iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50113

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Animal Detection for Vehicle

Nikita Mate[1], Pragati Bhat[2], Gauri Mahalle[3], Shrutika Konde[4], Sheetal Arvikar[5]
Department of Artificial Intelligence, G. H. Raisoni College of Engineering

Abstract: ***A severe challenge with which all established countries are now dealing seems to be the deaths and harm resulting from traffic accidents. Animal-vehicle collisions are an increasing concern for transportation organizations worldwide since they result in thousands of deaths each year for both humans and animals. As more roads are developed, the places where animals live are decreasing, resulting in more collisions between vehicles and animals. The human and animal deaths and injuries, as well as the material expenses of these accidents, indicate the necessity of a solution for this issue. Deep learning algorithms to prevent animal collisions will be developed using data. The method may be enhanced by adding additional characteristics necessary to boost productivity on the datasets. The recommended strategy has the ability to improve public safety by reducing or avoiding animal or human incidents.***
Keywords: YOLO, YOLOv2, YOLOv3, YOLOv4

I. INTRODUCTION

Obstacle avoidance is one of the key features of safety systems, an essential part of self-driving cars. When operating in the real world, self-driving cars use a set of sensors and processes to find, check, and obstacle avoidance. Animals have been observed roaming freely on roads in countries such as India, resulting in unexpected accidents. As a result, having a reliable system for detecting animals on roads is extremely crucial. Active sensors including Lidars, Lasers, and Radar are used in the most popular approach to obstacle detection. Their primary benefit is that they can use limited computational resources to directly estimate distance. These active sensors do, however, offer a number of drawbacks, such as limited scanning rate and low positioning accuracy. Furthermore, visual data is important in a variety of application fields, such as object identification, recognition of traffic signs, and route detection. However, due to the various forms within the classes, designing and using optical sensors for obstacle detection is quite difficult. Ecological factors greatly affect obtaining incredibly high images, if not even more.

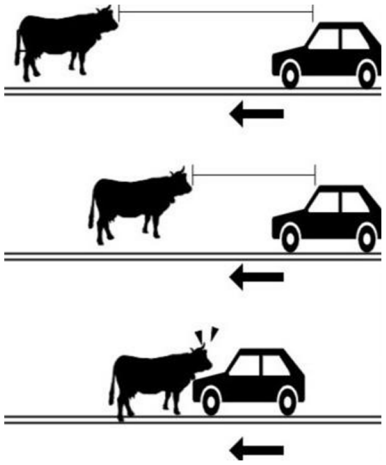


Fig. 1: Animal Collision

Deep learning has recently shown a lot of promise in the area of object identification and recognition. Convolutional Neural Networks (CNN) are specialized for vision-based methods and have excellent GPU acceleration capabilities in real-time applications. Originally designed for 3D modeling and visualization, GPUs are increasingly used to solve common image processing issues and provide considerable speed and performance gains over CPU-only solutions. A visual effects sequence's duration determines what is considered a video. FPS for picture resolution, or frames per second Therefore, the object detection methods used for this need to be reliable. adequate speed to handle numerous detections per second Real-time detection has been made possible in recent years by the development of object detection frameworks including YOLO (Bochkovskiy et al., 2020), Mask R-CNN (He et al., 2017), and Single Action Multibox Detection (Liu et al., 2016).

