Prethveraj, 2021, 9:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Animal Pathway Alert System to Prevent Wildlife-Vehicle collision

Prethveraj, Lecturer Umayal, Senior Lecturer Saravana Selvan
Department of Engineering & Computer Technology,
AIMST University,
Bedong, Malaysia-08100,
prethveraj.m.s@gmail.com, umayalsaravana@gmail.com, dsselvan@gmail.com

Abstract- The animal casualties because of road accidents are increasing rapidly. Currently, there is no wildlife pathway alert system commercially available with the features of detecting, identifying, and alerting the driver to ensure the safety of both wildlife animals and humans. In this paper, an IoT concept-based animal pathway alert system is proposed to reduce the chances of animal-vehicle collision. Prototyping is performed by using the NVIDIA Jetson Nano processor. A real-time computer vision-based AI camera is connected with Jetson Nano to capture the movement of animals. You Only Look Once (YOLO), a convolutional neural network (CNN) algorithm, help process 45 frames per second and takes the entire image to recognize the animal in one evaluation. Once the movement of animals is detected, the proposed algorithm first identifies whether the activity is due to an animal or any other factors. If the activity detected is due to an animal, it will send alerts to the forest base station and the roadside traffic light system. Hence, the paper aims to save the animals from road accidents and save numerous lives from risk. This paper achieves object recognition accuracy of up to 98.6%.

Keywords- Wildlife-Vehicle Collision, IoT, Neural Network, Object recognition.

I. INTRODUCTION

Wildlife-vehicle collision kills millions of wild animals all around the world. A report in the United States says that 1 to 2 million wildlife-vehicle collisions occur annually [1]. From 2010 to 2016, 51 animal-related accidents were reported in the state of Texas [2].

According to their report, most are recorded between 5 and 8 in the morning and 5 to 10 in the evening. These accidents are expected during the months of October, November, and December.

Wildlife-vehicle collisions (WVC) are 64% of total reports, events involving domestic animals (like dogs and cattle) are 31%, and the remaining 5% of reports are unspecified. Most animal-vehicle collisions (AVCs) in the state happen at night in dimly lit areas, generally on rural roads with little traffic.

In many countries, AVCs involving ungulate species are serious, and forecasting incidents and costs on a regional and national scale are critical for effective accident prevention. However, this prediction could be underestimated because of the absence of systematic monitoring for animal road-kill events.

This study [3] established a relatively simple method for estimating and forecasting the costs of current and potential traffic incidents involving moose, roe deer, and wild boar in Sweden, based on the assumption that AVCs are calculated by traffic volume and ungulate population dynamics.

Such forecasts necessitate an understanding of the interdependency of population growth and pressures such as traffic and hunting. Since the occurrence of wildlife traffic accidents is proportional to the size of the wildlife population, an increase in any of these pressures would decrease potential populations, and therefore the incidence of wildlife traffic accidents. Landscape features, in addition to these pressures, influence the nature of these populations.

The study's [4] primary hypothesis was that traffic loads and population sizes affect traffic accidents. For each ungulate animal U, where U is O (moose), D (roe deer), or P (wild boar), the number of accidents at a specific point in time, depends on a linear combination of traffic load and ungulate population, which is written as: where is the 'catch ability' coefficient, which measures the change in traffic accidents from a marginal increase in.

These species are responsible for 97 percent of all wildlife-related vehicle collisions. Few studies have accessed wildlife mortality on roads in many countries, but none has intended to identify the factors promoting this phenomenon [5]-[8].

II. RELATED WORKS

This paper [3] has stated a simple and low-cost approach for automatic animal detection on highways to prevent animal-vehicle collision using computer vision techniques. A method for finding the distance of the animal in real-world units from the camera-mounted vehicle has been proposed. The system proposed has trained more than 2200 images consisting of positive and negatives images and tested on various video clips of animals on highways with varying vehicle speeds. For animal identification, they use a combination of HOG and a boosted cascade classifier.

The OpenCV program implements all of the image processing techniques. Once the animal is identified in the video, the next move is to determine the animal's distance from the testing vehicle and warn the driver so that he can apply the brakes or take some other appropriate action, shown as a message on the command prompt.

