render_template('userhome.html',msg=msg,act=act,act2=act2,act3=act3)

@app.route('/register', methods=['GET', 'POST'])
def register():
    msg=""

    if request.method=='POST':
        name=request.form['name']
        mobile=request.form['mobile']
        email=request.form['email']
        location=request.form['location']
        uname=request.form['uname']
        pwd=request.form['pass']


        mycursor = mydb.cursor()
```

```
            mycursor.execute("SELECT max(id)+1 FROM farmer")
            maxid = mycursor.fetchone()[0]
            if maxid is None:
                maxid=1
            sql = "INSERT INTO farmer(id,name,mobile,email,location,uname,pass) VALUES (%s, %s, %s, %s,
%s, %s, %s)"
            val = (maxid,name,mobile,email,location,uname,pwd)
            mycursor.execute(sql, val)
            mydb.commit()
            print(mycursor.rowcount, "Added Success")
            act='success'
            return redirect(url_for('login_farmer',act=act))
        except:
            print("excep")



    msg1=""
    msg2=""
    mess=""
    ff=open("get_value.txt","r")
    get_value=ff.read()
    ff.close()
    s=""
    if get_value=="":
        s="1"
    else:

        msg1=get_value+" detected, "

    ff1=open("person.txt","r")
    pp=ff1.read()
```

```
    ff1.close()
    sc=""
    if pp=="":
        sc="1"
    else:
        msg2=""+pp+""

    mess=msg1+" "+msg2
    '''
    return render_template('process.html',mess=mess,act=act)

def object_detect(fname):
    # construct the argument parse
    parser = argparse.ArgumentParser(
        description='Script to run MobileNet-SSD object detection network ')
    parser.add_argument("--video", help="path to video file. If empty, camera's stream will be used")
    parser.add_argument("--prototxt", default="MobileNetSSD_deploy.prototxt",
                                  help='Path to text network file: '
                                      'MobileNetSSD_deploy.prototxt for Caffe model or '
                                      )
    parser.add_argument("--weights", default="MobileNetSSD_deploy.caffemodel",

    # MobileNet requires fixed dimensions for input image(s)
    # so we have to ensure that it is resized to 300x300 pixels.
    # set a scale factor to image because network the objects has differents size.
    # We perform a mean subtraction (127.5, 127.5, 127.5) to normalize the input;
    # after executing this command our "blob" now has the shape:
    # (1, 3, 300, 300)
    blob = cv2.dnn.blobFromImage(frame_resized, 0.007843, (300, 300), (127.5, 127.5, 127.5), False)
    #Set to network the input blob
    net.setInput(blob)
```

```
#Prediction of network
detections = net.forward()

#Size of frame resize (300x300)
cols = frame_resized.shape[1]
rows = frame_resized.shape[0]

#For get the class and location of object detected,
# There is a fix index for class, location and confidence
# value in @detections array .
for i in range(detections.shape[2]):
    confidence = detections[0, 0, i, 2] #Confidence of prediction
    if confidence > args.thr: # Filter prediction
        class_id = int(detections[0, 0, i, 1]) # Class label

        # Object location
        xLeftBottom = int(detections[0, 0, i, 3] * cols)
        yLeftBottom = int(detections[0, 0, i, 4] * rows)
        xRightTop   = int(detections[0, 0, i, 5] * cols)
        yRightTop   = int(detections[0, 0, i, 6] * rows)

        # Factor for scale to original size of frame
        heightFactor = frame.shape[0]/300.0
        widthFactor = frame.shape[1]/300.0
        # Scale object detection to frame
        xLeftBottom = int(widthFactor * xLeftBottom)
        yLeftBottom = int(heightFactor * yLeftBottom)
        xRightTop   = int(widthFactor * xRightTop)
        yRightTop   = int(heightFactor * yRightTop)
        # Draw location of object
        cv2.rectangle(frame, (xLeftBottom, yLeftBottom), (xRightTop, yRightTop),
```

```
        (0, 255, 0))
try:
    y=yLeftBottom
    h=yRightTop-y
    x=xLeftBottom
    w=xRightTop-x
    image = cv2.imread("static/dataset/"+fname)
    mm=cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
    #cv2.imwrite("static/trained/c_"+fname, mm)
    cropped = image[yLeftBottom:yRightTop, xLeftBottom:xRightTop]

    #gg="segment.jpg"
    #cv2.imwrite("static/result/"+gg, cropped)



    #mm2 = PIL.Image.open('static/trained/'+gg)
    #rz = mm2.resize((300,300), PIL.Image.ANTIALIAS)
    #rz.save('static/trained/'+gg)
except:
    print("none")
    #shutil.copy('getimg.jpg', 'static/trained/test.jpg')
# Draw label and confidence of prediction in frame resized
if class_id in classNames:
    label = classNames[class_id] + ": " + str(confidence)
    claname=classNames[class_id]

    aid=0
    if claname=="Bear":
        aid=1
    elif claname=="Cow":
        aid=2
```

```
        elif claname=="Elephant":
          aid=3
        elif claname=="Goat":
          aid=4
        elif claname=="Horse":
          aid=5
        elif claname=="Pig":
          aid=1
        elif claname=="Sheep":
          aid=1

        #mycursor.execute("update train_data set animal_id=%s where id=%s",(aid,rw[0]))
        #mydb.commit()

        labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.5, 1)

        yLeftBottom = max(yLeftBottom, labelSize[1])
        cv2.rectangle(frame, (xLeftBottom, yLeftBottom - labelSize[1]),
                      (xLeftBottom + labelSize[0], yLeftBottom + baseLine),
                      (255, 255, 255), cv2.FILLED)
        cv2.putText(frame, label, (xLeftBottom, yLeftBottom),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))