II. LITERATURE SURVEY

A Real AdaBoost algorithm is introduced in [4] with a real time classification example.With the help of Lidar 3D point clouds various feature objects of the road are analyzedMore than 90[9] presents a comparison study on object detection in autonomous vehicle situations utilizing deep convolutional neural networks, and lidar 3D aids in this (CNN). SSD Inception V2, Faster R-CNN Resnet 50, Faster R-CNN Resnet 101, and Faster R-CNN Inception V2 were the four well-known CNN models that were applied.After been pre-trained on the COCO dataset using transfer learning, these models underwent retraining on the new dataset. 517 images of 10 distinct items, including cars, bicycles, pedestrians, and seven traffic signals, were included in the new dataset, which was built using the GRAZ-01 and GRAZ-02 datasets. They employed the Q-learning technique to record and update the Q-values in a table for various moves, which the autonomous car will use to determine how and where to travel..The authors of [10] took on the challenge of assessing the effectiveness of semantic segmentation algorithms across a range of autonomous vehicle operating situations, including rain and light, and achieved an accuracy of 85.1Because even a minor change in the environment could have a big impact on how well and accurately the segmentation model classifies things, ultimately having disastrous effects. To tackle this challenging issue, they developed a pipeline that made use of a Lidar sensor to assess how well a particular model's semantic segmentation worked in various real-world scenarios.An encoder-decoder-based deep CNN model was proposed by the authors of [11] for the semantic segmentation of autonomous vehicle configurations. The VGG16 model is the foundation of the proposed model architecture. To maintain context and spatial information, they used residual training by carrying out segment addition and quick access connection.A comparison of their model's performance against well-known connections such as ENet and SegNet was included in the experiments, showing that it outperforms the existing both of them. The tests were run on a GPU made by NVIDIA called Titan X. When it comes to detecting animals, a convolutional neural network performs better than other machine learning methods. The most incredible experience The hypothetical CNN generates a sum for animal recognition [13].The experimentally collected data show that the proposed CNN offers the highest detection accuracy for a larger proportion of input trained data (about 98%). The categorization outcomes should be better when the image is split into more panes. The complexity of computing, however, will rise. The primary objective is to evaluate how well the suggested CNN approach stacks up against the combined recognition accuracy of such PCA, LDA, LBPH, and SVM approaches. Each feature map goes through a different pooling procedure. In general, the number of convolutional steps corresponds to an increase in the complexity of the features (such as edges). Up until the algorithm can accurately detect items, the procedure is carried out in stages. In comparison to other methods, the You Only Look Once (YOLO) technique for object detection has a number of benefits [14]. While the approach of convolutional neural networks does not fully analyze the image, YOLO's algorithm accomplishes this by calculating the bounding box coordinates and class probabilities for each of these boxes. As a result, compared to other algorithms, it finds the image more quickly. Yolo is composed of 75 convolution operations that are linked together by hidden neurons [17]. A process called feature extraction divides an input image into different scales. It produces output by applying a 1x1 kernel to a feature map. These specifications are sent to the detector to provide bounding box information. Darknet-53 is the feature extraction for the Yolo V3 model. The initial iteration of the YOLO model consisted of only 19 layers. As a result, the network extends the number of layer upon layer and for the Yolo V3 model from 19 to 53 [18]. The Yolo V3 algorithm has 53 extra layers to help it recognise the image as input [19]. The Yolo V3 detector is made to work on multiple scales. Darknet is a C and CUDA-based open-source framework. It is quick and simple to install and supports calculations on both the CPU and GPU. Darknet is a C and CUDA-based open-source framework. It is quick and simple to install and supports calculations on both the CPU and GPU. YOLOv3 forecasts a bounding scoreboard for each bounding box using logistic regression. [15] Regression One border box is connected to each ground truth real object. For object classes that are aggregated and close to one another, Yolo's test results are below average. Only a pair of the grid's boxes are assumed, and only those two boxes pertain to a new class of objects, which is the cause of this poor performance. As even more object courses are added, Yolo becomes less accurate [20].

III. PROPOSED METHODOLOGY

Deep learning has already shown significant promise for preventing collisions in the area of object identification and recognition.The YOLO algorithm, which stands for "You Only Look Once,," can detect and identify various objects in a picture or gather data in real-time. When solving a regression issue, YOLO conducts object detection and then delivers the class probabilities for the found images.The YOLO technique employs convolutional neural networks to recognise objects in real-time (CNN). As implied by the name, the method only requires one forward propagation through the neural network to detect objects. This demonstrates that a single algorithm run was sufficient to predict the entire scene. There are various iterations of the YOLO algorithm, including YOLO V2, YOLO V3, YOLO V4, etc.

A. YOLO

The Yolo methodology is renowned for its accuracy and rapidity. It has been used in a number of applications to find people, animals, and street lights. This program scans for and identifies various objects in a picture (in real-time). The technique just requires one forward propagation over a neural network to identify objects. This suggests that the whole the picture is predicted in a single run of the proposed technique. Multiple class probabilities and bounding boxes are forecasted simultaneously using the CNN. The YOLO methodology is crucial because of its accuracy, speed, and ability to learn. This method accelerates detection speed since it can recognize things in real-time. YOLO is a forecast strategy that generates precise results with little noise in the background. With its excellent learning abilities, the YOLO algorithm can acquire object representations and use them for object detection.