Three types of messages (indications) will be sent to the driver based on the distance of the animal from the camera-mounted vehicle: "animal quite near," "animal little far," and "quite far." This author purely proposed the work with software only. The coding gets the input from the camera, then sending the necessary alert information to the driver. We are still far from implementing effective corrective action to mitigate the road-wildlife conflict in the absence of this kind of study. Now wildlife has increased in extinction with reduced population sizes, which will lead to loss of eco-systemic functionality [9]-[11].

Fig 1. Wildlife-vehicle crossing in the forest road. [14]

There is a higher chance of animal-vehicle collision on any roads through the Forest. Therefore, there is a need for real-time monitoring and alert system in specific animal crossing pathways. Figure-1 shows the scenario of wildlife in the Forest crossing the road while the vehicles are moving.

Some researchers have used PIR sensors and ultrasonic sensors to implement these hotspots but still not yet in reality. A lack of networks in this area & accurate identification of species led to this system not ready for actual implementation [12]-[14].

The author [12] investigated the animal-vehicle collisions that occur around the world in this article. Using IoT, they have come up with a way to avoid animal-vehicle collisions. They have used motion sensors like the PIR sensor to track movement, IP cameras to identify animal movement, and the ANN artificial intelligence algorithm for object recognition.

The neural network has trained with animals' data sets and produces a result with 91% accuracy. The neural network uses the back propagation algorithm to train the datasets. When PIR sensors are satisfied with the circumstance, it sends a message to the MQTT broker, who then sends the message to the LED signage boards installed along the roads where animals are crossing. The MQTT broker cloud is used in this paper. The node is the hub for all client-server communication.

The main drawback of the research is when a user switches from one zone to the next, MQTT must be unsubscribed from the previous zone and subscribed to the new location. This paper [15] has described a

International Journal of Science, Engineering and Technology

An Open Access Journal

system with autonomously detected wildlife mammals using a low-cost UAV, a prediction model, computer vision, and thermal Imaging.

The system runs using a microcomputer (raspberry pi) with a detection algorithm using a low-cost thermal sensor and GPS module. The system has proved that it can detect an animal using a thermal sensor, and the GPS on the UAV can transfer data (image, GPS location) wirelessly to a GCS which can be analyzed.

In this paper [16], the authors wrote a low-cost alert system in real-time using IoT to monitor the wildlife from not entering the human-populated area. Here, the authors have proposed a method using PIR sensors for motion detection. An alternative IR camera can be used, but for simplicity, they use PIR sensors. The sensor tower also has a GPRS/3G module to connect to the control center. The Forest was nearer to the human-populated area so that they had a strong network.

It will make communication easier. After that, a snap has been taken and sent wirelessly to the control station. At the same time, an SMS will be sent to the person in charge. Later on, if wildlife is detected, a warning sign or noise can be activated from the control center. They calculated the location of each tower so that in case of emergency, the crew member can reach the concerned area quickly and do further detailing.

In this paper [17], the researchers used Raspberry Pi hardware for elephant detection to overcome these problems by incorporating working principles of PIR sensors, radio detection and ranging sensors, and tripwire systems, an animal detection method.

From this detection is predicted citizens are better prepared when these large animals enter the territorial dominion. Thus can prevent enormous human casualties. This tool testing results show that the PIR sensor can detect elephants with span distances between 1 - 8 meters with a reaching angle of 180 degrees.

In paper [18], Wildlife Monitoring using Raspberry Pi hardware along with machine learning techniques. Here the author proposed the YOLO method to train and test the predefined dataset to identify the animals in the zoo. Especially this model has trained to identify the bear movement, human appearance in the cage, and animal disappear from the cage.

The authors [19] compared boards in real-time object detection such as Raspberry Pi 3 B+ in combination with Intel Movidius Neural Computing Stick (NCS), Nvidia's Jetson Nano, and Jetson AGX Xavier. They concluded that Jetson Nano performs well with the FPS ratio and at a reasonable price. The papers [20] proposed the PIR sensor with a camera in the farm area to get the animal's input.

The camera-captured images are going to compare with the animal database. The index Image, image Set, and retrieve Image functions will compare the images and check the value match range from 0 to 1. If the value is 0, then the image is not matched. If it is 1, then the query image is the same as that of the stored image. Consequently, an SMS will be sent to the forest officials and the field owner as alert information.

III. SYSTEM OVERVIEW

To ensure and establish a system to reduce the road-kill events happening in forest areas. In this work, an IoT-based wildlife pathway alert prototype is proposed to reduce the chances of wildlife-vehicle collision. The overall system proposed in this work is displayed in Figure 2.