        #print(label) #print class and confidence


    # Object location
    xLeftBottom = int(detections[0, 0, i, 3] * cols)
    yLeftBottom = int(detections[0, 0, i, 4] * rows)
    xRightTop   = int(detections[0, 0, i, 5] * cols)
```

```
yRightTop   = int(detections[0, 0, i, 6] * rows)

# Factor for scale to original size of frame
heightFactor = frame.shape[0]/300.0
widthFactor = frame.shape[1]/300.0
# Scale object detection to frame
xLeftBottom = int(widthFactor * xLeftBottom)
yLeftBottom = int(heightFactor * yLeftBottom)
xRightTop   = int(widthFactor * xRightTop)
yRightTop   = int(heightFactor * yRightTop)
# Draw location of object
cv2.rectangle(frame, (xLeftBottom, yLeftBottom), (xRightTop, yRightTop),
          (0, 255, 0))
try:
   y=yLeftBottom
   h=yRightTop-y
   x=xLeftBottom
   w=xRightTop-x
   image = cv2.imread("static/dataset/"+fnn)
   mm=cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
   cv2.imwrite("static/result/"+fnn, mm)
   cropped = image[yLeftBottom:yRightTop, xLeftBottom:xRightTop]
   gg="segment.jpg"
   cv2.imwrite("static/result/"+gg, cropped)
   #mm2 = PIL.Image.open('static/trained/'+gg)
   #rz = mm2.resize((300,300), PIL.Image.ANTIALIAS)
   #rz.save('static/trained/'+gg)
except:
   print("none")
   #shutil.copy('getimg.jpg', 'static/trained/test.jpg')
# Draw label and confidence of prediction in frame resized
```

```
        if class_id in classNames:
            label = classNames[class_id] + ": " + str(confidence)
            labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.5, 1)

            yLeftBottom = max(yLeftBottom, labelSize[1])
            cv2.rectangle(frame, (xLeftBottom, yLeftBottom - labelSize[1]),
                          (xLeftBottom + labelSize[0], yLeftBottom + baseLine),
                          (255, 255, 255), cv2.FILLED)
            cv2.putText(frame, label, (xLeftBottom, yLeftBottom),
                     cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))

            print(label) #print class and confidence
    ####################
    return render_template('anitest.html',act=act,msg=msg,fnn=fnn,aud=aud)



@app.route('/result', methods=['GET', 'POST'])
def result():
    res=""
    afile="a3.mp3"
    password_provided = "xyz" # This is input in the form of a string
    password = password_provided.encode() # Convert to type bytes
    salt = b'salt_' # CHANGE THIS - recommend using a key from os.urandom(16), must be of type
bytes
    kdf = PBKDF2HMAC(
        algorithm=hashes.SHA256(),
        length=32,
        salt=salt,
        iterations=100000,
        backend=default_backend()
    )
```

```
key = base64.urlsafe_b64encode(kdf.derive(password)) # Can only use kdf once

f2=open("log.txt","r")