The YOLO method enlists three techniques-

1) *Residual Blocks:* There are various-sized grids arranged throughout the image. Every grid cell is capable of detecting objects that show up within other grid cells. For instance, if an item center falls inside a certain bounding box, that cell will be responsible for identifying it. Each square of the grid has a S x S dimension.
2) *Bounding Box Regression:* In object detection methods, bounded rectangle regression is a popular approach for improving or forecasting localized boxes. An item in an image is highlighted by a grid cell, which is a boundary. Each grid cell within the picture has a width, a height, a class, a center, and so on.
3) *IOU:* Box intersection is determined using intersection over union (IOU), which is used in object identification. Yolo constructs an actual output box that completely encloses the elements using IOU. The accuracy ratings of the bounding boxes and their identification are the responsibility of each grid cell. The IOU is equal to one if the predicted and actual boundary boxes are precisely the same. With this method, boundary areas that are less or larger than the real box are removed.

With regard to YOLO, YOLOv2 included incremental modifications, including batch normalization, image quality, and anchor boxes. YOLOv3 made predictions at three distinct levels of granularity, increased interconnections between the core network layers, and added an object category score to gridding prediction in order to boost performance on smaller dimensions. Perhaps the backbone, the neck, and the head are the three components that make up only one object recognition network known as YOLO v4. The framework might be a convolutional neural network with pre-trained models. The ”extraction of features network,” which calculates visual characteristics from input photographs, is the foundation of the Yolo v4 system.

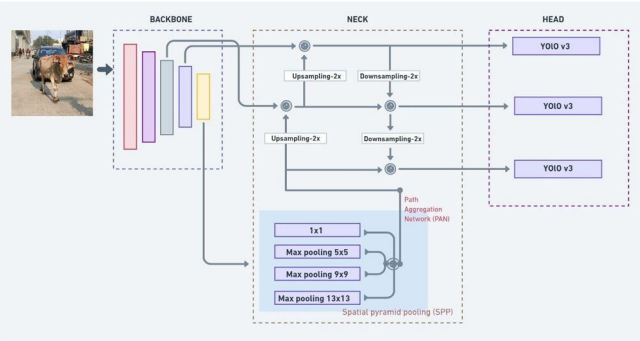


Fig. 2: YOLO v4 Architecture

The neck connects perhaps the head to something like the backbone. A spatial pyramid pooling (SPP) framework and route aggregating networks make up its two halves (PAN). The neck conveys inputs to the head by appending picture characteristics from various backbone network levels. The head processes the combined attributes and forecasts the data points, input image scores, and categorization scores. As detecting heads, the YOLO v4 network employs each classifier model, such as the YOLO v3. The YOLO v4 network extracts characteristics from input photos using CSPDarkNet-53 as the backbone. The Five Residual Blocks internal working model forms the backbone, and the outputs of their feature maps are pooled at the neck of the YOLO v4 network. To extract the most critical attributes, the SPP modules in the neck add the max-pooling outputs from the low-resolution feature map. The SPP module employs kernels with sizes of 1-by-1, 5-by-5, 9-by-9, and 13-by-13 for the max-pooling process. Set the stride parameter to 1. The perceptron’s backbone characteristics are enlarged by concatenating the image features, which also boosts the network’s capacity to distinguish microscopic objects. The high-resolution feature maps are coupled with the concatenated feature extraction from the SPP software using a PAN.

By integrating low-level and high-level features and employing upsampling and downsampling methods, the PAN creates bottom-up and top-down channels. Intersection over union (IoU) forecasts every anchor box's object class score. Anchor box offsets: Adjusts the location of the anchor box. Class probability: predicts the classifier that will be picked for each and every anchor box. A technique for estimating the animal's range in absolute real world units first from a camera-mounted vehicle is also provided. The suggested system is trained using photos from the Coco dataset, which comprises 20 classes: human, bicycle, car, motorcycle, bus, truck, traffic signal, bird, cat, dog, horse, sheep, cow, etc. The model is assessed using real-time visual representations of animals on roadways with different vehicle speeds. The video was acquired by the forward image sensor (camera) and displays a wandering animal amongst various stationary and non-stationary items, as well as the distance from the vehicles to the animal. In this instance, we are determining the distance between the cow and the car. should avoid utilizing pricey sensors like LIDAR, stereo cameras, etc. The technique uses the object's width rather than the image's height to determine outcomes. To effectively capture the reference image, the distance between the camera and the item, or animal, must be measured.

