Animal Intrusion Detection Using ESP32 Cam and Open CV

Dr. Arulmurugan L
Associative professor
Department of Electronics Engineering
Bannari Amman Institute of Technology
Sathy, Tamil Nadu

Pradeep S
Department of Electronics Engineering
Bannari Amman Institute of Technology
Sathy, Tamil Nadu

Nikhil Raghav V
Department of Electronics Engineering
Bannari Amman Institute of Technology
Sathy, Tamil Nadu

Bharath Kumar S P
Department of Electronics Engineering
Bannari Amman Institute of Technology
Sathy, Tamil Nadu

Abstract – In forest and agricultural field human and animal conflict is a major problem where enormous amount if resources are lost and human and animal life are endangered. Due to this animal lives are endangered. People lose their crops, cattle and in some extreme cases they lose their lives. In current era of continuously developing IOT sensors and cloud technology, Image processing in remote areas became easy and cheap. To tackle this problem, we have developed a system with cameras and remote cloud server to capture live footage of area to be protected and processing the footage in a live server to detect animal intrusion. For hardware components, we have used ESP32 camera module with Wifi to capture live footages. The captured footage is sent to a remote server or PC for further processing. A yolov5 based pre-trained algorithm is used for object detection. To achieve maximum accuracy the algorithm is trained with thousands of animal photos multiple times until desired accuracy is reached (~98%). The footage is processed and based on the type of animal detected a message or alert is sent to the farmer or forest department from the server and a speaker can be used to scare the animals.

Keywords:- *Animal Detection, ESP32 CAM, Yolov5.*

I. INTRODUCTION

India is a global agricultural powerhouse. It is the world's largest producer of milk, pulses, and spices, and has the world's largest cattle herd, as well as the largest area under wheat, rice and cotton. Most of these lands are located in rural and forest area. Animal intrusion is unavoidable in these areas. As humans started to change forests into fields and residential area wild animals are forced to enter into these areas for food and water. Elephants or wild boar tramp the vegetation in farm land in need of nutritious food. Human-Elephant is more frequent in south Asia and in Africa. This leads to conflict between humans and animals and in some cases endangerment of animal species. In recent times the number of these kinds of these kinds of conflicts are increasing. People lose their crops and cattle and in some extreme cases they lose their life, so the field or zone is continuously monitored to prevent entry of animals are any other unwanted intrusion. The animal intrusion detection system uses a camera and a remote server to detect animals.

ESP32 is a cost-efficient camera module used to stream live footage to the server via wifi. Arduino IDE is used to upload the firmware to the module. The firmware contains the configuration details like picture quality. Object detection algorithm based on yolov5 is used for object detection. The algorithm is pre-trained with thousands of annotated images multiple times to achieve maximum accuracy. The camera module is placed in multiple places in the field to achieve maximum coverage. The captured live footage is sent to a cloud server using wifi and it is processed using pre-trained algorithm. The objects in the video are detected and classified based on their similarities. Based on the animal detected on the footage an alert is sent to the farmer.

II. RELATED WORKS

Priya Sharma et al., **[1]** introduced a system designed to safeguard crops from potential threats posed by animals. Their innovative solution relies on a combination of a Raspberry Pi and a PIR sensor. The PIR sensor serves to detect any motion made by animals, prompting the camera to activate and capture images. These captured images are subsequently transmitted to the Raspberry Pi for analysis and classification of the detected animal. Based on this classification, the system takes appropriate actions to protect the crops.

Prajna. P et. al. **[2]** have developed a field monitoring system. This system begins by utilizing a sensor to identify any encroachments around the field, followed by activating a camera to capture an image of the intruder. Subsequently, it employs image processing techniques to categorize the intruder, and based on this classification, it takes suitable measures to address the intrusion. Finally, the system sends a notification to the farm owner to keep them informed.

Sk. Almas Tabassum et. al. **[3]** introduced a system designed to protect crops from potential threats posed by animals such as cows, buffaloes, and elephants. These animals occasionally enter crop fields and can cause significant damage by trampling and consuming the crops. The system is designed to identify these animals and promptly send a photograph of the detected animal to the farmer. Furthermore, it emits a continuous buzzing sound as long as the animal remains within the camera's view, alerting the farmer to the presence of the intruder.

Howard et. al n their paper **[4],** the authors introduce a category of efficient models known as Mobile-Nets, tailored for applications in mobile and embedded vision. Mobile-Nets are constructed using a streamlined architecture that leverages depth wise separable convolutions, resulting in lightweight deep neural networks. The authors also introduce two straightforward global hyperparameters that effectively balance the trade-off between computational speed and model accuracy. These hyperparameters empower model developers to select an appropriately sized model that aligns with the specific constraints of their application. The paper includes an extensive set of experiments evaluating the trade-offs between resource usage and model accuracy. The results demonstrate Mobile-Nets' robust performance when compared to other widely-used models in the context of ImageNet classification. Additionally, the authors illustrate Mobile-Nets' effectiveness in various applications and use cases, spanning object detection, fine-grained classification, face attribute analysis, and large-scale geo-localization.