vv=f2.read()
```

11.2. SCREENSHOT

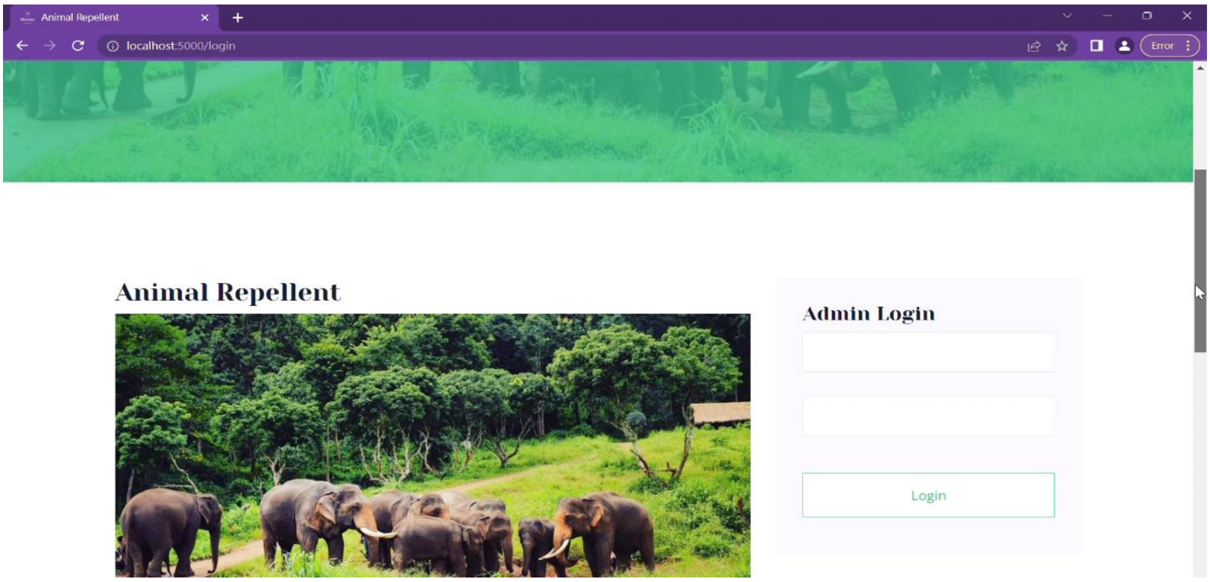


Figure no:11.1 Admin login

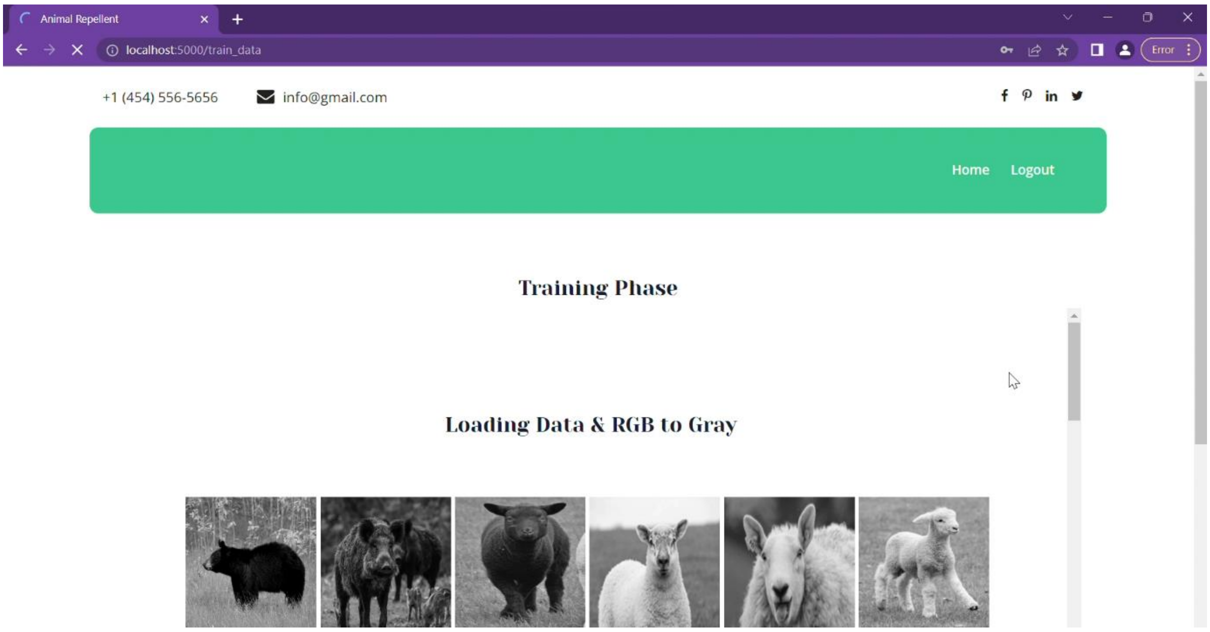


Figure no:11.2 Training Phase

Figure no:11.3 Processing -Noice Filter

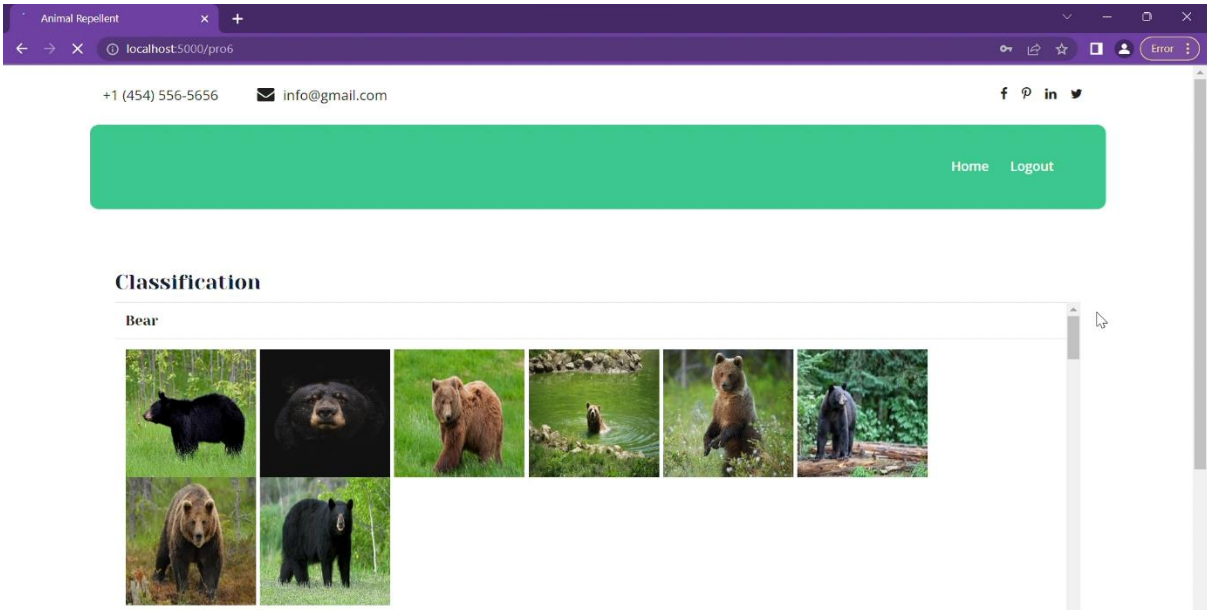


Figure no:11.4 Classification

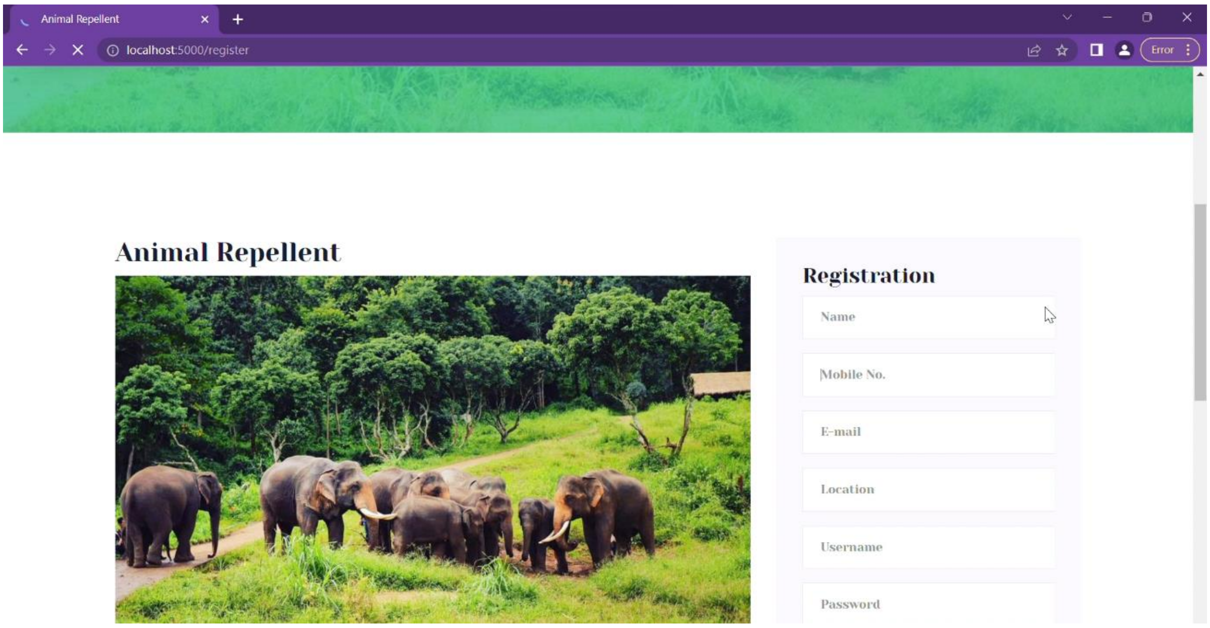


Figure no:11.5 Farmer Registration

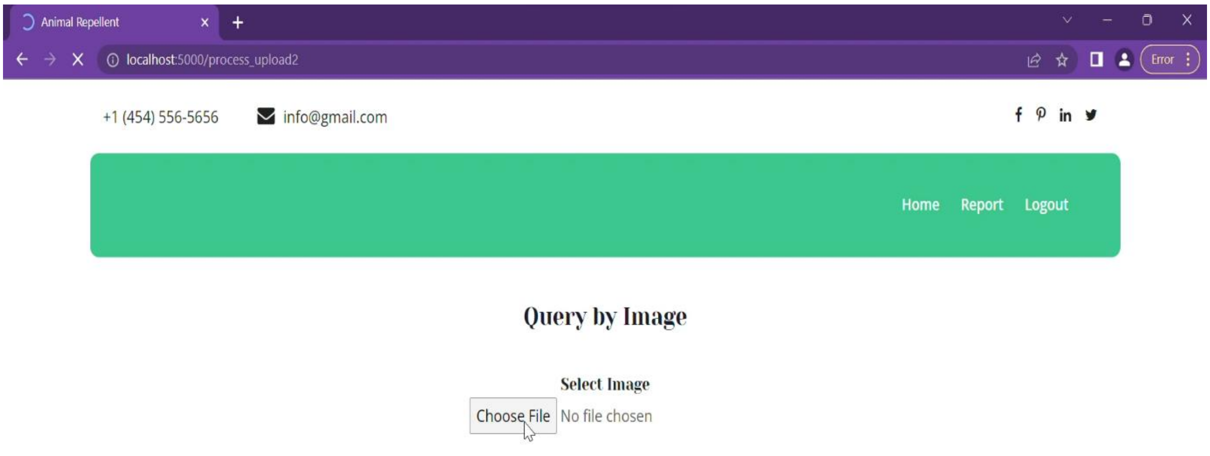


Figure no:11.6 Query By Image

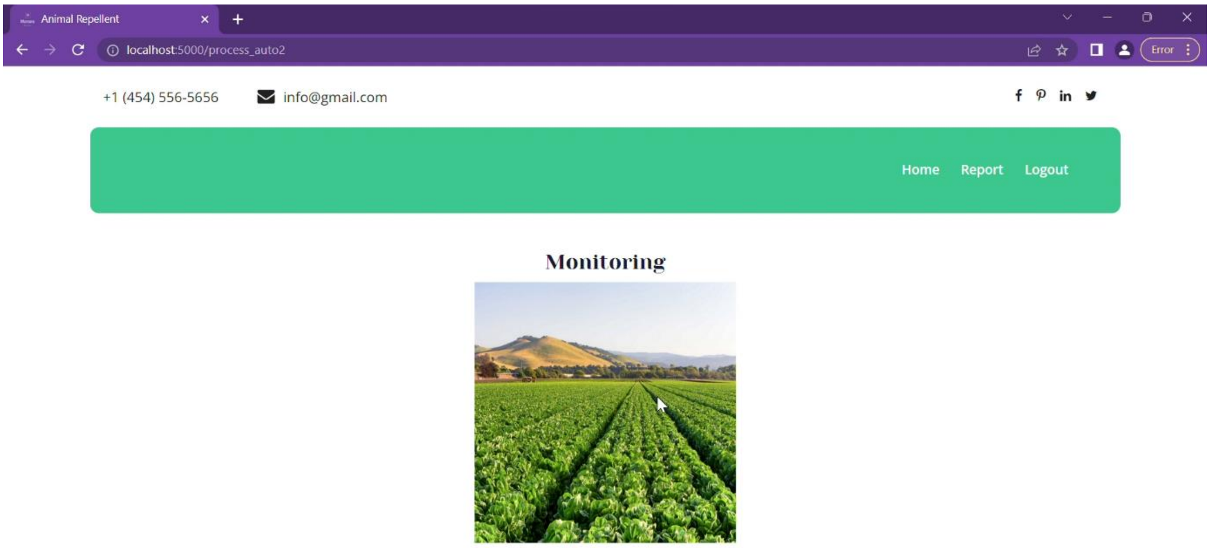


Figure no:11.7 Training Monitoring

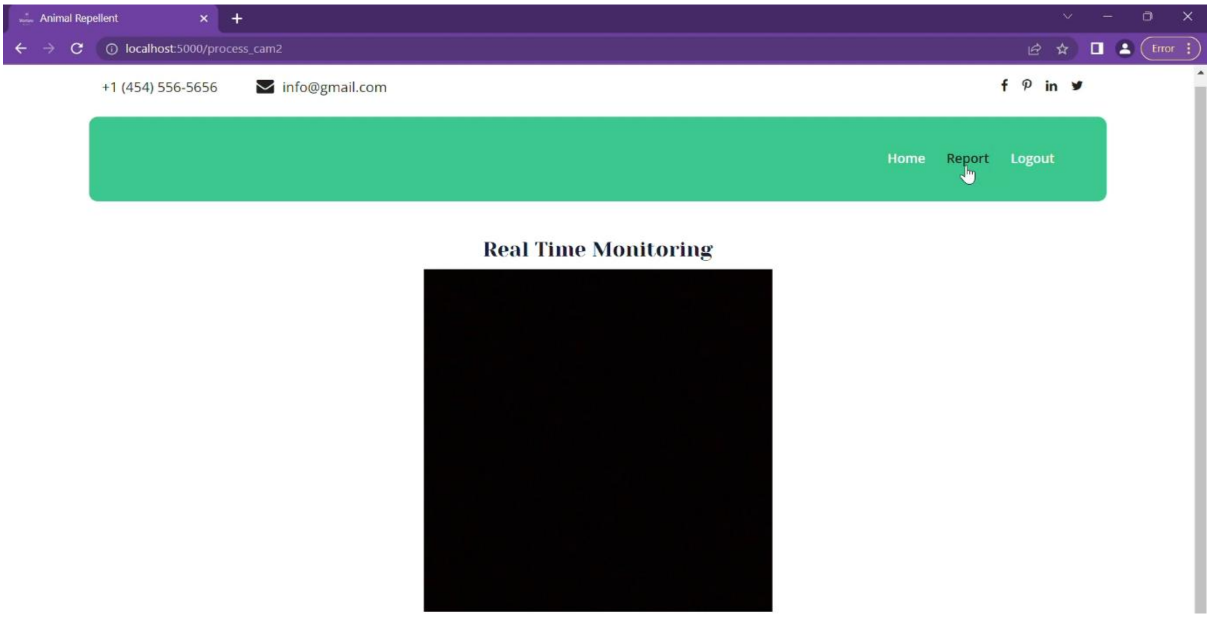


Figure no:11.8 Real Time Monitoring

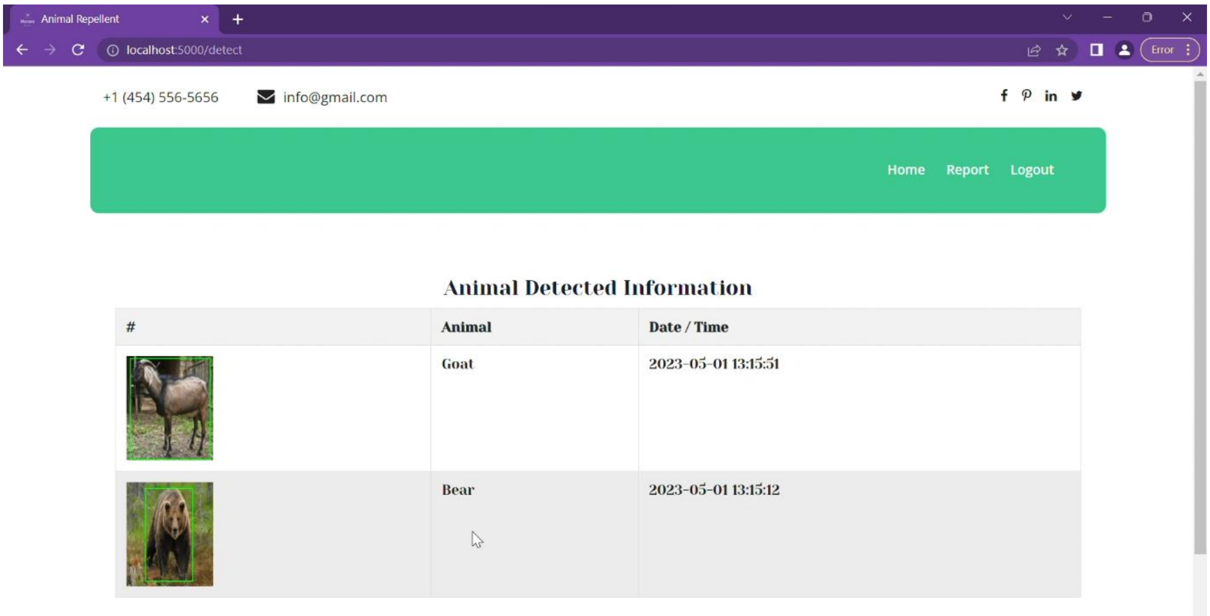


Figure no:11.9 Animal Detected Information

CHAPTER 12

REFERENCES

[1] M. De Clercq, A. Vats, and A. Biel, ``Agriculture 4.0: The future of farmingtechnology," in Proc. World Government Summit, Dubai, UAE, 2018,pp. 11-13.

[2] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, ``Fromindustry 4.0 to agriculture 4.0: Current status, enabling technologies,and research challenges," IEEE Trans. Ind. Informant., vol. 17, no. 6,pp. 4324334, Jun. 2021.

[3] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, ``A surveyon the role of IoT in agriculture for the implementation of smart farming,"IEEE Access, vol. 7, pp. 156237-156271, 2019.

[4] K. Kirkpatrick, ``Technologizing agriculture," Common. ACM, vol. 62,no. 2, pp. 14-16, Jan. 2019.

[5] A. Farooq, J. Hu, and X. Jia, ``Analysis of spectral bands and spatial resolutions for weed classification via deep convolutional neural network,"IEEE Geosci. Remote Sens. Lett., vol. 16, no. 2, pp. 183-187,Feb. 2018.