The bounding box occasionally does not fit the complete image, but for the most part, the algorithm provides the best bounding box. The model is constructed using openCV and the Numpy library. OpenCV is a collection of programming routines geared mostly at real-time computer vision. Cv.imread is used in the code to read the image that is located in the reference images folder. Cv.videocapture(0) is used to capture the real-time video from the webcam present. Cv.imshow is used to display the output.

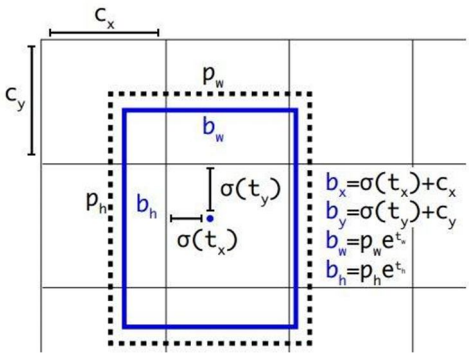


Fig. 3: Dimension priors and Location Prediction[25]

Fig. 4: Generating bounding box

A known constraint, together with the cow width and object detection constants, are specified. The colors for the bounding box are defined, along with the typeface that will be shown when the item is discovered. The process of creating a bounding box is shown in the diagram below. We anticipate overall height and width of the box as deviations from cluster centroids. We forecast the center coordinates of the box according to the position of the filter applications using a sigmoid function. The classes.txt file, which comprises all the object names, is opened in order to set up the openCV network. The cv.dnn DetectionModel is used to train the yolonet and then configure the input parameters. The object detector function is determined by which colors are defined for each class, identifying the class as well as creating the rectangle on the item. The distance finder feature estimates the distance between objects as well as the camera that detects them. cv.waitkey(1) is used in order to keep the screen from shutting. The screen may eventually be closed when the "q" key on the keyboard is pushed.

Fig. 5: A camera is mounted on a Car

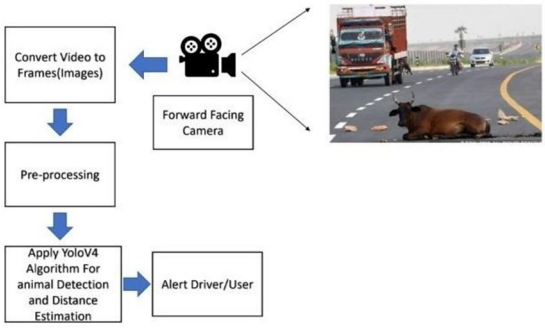


Fig. 6: To collect input

This clip is saved on the computer and then transformed into various frames. The picture is then enhanced using pre- processing techniques. Following preprocessing, we used YOLO v4, which displays the result of bounding boxes including real-time video.YOLO v4 converts image into grid and then finds the focal point of every object in the image and accordingly it shows class of every object in the image.

IV. RESULT

Our dataset's reference image and videos was manually gathered on location and for model we introduced CSPDarknet-53, which is superior to Darknet-53 because it requires only 66% of the parameters that version 3 did while producing better results that improve speed and accuracy.We conducted in-depth research and invested a great deal of time in testing the system under various road conditions. In our implemented model, it creates a marked rectangular box i.e., the bounding box. The animal (cow) is present in the video frame and our model correctly detects it and as soon as it detects it creates a bounding box accordingly. As shown in the figure 7 it has created bounding boxes. The model basically creates two bounding boxes, one which shows the class of that object which the system classifies based on the training. The other bounding box shows the distance from the camera to the object or animal. In the figure 8 above we can see that the distance is 2.92 meters as well as the cow class which the model has classified. Similar to the previous example in the case, the system mistakenly identified a dog as the animal even though a cow was visible in the video.This testing was done once from a stationary vehicle with 0 kmph speed while the animal(cow) was moving in the direction of the vehicle whereas rest of the data is trained from a moving vehicle with 39 kmph speed and the animal(cow) was moving in the direction of the vehicle.The system estimated near-around accurate distance from the vehicle and was gradually decreasing while the animal moved in the direction of the vehicle.