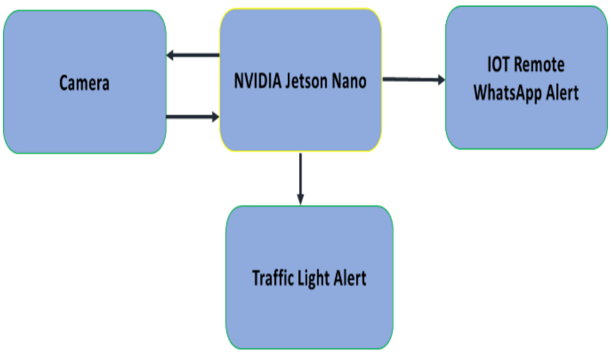


Fig 2. Overall proposed system block diagram.

NVIDIA Jetson Nano processor is presented in this work. The selection of Jetson Nano is not preplanned in this research. First of all, this research had started with raspberry pi, only that the real-time video capturing and activation of the processor based on the motion alert was worked finely. But after the animal identification, the processor needs to send the identified animal prediction accuracy value to the base station.

It needs to enable the Wi-Fi modem in the Raspberry Pi to connect to the forest rescue station. The problem happens in this step of work. Really for this process, the processor has to be connected to the base station of the forest rescue team and enable the traffic light to stop the vehicle movement.

The processor has fully engaged in identifying the animal with the help of the real-time camera. So throughput of the processor will reduce further, but still, it needs to transmit the real-time captured images with its accuracy value. So the processor speed does not do the remaining process because this prototype needs maximum ability to go ahead to the real-time implementation.

Under this research, a clear vision has got about that the importance of processor speed. So speed of the processor correlated with throughput. But through put is an essential factor in real-time implementation. Here there is no evidence to underestimate the accuracy of the algorithm processed by the processor.

The first responsibility of the processor to alert the camera to capture the motion pictures and the processor to convert them into multiple frames to undergo the process of species to identify them. Then the frames are transferred as input to the YOLO to implement CNN to convert into 13x13 frames to identify the objects and species.

So all processing had done with minimal usage of the processing time. But for the remaining part of the research only was affected by the overburden of the processor. So here is the understanding is Raspberry Pi is fit to do the first part of the work. It can be 100% well and good to identify the video-captured objects in real-time. The second part of the research was further affected by the processor's processing speed.

IV. NVIDIA JETSON NANO

The contribution of the NVIDIA Jetson Nano processor is playing the dominant role in touching the finishing line successfully on time. It is a portable and low-cost board capable of running any neural networks with an optimizer (TensorRT) for faster object detection [21]. Figure 3 shows the NVIDIA Jetson Nano hardware. An AI Camera is used to identifying the animals in real-time in the forest area to ensure the animals cross the road. The NVIDIA Jetson Nano released the first model, A02, in 2019. Later in 2020, a significant update Jetson family released a new Model B01 [22]. The significant difference between A02 and B01 is B01 supports 2 CSI (Camera Serial Interface), where A02 supports 1 CSI camera. There is no performance difference, but the B01 board is compatible with the Xavier NX.

Fig 3. NVIDIA Jetson Nano.

Jetson Nano has CUDA (Compute Unified Device Architecture) GPU, which can run all the neural networks quickly, and this board has been developed for Artificial intelligence. CUDA is a parallel computing platform and application programming interface (API) model created by NVIDIA [23]. A software layer gives direct access to the GPU's virtual instruction set and parallel computational elements to compute kernels.

TensorRT applications perform 40x quicker than CPU-only platforms during the inference. Using TensorRT, we can optimize neural network models trained in all major frameworks, calibrate with lower precision for higher accuracy, and finally deploy the data to hyper-scale data centers, embedded systems, or automotive product platforms.

Jetson Nano comes with a runtime optimizer TensorRT built using CUDA, which will minimize detection time with a higher FPS, which can solve detecting the animal in real-time. Because of that prediction of the future animal crossing, remote traffic light alert for the driver and send alert messages to the forest rescue team to avoid the accidents will be given by the project prototype.

Although it signals the traffic light, it also alerts the forest monitoring team to be ready for facing, monitoring, and controlling the scenario manually. Because of that, it will increase the time doubly the safety of animals and humans in the deep jungle.