[6] M. Apollonio, S. Ciuti, L. Pedrotti, and P. Banti, ``Ungulates and theirmanagement in Italy," in European Ungulates and Their Managementin the 21th Century. Cambridge, U.K.: Cambridge Univ. Press, 2010,pp. 475-505.

[7] A. Amici, F. Serrani, C. M. Rossi, and R. Primi, ``Increase in crop damagecaused by wild boar (Sus scrofa L.): The `refuge effect,'" AgronomySustain. Develop., vol. 32, no. 3, pp. 683-692, Jul. 2012.

[8] S. Giordano, I. Seitanidis, M. Ojo, D. Adami, and F. Vignoli, ``IoT solutionsfor crop protection against wild animal attacks," in Proc. IEEE Int.Conf. Environ. Eng. (EE), Mar. 2018, pp. 1-5.

[9] M. O. Ojo, D. Adami, and S. Giordano, ``Network performance evaluationof a LoRa-based IoT system for crop protection against ungulates," inProc. IEEE 25th Int.Workshop Comput. Aided Modeling Design Commun.Links Netw. (CAMAD), Sep. 2020, pp. 1-6.

[10] H. E. Heefner and R. S. Heffner, ``Auditory perception," in Farm Animalsand the Environment, C. Phillips and D. Piggins, Eds. Wallingford, U.K.:CAB International, 1992.

[11] https://www.indiatimes.com/news/india/in-the-battle-between-man-vs-wild-highways-railway- lines-emerge-as-new-challenges-375504.html

[12] https://adventuresinmachinelearning.com/neural-networks-tutorial/

[13] https://github.com/tensorflow/models

CHAPTER 13

CERTIFICATES

13.1. Conference Certificates

SRI SAI REC SRI SAI RANGANATHAN ENGINEERING COLLEGE

Approved by AICTE New Delhi Affiliated to Anna University,Chennai

Department of Computer Science and Engineering & Information Technology

Proudly Presents