Fig. 7: Accurate animal detection and Distance Estimation

Fig. 8: Accurate animal detection and Distance Estimation

The model has also detected other objects such as people in the frame, along with the animals as it is trained on the COCO dataset. The dataset has different classes which helps in the classification of objects.This was accomplished by recording the video using a laptop camera, which has less HD resolution. Less than 30 seconds were needed to detect the object and estimate its distance once it entered the capture frame.

A. *Specification:*

A camera installed on the car which had a 30 frame-per-second frame rate was used to record videos.

1) *Processor:* HP Pavilion x360 Convertible with an Intel(R) Core(TM) i5-8265U CPU processor clocked at 1.80GHz and 8GB of RAM served as the test's computing platform.
2) *Video Resolution:* 2688×1944
3) *Photo Resolution:* 1600P
4) *Image Sensor:* used a 5 MP CMOS sensor and the lens was 6-layer glass, an infrared filter,140°wide angle,F1.8 aperture.

To evaluate the model's performance the following parameters were used: Sensitivity (True Positive Rate), Specificity (True Negative Rate) and Accuracy which are given as:-

$$\text{Sensitivity} = TP/(TP + FN) \quad (1)$$

$$\text{Specificity} = TN/(TN + FP) \quad (2)$$

$$\text{Accuracy} = (TN + TP)/(TN + TP + FN + FP) \quad (3)$$

True negative, true positive, false negative, and false positive are all abbreviated as TN, TP, FN, and FP, respectively.False Positive means that even though the animal is not present in the frame at that specific place (or in the video), it is still detected. False Negative (FN) means that even though the animal is visible in that specific frame of the video, it is not actually there. We used 1200 frames from our animal detection system to generate rectangular boxes that indicate when an animal is present in those frames during those locations. Therefore, in this instance, the true negative is 270 and the false positive is 90. In a similar vein, 140 out of 840 frames reveal no animal detection. i.e., no rectangular box, despite the fact that there are animals in that frame. Thus, the true positive is 700 and the false negative is 140. When the above-mentioned parameter values are substituted in equations (1), (2), and (3), we obtain sensitivity close to 83.5%, specificity close to 75%, and classifier accuracy near to 80.2%.

Figure 9 & 10 shows the camera which was mounted on the car bonnet at the center (equal distance from both the sides) The model was tested by recording and analyzing the videos that were captured by the camera, feeding those videos to the model, and spending a lot of time evaluating the system under various road conditions.

Similar to figure 11, figure 12 demonstrates a false positive case in which a dog is mistakenly identified by the system as a cow, despite the fact that the cow was not actually present in the frame at that location. Figure 12 shows a false negative situation in which the system shows the animal's absence (no box) even though the animal (a cow) is visible in the video.