[5] Yiting Li presented this research paper with the aim of achieving real-time and highly accurate detection of surface defects using deep learning techniques. To accomplish this goal, the study adopted a Single Shot Multi-Box Detector (SSD) architecture, which was integrated into the MobileNet architecture at the bottom convolutional neural network (CNN) level, resulting in what is referred to as MobileNet SSD. To collect target images for the detection process, the system utilizes a combination of parallel cameras, digital cameras, and charge-coupled devices (CCD cameras). These cameras are employed to capture images of the objects of interest, extract relevant features, and establish mathematical models for analysis. MobileNet SSD primarily relies on paired cameras, digital cameras, depth cameras, and CCD cameras to achieve its objectives.

[6] In his research paper, Debojit Biswasa proposed the implementation of SSD (Single Shot Detection). The most labour-intensive task for any object detection algorithm is the creation of training datasets. In this work, approximately 500 objects were labelled across 450 images. Each labelled image generated an .xml file containing detailed information, including the location, height, and width of the labelled objects. Before presenting the labelled dataset to SSD, a mapping was created to establish the dataset's context. SSD was implemented using TensorFlow, serving as the base model for Mobile-Net-SSD. The Mobile-Net architecture was stored as Mobile-Net and was used for cross-training with TensorFlow SSD. The processes generated in this study can be effectively employed within TensorFlow for object detection.

[7] In her research paper, Anita Chaudhari proposed an object detection system that employs CNN (Convolutional Neural Network) technology. This system is designed to take images of various fruits and vegetables, categorizing them into distinct classes and providing nutritional information. When you capture an image of your plate of food, the application uses these images to perform comparisons. Subsequently, it generates a list of items that are most likely to be present in your meal. Instead of presenting a single choice with all the items on your plate combined, it lists each recognized item individually. For instance, if there are oranges on your plate, the object class detection component of the system identifies and reports various features of the oranges, such as their shape, colour, texture, and so on. This enables a more detailed and accurate assessment of the objects present in the image.

III. PROPOSED WORK

In this project ESP32 cam module is used. With the ESP32-S chip, the ESP32-CAM is a tiny camera module with wifi module. A microSD card slot is included in addition to the OV2640 camera and several GPIOs for connecting peripherals. This slot can be used to store images captured by the camera or files that will be provided to clients. The ESP32-CAM doesn't come with a USB connector, so you need an FTDI programmer to upload code through the U0R and U0T pins (serial pins).

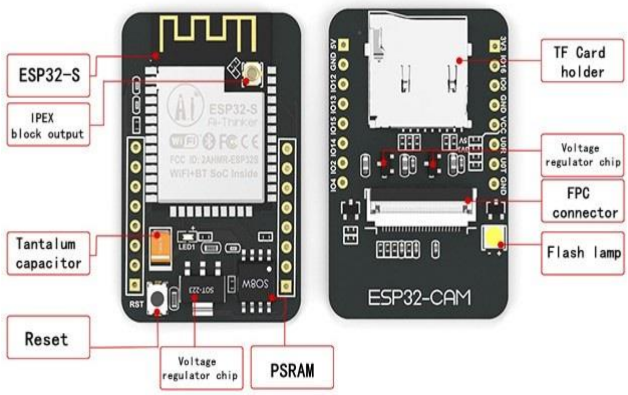


Fig 1: ESP32 Cam

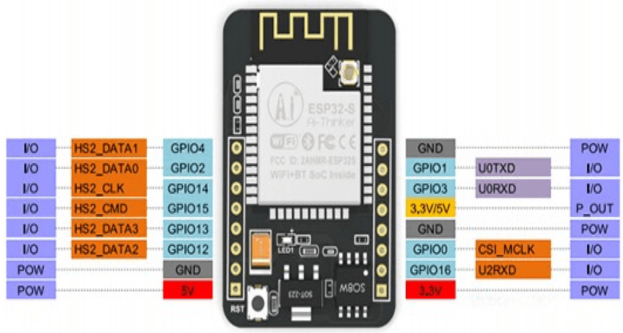


Fig 2: ESP32 pinout

FTDI USB to TTL serial converter modules are used for general serial applications. They are popularly used for communication to and from microcontroller development boards such as ESP-01s and Arduino micros, which do not have USB interfaces. It is used to program ESP32 module and upload firmware.

Arduino IDE is used to program ESP32 module. ESP32 cam library is imported for video streaming. The esp32cam library provides an object-oriented API to use OV2640 camera on ESP32 microcontroller. It is a wrapper of esp32-camera library. It is programmed to connect to a local wifi and send the video to a remote pc.

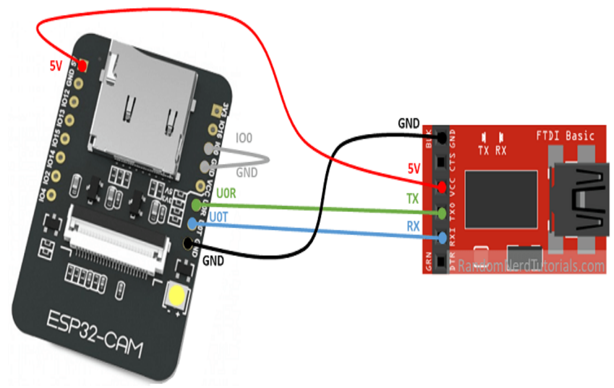


Fig .3: Circuit connections

Once the ESP32 cam is configured the live footage is sent to a remote server or PC. On the server yolov5 is used for object detection. A custom, annotated image dataset is vital for training the YOLOv5 object detector. It allows us to train the model on specific objects of interest, leading to a detector tailored to our requirements. Annotated animal image dataset is used for the model training.

YOLOv5 is a recent release of the YOLO family of models. YOLO was initially introduced as the first object detection model that combined bounding box prediction and object classification into a single end to end differentiable network. It was written and is maintained in a framework called darknet. YOLOv5 is the first of the YOLO models to be written in the PyTorch framework and it is much more light weight and easy to use. That said, YOLOv5 did not make major architectural changes to the network in YOLOv4 and does not outperform YOLOv4 on a common benchmark, the COCO dataset. Yolov5 seek to identify the presence of relevant objects in images and classify those objects into relevant classes. We need to train the yolov5 to recognize each one of those objects and classify them correctly. This model will separate the bounding box regression from object classifications in different areas of a connected network.

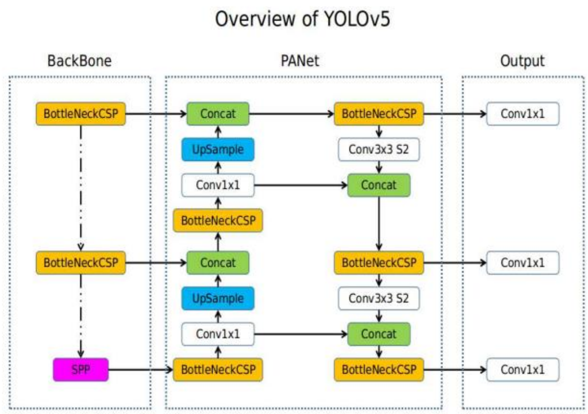


Fig 4: Overview of YOLOv5

In order to get your object detector off the ground, you need to first collect training images. You want to think carefully about the task you are trying to achieve and think ahead of time about the aspects of the task your model may find difficult. I recommend narrowing the domain that your model must handle as much as possible to improve your final model's accuracy.

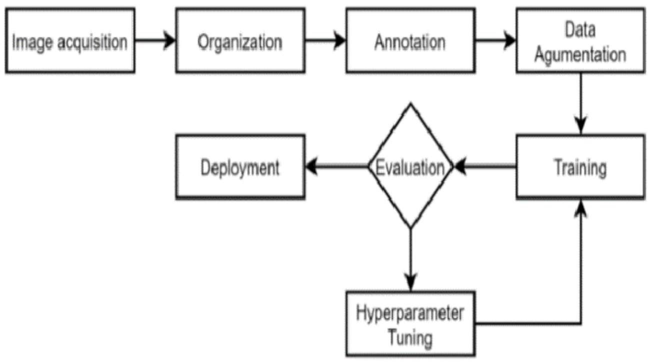


Fig 5: Object detection block diagram

The model is trained with thousands of annotated animal images for multiple epochs until desired accuracy is reached. The accuracy of the model is calculated by dividing the image dataset into training and evaluation set. 80 percentage of images are used for training and the remaining 20% is used for evaluation. When animal is detected the class of the animal is identified. If the particular animal is dangerous a message is sent to the person using python messages library.