International Conference on Innovative Engineering and Information Technology

ICIEIT' 2K23

CERTIFICATE

This is to certify that Dr. /Prof./Mr. /Mrs. /Ms. R.MURUGESAN *of* PAAVAI COLLEGE OF ENGINEERING *has presented a paper titled* IMPLEMENTATION OF AGRICULTURE BASED IN ANIMAL REPELLENT SYSTEM USING DEEP LEARNING *in the International Conference on Innovative Engineering and Information Technology(ICIEIT' 23) held at Sri Sai Ranganathan Engineering College , Coimbatore during 28th,29th April 2023.*

PRINCIPAL CHAIRMAN

SRI SAI REC SRI SAI RANGANATHAN ENGINEERING COLLEGE

Approved by AICTE New Delhi Affiliated to Anna University,Chennai

Department of Computer Science and Engineering & Information Technology

Proudly Presents

International Conference on Innovative Engineering and Information Technology

ICIEIT' 2K23

CERTIFICATE

This is to certify that Dr. /Prof./Mr. /Mrs. /Ms. JAYASURYA.S *of* PAAVAI COLLEGE OF ENGINEERING *has presented a paper titled* IMPLEMENTATION OF AGRICULTURE BASED IN ANIMAL REPELLENT SYSTEM USING DEEP LEARNING *in the International Conference on Innovative Engineering and Information Technology(ICIEIT' 23) held at Sri Sai Ranganathan Engineering College , Coimbatore during 28th,29th April 2023.*

PRINCIPAL CHAIRMAN

SRI SAI REC SRI SAI RANGANATHAN ENGINEERING COLLEGE

Approved by AICTE New Delhi Affiliated to Anna University,Chennai

Department of Computer Science and Engineering & Information Technology

Proudly Presents

International Conference on Innovative Engineering and Information Technology

ICIEIT' 2K23

CERTIFICATE