Fig. 9: Side view

Fig. 10: Front view

Fig. 11: False Positive

Fig. 12: False Negative

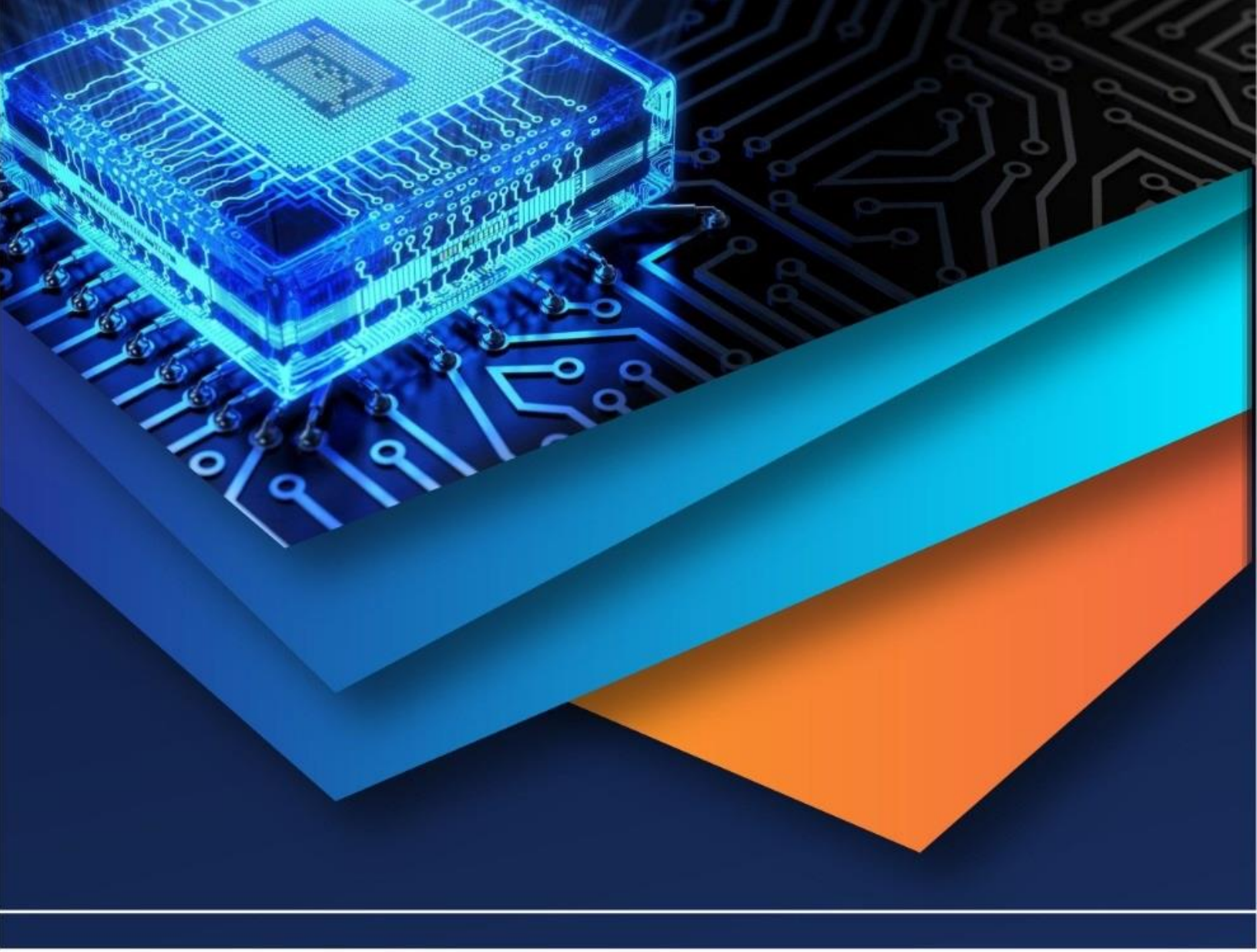
V. CONCLUSION

Currently, deaths and injuries from traffic accidents are a major issue for many countries. Road accidents are caused by a number of different factors, one of which is an animal colliding with a car on the road, especially in India. A prototype will be generated for the rural areas, which will guarantee that vehicles won't be influenced by the different colors of another ground or challenge and won't be disturbed by the illumination of the light sources, in addition to being capable of dealing with the multiple severe weather events in addition to seasonal effects in India. This model will help in minimizing the deaths of animals by vehicles. In order to locate moving animals in the video, a number of image processing techniques have been studied and reviewed in the past years.We observed that our proposed approach i.e. YOLOV4 algorithm performs well in identifying cows in video frames.This model will help in minimizing the deaths of animals by vehicles due to collision.

By developing this prototype, we have seen that the YOLOV4 model operates and recognises animals successfully. When we execute the YOLO V4 algorithm, it produces both right forecasts and delivers an exact distance. In the future we can gradually improve the model by adding an alert (buzzer) message that needs to be generated and should be sent to the driver or user.This suggested study has focussed on automated animal recognition with relation to Indian highways. Both the animal identification and range calculation is done by YOLOV4 technique. Even farther frameworks similar to YOLOX may be utilized for identification of many other species just after proper training and evaluation.