International Journal of Science, Engineering and Technology

An Open Access Journal

V. WI-FI CONNECTION FOR SENDING THE DATA

The Nano module needs a Wi-Fi connection to transmit the data to the base station for this remote alert. Unfortunately, it doesn't have Wi-Fi integrated, so it needs to feature it manually.

The Edimax N150 a separate module having the features of a USB Wi-Fi / Bluetooth combination adapter. The Edimax 2-in-1 Wi-Fi and Bluetooth 4.0 Adapter EW-7611ULB is a nano-sized USB Wi-Fi adapter with Bluetooth 4.0 that supports Wi-Fi up to 150Mbps while allowing users to attach all or any of the newest Bluetooth devices like mobile phones, tablets, mice, keyboards, printers, and more.

Compatible with the latest Bluetooth 4.0 release with Bluetooth Smart Ready, this adapter delivers ultra-low power consumption with Bluetooth Low Energy (BLE) while shifting data or connecting devices.

First, it is required to attach the community peripherals to the Jetson Nano. The Wi-Fi adapter is a USB key, but it needs an Ethernet cable, NVIDIA Jetson Nano Developer Kit, and a 5V 4A power supply. Connect an adaptor to any open port of Jetson nano.

The board is getting power from a 5 V 4 a barrel jack power supply covered with the DLI Course Kit. The installed Linux OS at the board operates after the above hard-wired connections. It ensures that the Wi-Fi module becomes ON via typing the subsequent command: nmcli r Wi-Fi on [Enter].

It scans and lists all seen Wi-Fi networks by typing the following command: nmcli d Wi-Fi list [Enter]. It needs to list viable network availability comprising the current status regarding signal strength, data rate, channel, security, etc. Now that the Jetson Nano is attached wirelessly to the network, it's time to include it in the development.

VI. METHODOLOGY

This proposed system will be suitable for the jungle area. Animals are crossing the road area in the Forest at any instance. In this scenario, controlling the vehicle is essential because it is impossible to control the wildlife movements. The flowchart of this proposed system is shown below in Figure 4.

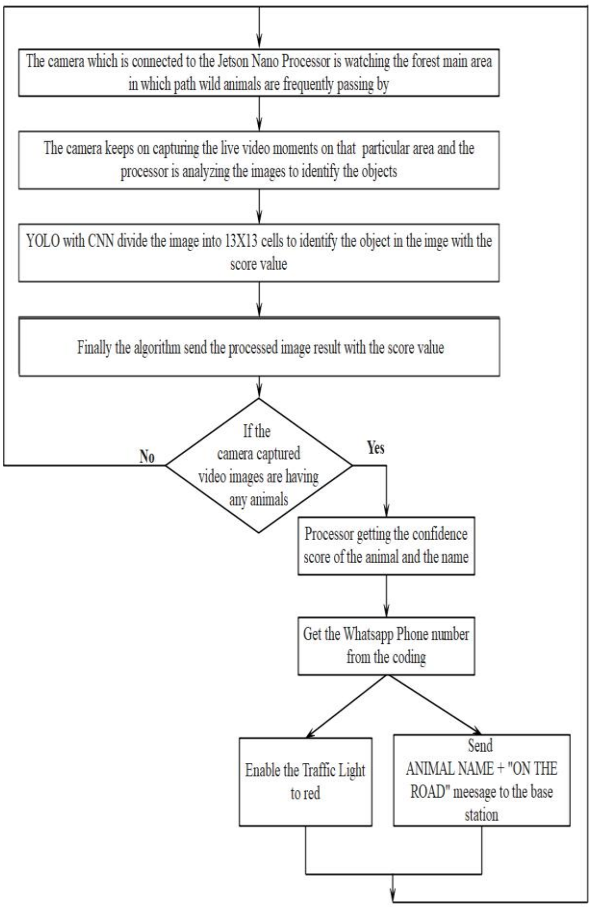


Fig 4. Flow chart of prototype model developed in this work.

First, the project will start monitoring the movement of the wildlife creatures. The Jetson Nano AI Camera can capture the object's moves with good clarity at that movement. This camera catches over 39 frames per second and can capture the real-time movement of animals.

It is suitable for cyber shots. During that instance, it can capture and also identified the objects and animals accurately. Second, these live stream captured videos convert into multiple frames of photos. Subsequently, it will send to the Jetson Nano processor since it has a Cuda GPU that can rush the entire neural network.