This is to certify that Dr. /Prof./Mr. /Mrs. /Ms. **MAHESUWARAN.S** ***of*** **PAAVAI COLLEGE OF ENGINEERING** ***has presented a paper titled*** **IMPLEMENTATION OF AGRICULTURE BASED IN ANIMAL REPELLENT SYSTEM USING DEEP LEARNING** ***in the International Conference on Innovative Engineering and Information Technology(ICIEIT' 23) held at Sri Sai Ranganathan Engineering College , Coimbatore during 28th,29th April 2023.***

PRINCIPAL **CHAIRMAN**

SRI SAI REC SRI SAI RANGANATHAN ENGINEERING COLLEGE

Approved by AICTE New Delhi Affiliated to Anna University,Chennai

Department of Computer Science and Engineering & Information Technology

Proudly Presents

International Conference on Innovative Engineering and Information Technology

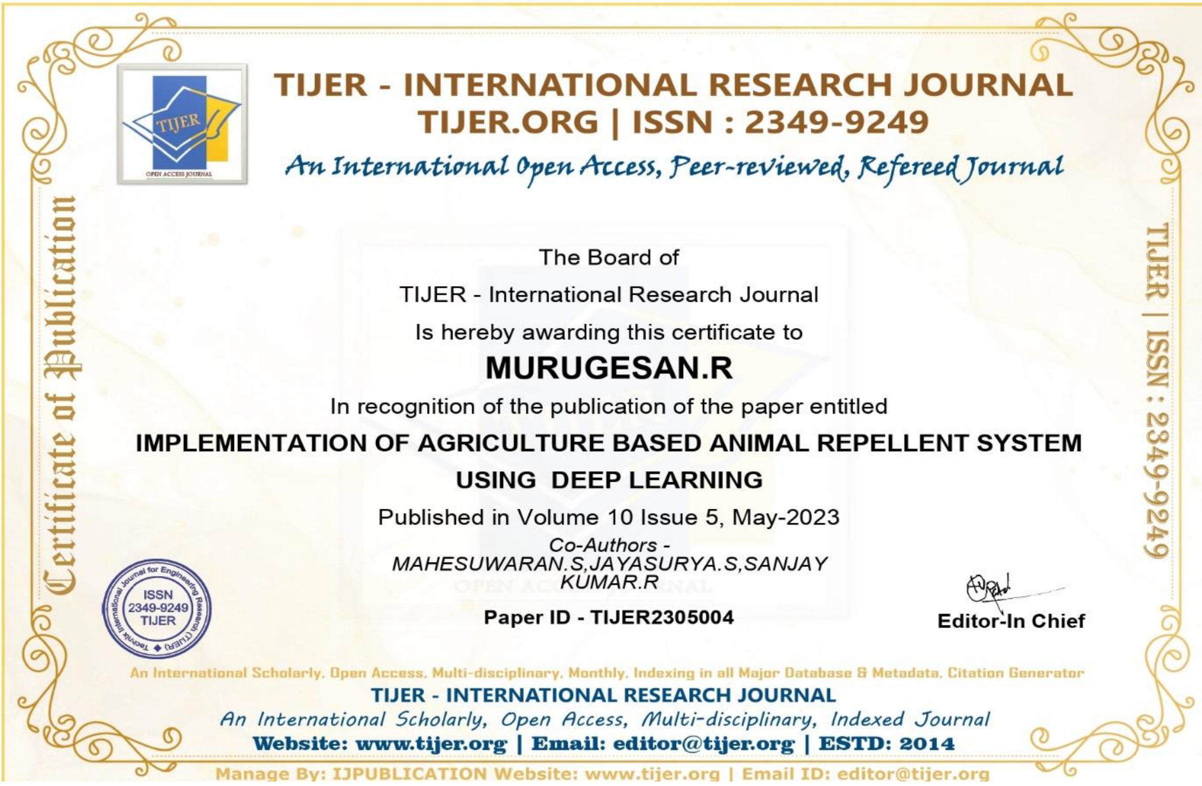
ICIEIT' 2K23

CERTIFICATE

This is to certify that Dr. /Prof./Mr. /Mrs. /Ms. **SANJAY KUMAR.R** ***of*** **PAAVAI COLLEGE OF ENGINEERING** ***has presented a paper titled*** **IMPLEMENTATION OF AGRICULTURE BASED IN ANIMAL REPELLENT SYSTEM USING DEEP LEARNING** ***in the International Conference on Innovative Engineering and Information Technology(ICIEIT' 23) held at Sri Sai Ranganathan Engineering College , Coimbatore during 28th,29th April 2023.***

PRINCIPAL **CHAIRMAN**

13.2. Journal Certificates



TIJER - INTERNATIONAL RESEARCH JOURNAL
TIJER.ORG | ISSN : 2349-9249
An International Open Access, Peer-reviewed, Refereed Journal

Certificate of Publication

TIJER | ISSN : 2349-9249

The Board of
TIJER - International Research Journal
Is hereby awarding this certificate to

MURUGESAN.R

In recognition of the publication of the paper entitled