REFERENCES

[1] Gupta, S., Chand, D., Kavati, I. (2020). Computer Vision based Animal Collision Avoidance Framework for Autonomous Vehicles. Retrieved 3 August 2022, from https://arxiv.org/abs/2012.10878
[2] Computer Vision based Animal Collision Avoidance Framework for Autonomous Vehicles
[3] Sato, D., Zanella, A., Costa, E. (2021). Computational classification of animals for a highway detection system. Brazilian Journal Of Veterinary Research And Animal Science, 58, e174951. doi: 10.11606/issn.1678-4456.bjvras.2021.174951
[4] M. Yoshioka, N. Suganuma, K. Yoneda, and M. Aldibaja, "Real-time object classification for autonomous vehicle using LIDAR," in Proc. Int. Conf. Intell. Informat. Biomed. Sci. (ICIIBMS), Nov. 2017, pp. 210–211.
[5] J. Ciberlin, R. Grbic, N. Tesli c´, and M. Pilipovic´, "Object detection and object tracking in front of the vehicle using front view camera," in Proc. Zooming Innov. Consum. Technol. Conf. (ZINC), May 2019, pp. 27–32.
[6] K. Pranav and J. Manikandan, "Design and evaluation of a realtime pedestrian detection system for autonomous vehicles," in Proc. Zooming Innov. Consum. Technol. Conf. (ZINC), May 2020, pp. 155–159.
[7] H.-T. Tseng, C.-C. Hsieh, W.-T. Lin, and J.-T. Lin, "Deep reinforcement learning for collision avoidance of autonomous vehicle," in Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-Taiwan), Sep. 2020, pp. 2063–2068.
[8] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning," IEEE Trans. Veh. Technol., vol. 69, no. 6, pp. 5876–5888, Jun. 2020.
[9] G. Ozturk, R. Koker, O. ELDOgAN, and D. Karayel, "Recognition of vehicles, pedestrians and traffic signs using convolutional neural networks," in Proc. 4th Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT), Oct. 2020, pp. 1–8.
[10] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 5, pp. 1951–1963, May 2020.
[11] Y. G. Naresh, S. Little, and N. E. Oconnor, "A residual encoder decoder network for semantic segmentation in autonomous driving scenarios," in Proc. 26th Eur. Signal Process. Conf. (EUSIPCO), Sep. 2018, pp. 1052–1056.
[12] M. Feng, S. Hu, G. Lee, and M. Ang, "Towards precise vehicle-free point cloud mapping: An on-vehicle system with deep vehicle detection and tracking," in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Oct. 2018, pp. 1288–1293
[13] Trnovszky, Tibor, et al. "Animal recognition system based on convolutional neural network." Advances in Electrical and Electronic Engineering 15.3 (2017):.517-525.
[14] Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam "Real-Time Object Detection with Yolo" International Journal of Engineer- ing and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8, Issue-3S, February 2019, Crossref,https://www.ijeat.org/wp-content/uploads/papers/v8i3S/C11240283S19.pdf
[15] Reddy, B. Karthikeya, et al. "Convolutional Network based Animal Recognition using YOLO and Darknet." 2021 6th International Conference on Inventive Computation Technologies (ICICT)..IEEE,.2021.
[16] Manoharan, Samuel. "Embedded Imaging System Based Behaviour Analysis of Dairy Cow." Journal of Electronics 2, no. 02 (2020): 148-154.
[17] Kathuria "What's new in YOLO v3?. A review of the YOLO v3 Object" Towards data science , April 23rd 2018.
[18] Manogna Mantripradaga "Digging deep into YOLO v3 – A handson guide" Towards data science, August 16th.
[19] Joseph Redmon "Darknet-53 Explained" Cornell University, April 2018, https://paperswithcode.com/method/darknet-53.
[20] Jiang, Peiyuan, et al. "A Review of Yolo algorithm developments." Procedia Computer Science 199 (2022):.1066-1073.
[21] Tan, Mengyu, et al. "Animal Detection and Classification from Camera Trap Images Using Different Mainstream Object Detection Architectures." Animals 12.15.(2022):.1976.
[22] Gomez, Alexander, Salazar, Augusto, Vargas, Francisco, To- wards Automatic Wild Animal Monitoring: Identification of Animal Species in Camera-trap Images using Very Deep Convolutional Neural Networks, Ecological Informatics(2017), doi: 10.1016/j.ecoinf.2017.07.004
[23] Felzenszwalb, Pedro F., et al. "Object detection with discriminatively trained part-based models." IEEE transactions on pattern analysis and machine intelligence 32.9 (2010):.1627-1645.
[24] Gao, Cui, Qiang Cai, and Shaofeng Ming. "YOLOv4 object detection algorithm with efficient channel attention mechanism." 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)..IEEE,.2020.
[25] Redmon, Joseph and Ali Farhadi. "YOLOv3: An Incremental Improvement." ArXiv abs/1804.02767 (2018): n. pag.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY