Animal Trespassing Detection System

Anand Rakesh
Department of Computer Science
Mar Baselios College of Engineering and Technology
Trivandrum, India

Anand Vinod
Department of Computer Science
Mar Baselios College of Engineering and Technology
Trivandrum, India

Mridhul Madhu
Department of Computer Science
Mar Baselios College of Engineering and Technology
Trivandrum, India

Roshan Daniel
Department of Computer Science
Mar Baselios College of Engineering and Technology
Trivandrum,India

Jisha Jose
Asst. professor
Department of Computer Science
Mar Baselios College of Engineering and Technology
Trivandrum, India

Robin Joseph
Asst. professor
Department of Computer Science
Mar Baselios College of Engineering and Technology
Trivandrum, India

***Abstract*—Protection of our home land is more of a necessity than a luxury. Detection system is a term that we all have been familiar with for the past few years. Over the years, the hardware and the technology used to develop this system has been constantly upgraded to assist in detecting intrusions with the help of machine learning and Artificial Intelligence. In modern security systems, cameras are used in detection, especially when there is a need to perceive intrusions during circumstantial conditions. Implementation of cameras in security systems leads to a substantial increase of information that is fed to the security system operator. The operator is the person that manages the security systems. Usage of image processing systems help the operator by enhancing the relevant information of the image, which enables to discern important details of the image. In our project, we implement an animal trespassing detection system with the help of cameras and image processing. In this projectwe are planning to implement by using a camera on a spot further away from the property. When an animal comes into the Fieldof View (FOV) of the camera, the device scans the output image using machine learning as well as deep learning algorithms to detect the characteristics of the animal. If the device recognizes the animal as an elephant or a boar, an alarm will be issued across the area, and thus the farmers can avert the animal with or without minimal loss. The device can also be implemented in such a way that it can be used to prevent human casualties during the event of wild animal trespassing by warning the farmersof where the animal is present. We use python programming language and main hardware components used for this project are cameras and the hardware used is Raspberry Pi 4.**

***Index Terms*—Rasspberry Pi 4, camera**

I. INTRODUCTION

Protection of our home land is more of a necessity than a luxury. Detection system is a term that we all have been KSCSTE Student Project Scheme Financial Assistance. familiar with for the past few years. Over the years, the hardware and the technology used to develop this system has been constantly upgraded to assist in detecting intrusions with the help of machine learning and Artificial Intelligence. In modern security systems, cameras are used in detection, especially when there is a need to perceive intrusions during circumstantial conditions. Implementation of cameras in se- curity systems leads to a substantial increase of information that is fed to the security system operator. The operator is the person that manages the security systems. Usage of image processing systems help the operator by enhancing the relevant information of the image, which enables to discern important details of the image. In our project, we implement an animal trespassing detection system with the help of cameras and image processing. In many of the rural areas to the north of Kerala like Wayanad, there are acres of land with richagriculture that provides a means of employment to the local people as well as buyers. However, these places are also hometo dangerous wild animals like Elephants, Wild Boars and these animals damage the crops and the farmers have to bear the loss. This loss can exceed up to lakhs and crores. In this project we are planning to implement by using a camera on a spot further away from the property. When an animal comes into the Field of View (FOV) of the camera, the device scans the output image using machine learning as well as deep learning algorithms to detect the characteristics of the animal. If the device recognizes the animal as an elephant ora boar, an alarm will be issued across the area, and thus the farmers can avert the animal with or without minimal loss. The device can also be implemented in such a way that it can be used to prevent human casualties during the event of wild animal trespassing by warning the farmers of where the animal is present. We use python programming language and main hardware components used for this project are cameras and the hardware used is Raspberry Pi 4.

II. LITERATURE SURVEY

A. Prajna P, Soujanya B.S., Mrs. Divya, "IoT based Wild Animal Intrusion Detection System", IJERT-2018

This research is based on detection of movement of animals using PIR Sensors and cameras, where the PIR sensors are used for detection of movement. Here a micro-controller is used to transmit the image captured by the camera, triggered by the sensor for processing. The processing and classification of animals from images captured by the camera via the micro- controller, is performed by using a PC. The PC will send the signal to the repellent system via the micro-controller to take appropriate action, if the animal is found to be a threat[1].

B. Saieshwar Radhakrishnan, R. Ramanathan "A Support Vector Machine with Gabor Features for Animal Intrusion Detection in Agriculture Fields", 2018.

This study is based on an animal detection system by using approaches based on image processing and machinelearning. The image is then divided with the help of a watershed algorithm in order to extract the features of an image. It then examines whether the animal found in the image is a threat or not. The features are extracted from the training image data-sets 4 using Two-dimensional Gabor filterbank. The algorithm is used to create a barrier which acts as the contour depending on certain criteria. The criteria is that the marked regions must meet different markers. This study ensures to increase efficiency of the training model by raising the number of training data-sets in-order to find the minimum possible combination of the filter bank and test images. Another example of a supervised learning algorithm is the Linear SVM. It is used to train data in order to classify between text and hypertext [2].

C. K. Jai Santoshi, Bhavana. S. "Intruder recognition in a farm through wireless sensor network", IJARIIT-2018

This study uses wireless sensor network (WSN) technology to detect intrusions in agricultural lands. Motion sensors are placed at various locations in order to sense movements and communicate the event to the organizer using a Radio fre- quency transceiver. The system performs detection and the or- ganizer is alerted first. Then he/she sends an alert call/message to the farm owner's mobile number using a GSM module. An Arduino board is fixed in such a way that it is close to the centralized sensor. The GSM module is interfaced with buzzers and RFID transmitters. Radio-frequency identification (RFID) tags are used in order to differentiate between unauthorized and authorized entries in the farmland.[3]

III. PROPOSED SYSTEM

We are proposing a system that can train the model, which consists of thousands of image datasets of animals like Ele- phants, Wild Boar, Wild Buffalo etc . Then raising conditions in the model in such a way that the system will raise detection alerts for these animals. The training process of the model to detect the animals in the dataset is done using either Google Colab or Anaconda on our own desktops which will make the detection easier and faster, which will be very slow in Raspberry Pi due to CPU core limitations and absence of a dedicated or integrated GPU. Hence to make the processing easier the trained model will be converted to TFLite.

A. *Advantages of our proposed system*

- The advantages of our system are that it contains real-time animal/human detection.
- It alerts forest officials and the public in case of a threat
- It will help in reducing panic due to fake alarms.
- It will help in identifying the animal which will help toact accordingly.)

B. *Design flow diagram of the system*

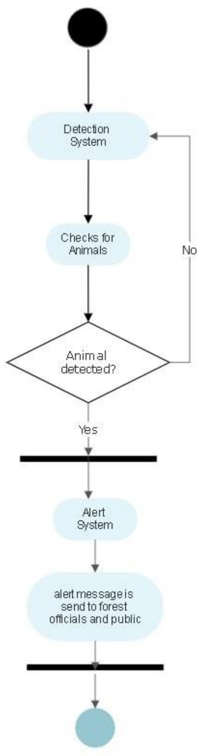


Fig. 1. Design Flow Diagram

a) W: e will be first training the model using our desktops using the image datasets that we have collected. After successful detection, we shift the entire process onto the Raspberry Pi board and move on to real-time recognition. Once real-time object detection is achieved, we will move onto the next step of classification and localization. After these steps, we will train the model in such a way that in a multiple object scenario, each object can be classified and localized to its own area.

C. *Resources required*

a) Datasets: The data set that we have collected are thousands of pictures of wild elephants, wild boars, and other animals as such used to train the model to classify the animals. Our training dataset consists of 25,000+ images with sizes ranging from 200x200 pixels to 500x500 pixels. Each of these pictures must be unique and distinct to each other.

b) Components:

- Raspberry Pi 4: Raspberry Pi 4 is a single-board DIY friendly processing unit used to learn programming skills, build hardware projects and even be used in industrial applications. Raspberry Pi has seen a rise in usage over the past few years mainly in the field of Home Automa- tion and other DIY projects for students and professionals alike. The Raspberry Pi is very essential for running machine learning projects and exploring the concept of Internet of Things (IoT). However, since Raspberry Pi is a small processing unit many of these projects require additional modules for the board to help process the codeand the workflow.
- REES52 Raspberry Pi 5MP OV5647 Night Vision Cam- era Module is the module that we will be using for our project
- A suitable desktop that can run TensorFlow Lite and Raspberry Pi OS.

c) Softwares, Libraries and Models:

- Google Colaboratory: Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows any- body to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.
- OpenCV: It is a library of several programming functions mainly aimed for using in real-time computer vision applications.It was originally developed by Intel, and was later supported by Willow Garage then by Itseez. The library is a cross-platform and also free to use under the open-source Apache 2 License.
- Tensorflow: TensorFlow is a free and open-source soft- ware library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a sym- bolic math library based on dataflow and differentiable programming.
- Visual Studio Code: Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
- Anaconda: Anaconda is a distribution of the program- ming languages Python and R which is used for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.
- MobileNet V2: MobileNetV2 is a convolutional neural network architecture that seeks to perform well on mobileplatforms and devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. To filter features as a source of non- linearity, the intermediate expansion layer will be using lightweight depthwise convolutions. We mainly chose MobileNet V2 because of its lesser processing content which will help to process the model faster in Raspberry Pi.