IMPLEMENTATION OF AGRICULTURE BASED ANIMAL REPELLENT SYSTEM USING DEEP LEARNING

Published in Volume 10 Issue 5, May-2023

Co-Authors - MAHESUWARAN.S,JAYASURYA.S,SANJAY KUMAR.R

Paper ID - TIJER2305004

Editor-In Chief

ISSN 2349-9249 TIJER

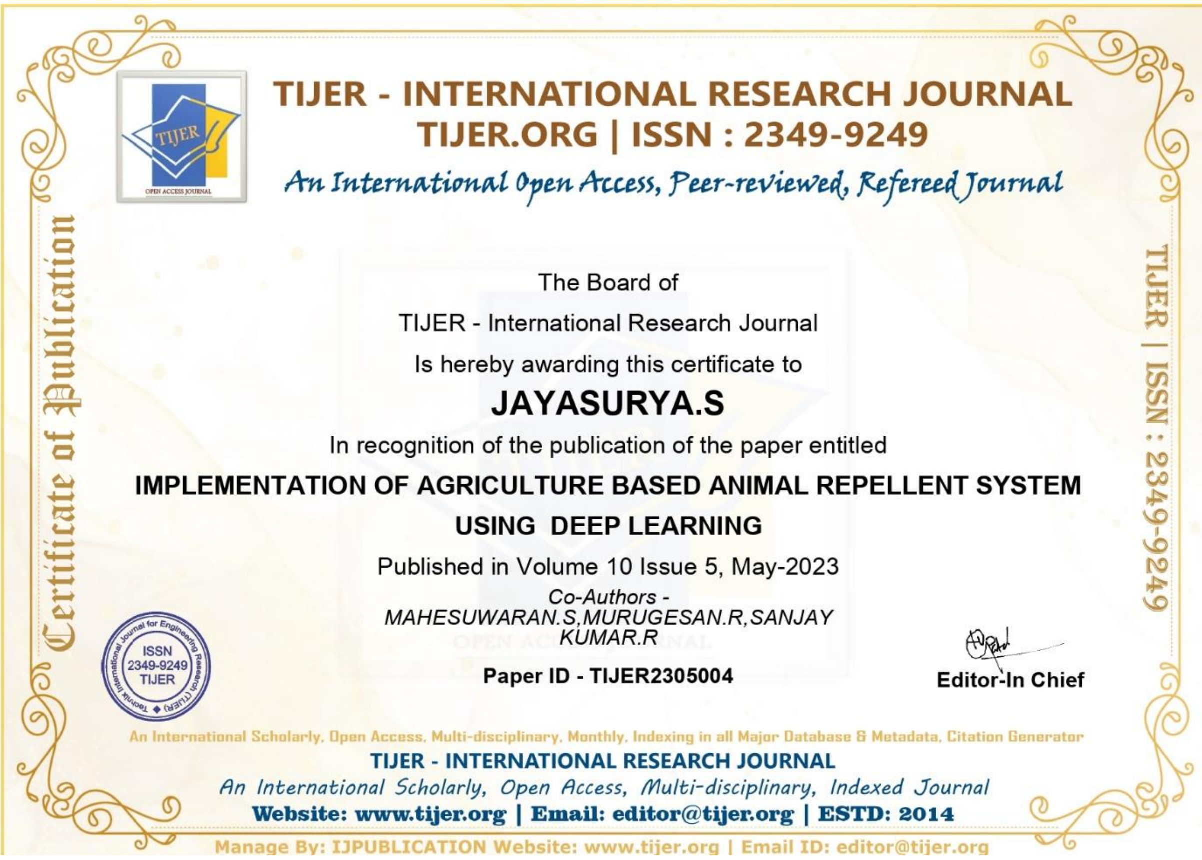
An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator

TIJER - INTERNATIONAL RESEARCH JOURNAL

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.tijer.org | Email: editor@tijer.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.tijer.org | Email ID: editor@tijer.org



TIJER - INTERNATIONAL RESEARCH JOURNAL
TIJER.ORG | ISSN : 2349-9249
An International Open Access, Peer-reviewed, Refereed Journal

Certificate of Publication

TIJER | ISSN : 2349-9249

The Board of
TIJER - International Research Journal
Is hereby awarding this certificate to

JAYASURYA.S

In recognition of the publication of the paper entitled

IMPLEMENTATION OF AGRICULTURE BASED ANIMAL REPELLENT SYSTEM USING DEEP LEARNING

Published in Volume 10 Issue 5, May-2023

Co-Authors - MAHESUWARAN.S,MURUGESAN.R,SANJAY KUMAR.R

Paper ID - TIJER2305004

Editor-In Chief

ISSN 2349-9249 TIJER

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator

TIJER - INTERNATIONAL RESEARCH JOURNAL

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.tijer.org | Email: editor@tijer.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.tijer.org | Email ID: editor@tijer.org

Certificate of Publication

TIJER | ISSN : 2349-9249

TIJER - INTERNATIONAL RESEARCH JOURNAL
TIJER.ORG | ISSN : 2349-9249
An International Open Access, Peer-reviewed, Refereed Journal

The Board of

TIJER - International Research Journal

Is hereby awarding this certificate to

MAHESUWARAN.S

In recognition of the publication of the paper entitled

IMPLEMENTATION OF AGRICULTURE BASED ANIMAL REPELLENT SYSTEM USING DEEP LEARNING

Published in Volume 10 Issue 5, May-2023

Co-Authors - MURUGESAN.R,JAYASURYA.S,SANJAY KUMAR.R

ISSN 2349-9249 TIJER

Paper ID - TIJER2305004

Editor-In Chief

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator
TIJER - INTERNATIONAL RESEARCH JOURNAL
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.tijer.org | Email: editor@tijer.org | ESTD: 2014
Manage By: IJPUBLICATION Website: www.tijer.org | Email ID: editor@tijer.org

Certificate of Publication

TIJER | ISSN : 2349-9249

TIJER - INTERNATIONAL RESEARCH JOURNAL
TIJER.ORG | ISSN : 2349-9249
An International Open Access, Peer-reviewed, Refereed Journal

The Board of

TIJER - International Research Journal

Is hereby awarding this certificate to

SANJAY KUMAR.R

In recognition of the publication of the paper entitled

IMPLEMENTATION OF AGRICULTURE BASED ANIMAL REPELLENT SYSTEM USING DEEP LEARNING

Published in Volume 10 Issue 5, May-2023

Co-Authors - MAHESUWARAN.S,MURUGESAN.R,JAYASURYA.S

ISSN 2349-9249 TIJER

Paper ID - TIJER2305004

Editor-In Chief

An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator
TIJER - INTERNATIONAL RESEARCH JOURNAL
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.tijer.org | Email: editor@tijer.org | ESTD: 2014
Manage By: IJPUBLICATION Website: www.tijer.org | Email ID: editor@tijer.org