This board has been developed for artificial intelligence. So this processor will forward the captured images into the software for image processing.

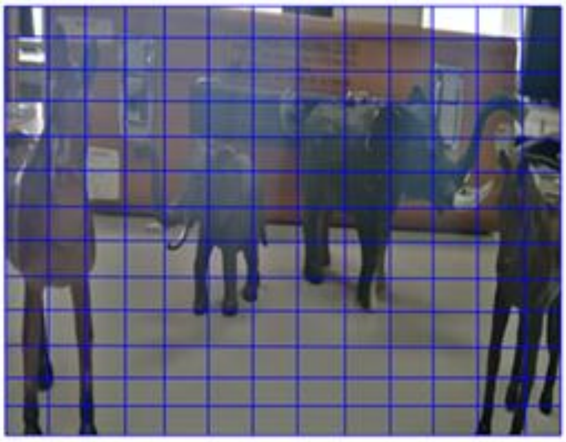


Fig 5. 13x13 cell division to identify the objects.

VII. YOLO LIBRARY

The python script is used to process the image, classify the objects, and identify objects or animals with their labels. For this process, some important artificial intelligence image and object identification library files are imported here. With the support of library files, the camera-captured frames of images are uploaded and compared into the database to identify the animals.

Here YOLO had taken responsibility to determine the objects [24]. Yolo takes an entirely unique approach. It is not a traditional classifier that is repurposed to be an object detector.

YOLO looks at the image just once (hence its name: You Only Look Once) but cleverly. YOLO divides up the captured animal image with background into a grid of 13 by 13 cells: each of these cells predicts five bounding boxes to detect the animal from the image frame shown in Figure 5.

Fig 6. Object identification without a score.

A bounding box describes the rectangle shade that encloses an object that shows in Figure 6. YOLO also outputs a confidence score that tells us how certain the predicted bounding box holds some animals. This score says nothing about what kind of animal is in the shaded box, just if the shape of the box is any good. The predicted bounding boxes may look like the following (the higher the confidence score, the fatter the box is drawn): for each bounding box, the cell also predicts a class.

This works just like a classifier: it gives a probability distribution over all the classes. From the 845 total bounding boxes, the above figures gave the best results. But note that even though there were 845 different predictions, they were all made simultaneously the neural network just ran once [25]. And that's why YOLO is so powerful and fast.

Fig 7. Object identification with the score.

This neural network only uses standard layer types: convolution with a 3x3 kernel and max-pooling with a 2x2 kernel. The very last convolutional layer has a 1x1 kernel and exists to reduce the shape 13x13x125. This 13x13 should look familiar: the size of the grid that the image gets divided. These 125 numbers are the channel for every 5 bounding boxes.

A bounding box is described by 25 data elements like x,y, width, height for the bounding box's rectangle, the confidence score, and the probability distribution over the classes. Using YOLO is simple: you give it an input image (resized to 416x416 pixels), it goes through the convolutional network in a single pass, and comes put the other end as a 13x13x125 tensor describing the bounding boxes for the grid cells. We need to compute the final scores for the bounding boxes and throw away the ones coring lower than 30%. The confidence score for the shaded box and the class prediction are combined into one final score that tells us the probability that this bounding box contains a specific type of object.

Since there are 13x13 = 169 grid cells and each cell predicts five bounding boxes, we end up with 845 bounding boxes in total. It turns out that most of these boxes will have very low confidence scores, so we only keep the boxes whose final score is 30% or more. The final prediction is then shown in Figure 7.

Fig 8. Real-time detection of an object by using a webcam.

Once the python script identified an animal equivalent to the list of animals in the script, it will signal the processor to enable the traffic light and send information to the base station. NVIDIA Jetson Nano data processor receives the signal from the python script and will decide on future action. If the camera taken images compared with the script is not equivalent, there is no animal movement.

It results from twist a few plants or trees are shaking. At that moment, camera taken shots will not be matching with the animal measure. So the python script will provide the information message to the processor that objects or animals are not matching. So processor will interpret the signal and not taking any action here.

The camera pictures and the scripting picture list are comparable in that there are a few creatures. The scripting picture is nothing but the proposed YOLO; a CNN algorithm developed using python language. Figure 8 shows the real-time detection of animals and object by using the webcam. The output window has revealed the real-time identification of the elephant and person from a live video shown to the web camera. The system has detected and predicted the percentage of assurance of the object.

Using this system, we can classify each species so this technique will only alert for specified species as preferred. This system can identify 91 objects which are pertained in Yolo. It will compare the pertaining models in the artificial intelligence algorithm with the camera images. If any matches are found, the algorithm next tries to find the animal name. For example, the captured image is an elephant, so the elephant's 45 frames of images will compare with the algorithm's pertaining models.

All the animals have unique features; with that unique knowledge, the captured image features are mapping to identify the animal. If the captured image feature is equivalent to any one animal's feature in the list, it will map with the label of the creature which the algorithm is having. If the algorithm found the creature with accuracy, it will provide the data to the Jetson Nano processor for decision making. Suppose the processor has decided to stop the traffic flow to that area because of an animal crossing the road. In that case, it will send the detected animal name to the forest monitoring base station. At the same time, it will alert the traffic light.

VIII. BASE STATION ALERT

To alert the base station to monitor the situation and take the counteraction to avoid unnecessary accidents, already coding has been written in python. Imported Jetson.inference, Jetson.utils and Jetson.GPIO libraries are used in the coding to establish the client and server model communication, and the processor and the required mobile device, had found the connection to transmit the data to the destination device.

These steps are mandatory to enable the communication between the Jetson Nano processor with its destination device. These steps must be completed during the task development period, after which we can fully implement the project in real-time. To send the text message using Jetson Nano, we must first add the WhatsApp number in the coding to go to the said WhatsApp number. If in the future, need to change the number, need to update the coding.

Here in this work, we use that animal often cross. The forest monitoring base station will determine this path selection past dataset. This prototype will be implemented in the entire forest area because some animals are not following any proper pathway to move around the Forest.

But sometimes, when the animals are crossing the road in the Forest will use the same path. So this particular prototype proposes for implementing proper and determined pathway crossing only. If it is coming for the appropriate real-time device to monitor the forest area, periodic monitoring devices are suggested.

The forest monitoring team will receive a message from the processor to alert the nearby squad. The team will be ready to manually monitor and manage the situation to avoid accidents to the animals or the people.

Usually, when the camera does not detect animals, the traffic light is green. But when the animals are detected, it has a high probability to proceed towards the road. So the better decision is to stop the vehicle movement for a while until the animals are entirely crossing. If the animals have been crossed the road, the red light will become green for the free flow of the vehicle.

IX. RESULTS & DISCUSSION

The proposed IoT based animal crossing pathway alert system has been implemented in a prototype model.

Figure 9 shows the working prototype model developed in this work. In that prototype model, the movement of animal toys is used across the road to detect.

The execution of this prototype and its results are discussed below in this section.

- **Step 1:** Switching on the Jetson Nano Window
- **Step 2:** Open the terminal to run the command.
- **Step 3:** Run Python 3 script to run the application.
- **Step 4:** Waiting for the application to start running.
- **Step 5:** The application is ready to run, toys of elephants and horses are placed in front of the camera for detection.

Fig 9. Working Prototype of Animal Crossing Pathway Alert System.

A bounding box with a prediction score has been obtained in real-time detection. An elephant has been detected using the animal crossing pathway alert system. A remote red light alert has been given for the driver to stop, and an IoT WhatsApp message "ELEPHANT ON THE ROAD" using a cloud API.

Figure 10 shows the result of the message which had been achieved about the animal crossing in WhatsApp.

Fig 10. Real-time alert on WhatsApp with the evidence of animal detected.

Table 1. Prediction score accuracy of detection of animals.

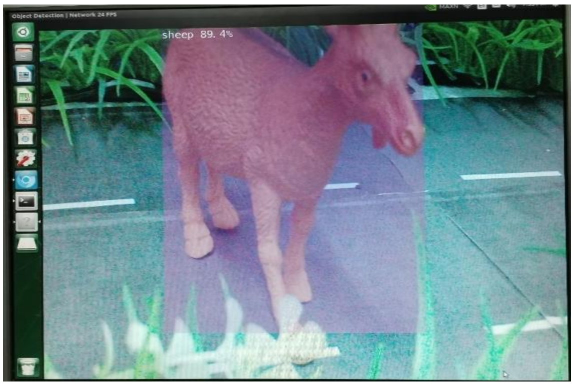
Animals	Prediction score of accuracy
Elephant	90.2%
Sheep	89.4%
Giraffe	93.3%
Horse	98.6%

Suppose the animals are going to pass through the road. In that case, the processor sends information to the data center to alert the team to handle the situation manually and turn the signal to red on the traffic light to stop the flow of vehicles in that area.

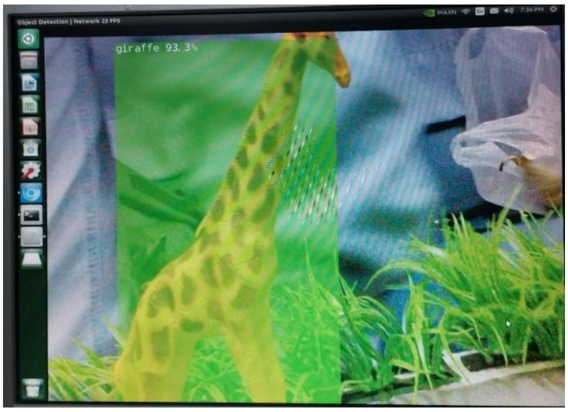
Otherwise, the processor will not send information to the base station and continue with the green color; it doesn't disturb the traffic flow. The system is connected to a cloud API for real-time alerts for the user to know whenever an animal passes by the road to handle or take precautions according to the complexity using a WhatsApp message.

Table 1 shows the prediction score accuracy of the detection of animals achieved by this proposed prototype system. Here a Horse has been identified with the highest prediction score accuracy of 98.6% with 45 FPS by using this animal crossing pathway alert system shown in Figure 11.

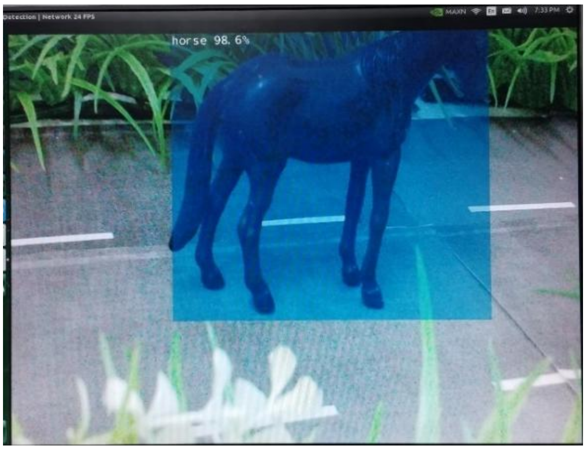
Other animals like Elephants, Sheep, Giraffes and their percentage of prediction score accuracy are mentioned. This system is highly faster and accurate in detecting animals and giving a remote alert to the user in real-time.



(a)



(b)



(c)

Fig 11. Detection of animals (a) Sheep (b) Giraffe (c) Horse with its prediction scores are displayed.

X. CONCLUSION

An AI-enabled IoT-based animal pathway alert prototype system has been designed in this work. Nvidia Jetson Nano processor with YOLO-based CNN algorithm is used to recognize the animal.

This system is built specifically for the frequent animal crossing pathway or hotspot detected in real-time. It achieves a high speed to give a remote traffic light alert for the driver and a WhatsApp message to the base station who can handle the situation depending on its complexity.

This system can detect, identify and alert with no human effort. This system can reduce the accidents happening on the highway and protect the animals from getting killed and humans from getting injured. An environmentally friendly IoT-based alert system has been designed and developed to reduce the chance of wildlife-vehicle accidents and real-time monitoring.

REFERENCES

[1] R. Pagany. Wildlife-vehicle collisions-Influencing factors, data collection and research methods, Biological Conservation, vol. 251, pp. 108-116, Jan 2020.

[2] C. C. St. Clair et al. Animal learning may contribute to problems and solutions for wild life–train collisions, Philosophical Transactions of the Royal Society B, vol. 374, no. 1781, p. 201-205, Sep 2019.

[3] M. Gren and A. Jägerbrand, "Calculating the costs of animal-vehicle accidents involving ungulate in Sweden," Transp. Res. Part Transp. Environ., vol. 70, pp. 112–122, 2019

[4] J. W. Gagnon, N. L. Dodd, S. C. Sprague, K. S. Ogren, C. D. Loberger, and R. E. Schweinsburg. Animal-activated highway crosswalk: long-term impact on elk-vehicle collisions, vehicle speeds, and motorist braking response, Human Dimensions of Wildlife, vol. 24, no. 2, pp. 132–147, Mar 2019.