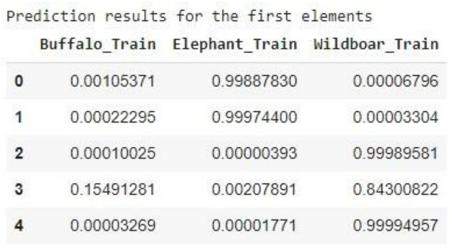
IV. Methodology

In this project various algorithms and pretrained models were used for testing the effective classification and detection on both personal workstations and mobile processors.

A. Image Classification of Animals

a) Algorithm:

- First all the required libraries are imported into the Google Colab IDE (Tensorflow, matplotlib, numpy, pan- das, etc).
- Using the ImageDataGenerator class various batches of images are created. ImageDatagenerator is a class used for expanding a training dataset inorder to improve the performance of the model to generalize.
- Two sets of batches will be created, One for training and one for validation.
- Export the pre-trained model from Tensorflow Hub. Here,we’ll be using MobileNet V2 model.
- After exporting, the pretrained model is now trained using our specifications and configurations. The time of executing depends on the number of epochs assigned to the training phase.
- After displaying the final accuracy and loss, plot the graph for both accuracy and loss using matplotlib library.
- Export the currently trained model and save it for further training and validation.
- Using the predict function an overall prediction percent- age of each label is brought as output
- Finally, the prediction is applied on a single batch of images.



Prediction results for the first elements

	Buffalo_Train	Elephant_Train	Wildboar_Train
0	0.00105371	0.99887830	0.00006796
1	0.00022295	0.99974400	0.00003304
2	0.00010025	0.00000393	0.99989581
3	0.15491281	0.00207891	0.84300822
4	0.00003269	0.00001771	0.99994957

Fig. 2. Prediction result

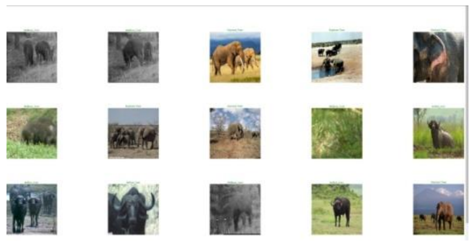


Fig. 3. classified image batch

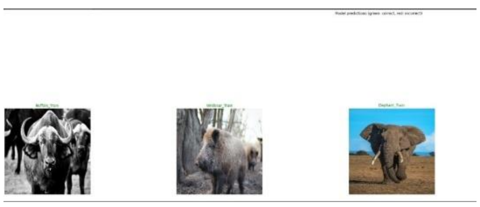


Fig. 4. test image prediction

B. *Object Detection Phase using MobileNet V2*

a) Training the model:

- First we have to import the follow-ing requirements (tensorflow=1.15.2, keras=2.3.1,imutils=0.5.3,numpy=1.18.2,opencv-python=4.2.0.*, matplotlib=3.2.1, scipy=1.4.1)
- Initialize the initial learning rate, number of epochs andbatch size for the training process.
- Perform one-hot encoding on the labels.
- Construct an image generator for training purposes to beused in data augmentation.
- Load the MobileNetV2 network and also ensure that thehead FC layer sets are all left off.
- After constructing, the head FC model is placed on topof the base model
- Compile the model.
- Train the head of the network and make predictions usingthe testing set.
- For each image in the testing set, we need to find the index of the label along with its corresponding predicted probability whose value is highest.
- We plot the respective graphs for both accuracy as wellas loss.
- Model is now trained successfully.

b) Detection:

- Import the trained model.
- First grab the dimensions of the frame and then construct a blob from it using cv
- Pass the blob through the network and obtain the detec- tion (In this case, we trained the model using the human dataset also as).
- Initialize our list of humans, elephants, buffalo and their corresponding coordinates , and the list of predictions from the model.
- Loop over the detection and extract the confi-dence(i.e.,probability) associated with the detection.
- Compute the (x, y)-coordinates of the bounding box for the object.
- Extract the image ROI, convert it from BGR to RGB channel ordering and resize it to 224x224, and preprocess it. Load the serialized detector model from disk.
- Initialize the video stream and loop over frames from the video stream.
- Grab the frame from the threaded video stream and resize it in order to achieve a maximum width of 400 pixels.
- Unpack the bounding box and predictions to determine the class label and color we'll use to draw the bounding box and text.
- Include the probability in the label and display the label aswell as the bounding box rectangle on the output frame.
- If, the webcam is ON, and detection starts taking place.
- Inorder to exit from the loop or webcam 'q' key is pressed

C. *Second Object Detection Phase using YOLOv4*

a) Training the Model:

- The object detection is done on webcam in google colab using yolov4. We will be using scaled-YOLOv4 (yolov4- csp), the fastest and most accurate object detector
- Import the following dependencies (evaljs, cv2imshow, b64decode, b64encode, cv2, numpy, PIL, io, mat-plotlib.pyplot as plt)
- Cloning and setting up Darknet for YOLOv4, we usedthe famous AlexeyAB's darknet repository to perform YOLOv4 detections.
- Change makefile to have GPU, OPENCV and LIBSO enabled
- Make darknet (builds darknet so that you can then use the darknet.py file and have its dependencies)
- In order to utilize YOLOv4 with Python code we will use some of the pre-built functions found within darknet.py by importing the functions into our workstation
- Import darknet functions to perform object detections then load in YOLOv4 architecture network and use darknet helper function to run detection on image
- Get image ratios to convert bounding boxes to proper size, run model on darknet style image to get detections
- Run test on person.jpg image that comes with repository
- Convert the JavaScript object into an OpenCV image anddecode base64 image then convert bytes to numpy array and last decode numpy array into OpenCV BGR image
- Convert OpenCV Rectangle bounding box image into base64 byte string to be overlaid on video stream

b) Detection: Running YOLOv4 on images taken from a webcam is fairly straight-forward. We will utilize code within Google Colab's Code Snippets that has a variety of usefulcode functions to perform various tasks. We will be using the code snippet for Camera Capture which runs JavaScript code to utilize the computer's webcam. The code snippet will take a webcam photo, which we will then pass into the YOLOv4 model for object detection.

- Resize the output to fit the video element and wait for Capture to be clicked and wait for Capture to be clicked.
- Get photo data and get OpenCV format image.
- Call the darknet helper on webcam image and loop

through detections and draw them on webcam image and at last save image.

- Errors will be thrown if the user does not have a webcam or if they do not, so grant the page permission to accessit.

D. Comparison of Object Detection Models, MobileNet V2 model and YOLOv4 model.

YOLOv4 is way better than MobileNet V2 object detection because the time taken for executing YOLOv4 was less compared to MobileNet V2, but in the case ofYOLOv4 we have to capture the image then only a label box appears while in MobileNet V2 it shows label box in real time. Overall YOLOv4 has high accuracy and low loss comparedto MobileNet V2

V. Conclusion

Animals trespassing onto agricultural fields has been a problem that existed even in biblical times. However, countlesshuman lives have been put on the line out of which there has been reported to have major deaths and major injuries. These attacks also have struck fear into the hearts of many,be it the forest officials and the public alike. We aim to help these farmers to extinguish their fear and concerns about their hard work that has been trampled by these animals. However, these animals are also a part of our ecosystem and we must make sure that they are not harmed because of their defensive actions. And as we humans do live alongside each other, we must also learn to live in harmony amongst these animals by reducing the amount of how much forests are being cut down every year.

References

[1] Prajna. P, Soujanya B.S, Mrs. Divya, IoT-based Wild Animal Intrusion Detection System, INTERNATIONAL JOURNAL OF ENGINEERINGRESEARCH TECHNOLOGY (IJERT) ICRTT - Issue 15, 2018

[2] Saieshwar Radhakrishnan, R.Ramanathan "A Support Vector Machine with Gabor Features for Animal Intrusion Detection in Agriculture Fields", 2018.

[3] K. Jai Santhoshi, Bhavana. S. "Intruder recognition in a farm through wireless sensor network", IJARIIT-2018.

[4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bot- tlenecks"; Proceedings of the IEEE Conference on Computer Vision andPattern Recognition (CVPR), pp. 4510-4520, 2018.

[5] [5]. Nirit Datta and Souvik Sarkar, "Automatic Tracking and Alarm System for Eradication of Wild Life Injury and Mortality," IEEE Conference, 2016.

[6] Sachin umesh sharma and dharmesh j. Shah, "A Practical Animal Detection and Collision Avoidance System Using Computer Vision Technique," Special section on innovations in electrical and computer engineering education," September 27, 2016.

[7] R.Shanmugasundaram and S.Pavithra, ,"IoT based animal tracking and monitoring system in Zoo," South Asian Journal of Engineering and Technology , Vol.3, No.2, 2017.

[8] Dr. P. Uma Maheswari and Anjali Rose Rajan , " Animal Intrusion Detection System Using Wireless Sensor Networks," International Journal of Advanced Research in Biology Engineering Science and Technology, Vol. 2, March 2016.