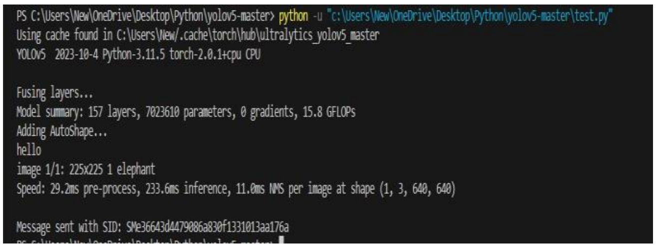
The final python script is uploaded to a cloud server or executed in a remote PC for live animal detection. It is cheaper than surrounding the property with electrical fence and other protection methods. It can be implemented in forests to find and track endangered species and animal pouching can also be prevented.

ISSN No:-2456-2165

IV. RESULT AND DISCUSSION

Employing the ESP32-CAM in tandem with OpenCV for animal intrusion detection represents an inventive and practical strategy for safeguarding areas susceptible to wildlife or animal presence. The integration of the ESP32-CAM's versatile capabilities, encompassing built-in Wi-Fi and camera functionality, with the robust computer vision library, OpenCV, empowers the development of an advanced real-time system for identifying and alerting to the presence of animals. This technology holds significant potential across various domains, including wildlife conservation, agricultural security, and residential protection, providing an effective and automated means of detecting and responding to animal intrusions. This project will explore the hardware setup, software implementation, and potential application scenarios of this solution for animal intrusion detection.

Fig 6: Result


```
PS C:\Users\New\OneDrive\Desktop\Python\yolov5-master> python -u "c:\Users\New\OneDrive\Desktop\Python\yolov5-master\test.py"
Using cache found in C:\Users\New/.cache\torch\hub\ultralytics_yolov5_master
YOLOv5  2023-10-4 Python-3.11.5 torch-2.0.1+cpu CPU

Fusing layers...
Model summary: 157 layers, 7023610 parameters, 0 gradients, 15.8 GFLOPs
Adding AutoShape...
hello
image 1/1: 225x225 1 elephant
Speed: 29.2ms pre-process, 233.6ms inference, 11.0ms NMS per image at shape (1, 3, 640, 640)

Message sent with SID: SMe36643d4479086a830f1331013aa176a
```

Fig 7: Alert

V. CONCLUSION

In summary, the application of the ESP32-CAM module for animal intrusion detection stands as a versatile and valuable solution across diverse sectors and industries. It holds the promise of bolstering wildlife preservation initiatives, safeguarding agricultural assets such as crops and livestock, fortifying property security, averting infrastructure damage, and facilitating essential environmental research.

While the ESP32-CAM module offers numerous advantages, such as cost-effectiveness and scalability, it is essential to recognize and address its inherent limitations. These limitations encompass factors such as detection range limitations, sensitivity to varying weather and lighting conditions, and the potential for false positives. Diligent planning, customization, and optimization strategies are imperative to counterbalance these constraints and ensure the system's reliability and efficacy. In the bigger picture, the utilization of the ESP32-CAM module for animal intrusion detection represents a promising strategy for mitigating the repercussions of animal intrusions on various facets of human existence and the natural world. By harnessing this technology, we can enhance the safeguarding of invaluable assets, reinforce safety measures, and make meaningful contributions to the preservation of wildlife and ecosystems.

REFERENCES

[1]. Priya Sharma, Sirisha C K, Soumya Gururaj, and Padmavathi C, “Neural Network Based Image Classification for Animal Intrusion Detection System”, IJPRSE, vol. 1, no. 4, pp. 1–7, Jul. 2020.
[2]. Prajna. P, Soujanya B.S, Mrs. Divya, “IoT-based Wild Animal Intrusion Detection System”, International Journal of Engineering Research & Technology (IJERT) ICRTT – 2018 (Volume 06 – Issue 15).
[3]. Sk. Almas Tabassum, B. Sri Vaishnavi, Dr K. S. Sagar Reddy, “SMART CROP PROTECTION WITH IMAGE CAPTURE OVER IOT”, International Journal of Research in Engineering, IT and Social Science, ISSN 2250-0588, Impact Factor: 6.565, Volume 09, Special Issue 2, May 2019.
[4]. Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications."
[5]. A. Yiting Li, Haisong Huang, Qingsheng Xie and Liguo Yao, “Research on a Surface Defect Detection Algorithm supported Mobile-Net SSD” IEEE Transactions on Pattern Analysis, vol. 34, pp. 743–761, Sep 2018.
[6]. B. Debojit Biswasa, Hongbo Sua, Chengyi Wangb, Aleksandar Stevanovica, Weimin Wangc, “An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD”, pp. 3354–3361, Dec. 2018.
[7]. C. Anita Chaudhari, Shraddha More, Sushil Khane, Hemali Mane, Pravin, “Object Detection using Convolutional Neural Network within the Application of Supplementary Nutrition Value of Fruits,” International Journal of Innovative Technology and Explore Engineering, vol. 8, pp. 2278-3075, Sep. 2019.