[5] Y. Dorrazehi, M. Mehrjoo, and M. Kazeminia. A Warning System Design for Camel-Vehicle Collisions Mitigation, in 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE), 2020, pp. 203–208.

[6] G. S. Karthick, M. Sridhar, and P. B. Pankajavalli. Internet of Things in Animal Healthcare (IoTAH): Review of Recent Advancements in Architecture, Sensing Technologies and Real-Time Monitoring, SN Computer Science, vol. 1, no. 5, pp. 1–16, Sep 2020.

[7] S. U. Sharma and D. J. Shah, "A practical animal detection and collision avoidance system using computer vision technique," IEEE Access, vol. 5, pp. 347–358, 2016.

[8] M. Goswami, V. P. Prakash, and D. Goswami, "Animal-Vehicle Collision Mitigation Using Deep Learning in Driver Assistance Systems," in International Conference on Advances in Computing and Data Sciences, 2019, pp. 284–295

[9] K. Sugumar and A. S. Kumar. IoT Concept for Animal Detection Using ANN to Prevent Animal Vehicle Collision on Highways, International Journal of Science and Engineering Applications, vol. 8, pp. 492-497, Oct 2019.

[10] Collision on Highways, International Journal of Science and Engineering Applications, vol. 8, pp. 492-497, Oct 2019.

[11] M. Begum, D. A. Janeera, and A. K. AG. Internet of Things based Wild Animal Infringement Identification, Diversion and Alert System, in 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 801–805.

[12] N. M. Prasanna et al. Radar based animal detection system for road safety, Journal of Xi'an University of Architecture & Technology, vol. 7, pp. 1695-1700, Mar 2020.

[13] D. C. Wilkins, K. M. Kockelman, and N. Jiang. Animal-vehicle collisions in Texas: How to protect travelers and animals on roadways, Accident Analysis & Prevention, vol. 131, pp. 157–170, Oct 2019.

[14] W. Saad and A. Alsayyari. Loose animal-vehicle accidents mitigation: Vision and challenges, in 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019, pp. 359–364.

[15] S. Ward, J. Hensler, B. Alsalam, and L. F. Gonzalez, "Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision," in 2016 IEEE Aerospace Conference, 2016, pp. 1–8.

[16] S. Sheela, K. R. Shivaram, U. Chaitra, P. Kshama, K. G. Sneha, and K. S. Supriya, "Low cost alert system for monitoring the wildlife from entering the human populated areas using IOT devices," Int. J. Innov. Res. Sci. Eng. Technol., vol. 5, no. 10, pp. 128–132, 2016.

[17] E. Maulana, C. S. Nugroho, and A. B. Dianisma. Animal Presence Detection for Elephants and Extruding Method Based on Bee Frequency, in 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), 2018, pp. 119–122.

[18] Y. Zhang, H. Peng, and P. Hu. Cs341 final report: Towards real-time detection and camera triggering, Technical Report., Jan 2017.

[19] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge. Real-time apple detection system using embedded systems with hardware accelerators: an edge AI application, IEEE Access, vol. 8, pp. 9102–9114, Jan 2020.

[20] "Safer Crossings," The Nature Conservancy. https://www.nature.org/en-us/about-us/where-we-work/united-states/wyoming/stories-in-wyoming/wildlife-collisions/.

[21] H.-B. Yoo, M.-S. Park, and S.-H. Kim. Real Time Face detection Method Using TensorRT and SSD, KIPS Transactions on Software and Data Engineering, vol. 9, no. 10, pp. 323–328, 2020.

[22] S. Cass. Nvidia makes it easy to embed AI: The Jetson nano packs a lot of machine-learning power into DIY projects-[Hands on], IEEE Spectrum, vol. 57, no. 7, pp. 14–16, 2020.

[23] A. Süzen, B. Duman, and B. Şen. Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN, in 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1–5.

International Journal of Science, Engineering and Technology

An Open Access Journal

[24] D. Sato, A. J. Zanella, and E. X. Costa, "Computational classification of animals for a highway detection system," Braz. J. Vet. Res. Anim. Sci., vol. 58, pp. e174951–e174951, 2021.

[25] B. K. Reddy, S. Bano, G. G. Reddy, R. Kommineni, and P. Y. Reddy, "Convolutional Network based Animal Recognition using YOLO and Darknet," in 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1198–1203.