Chapter 1
Automatic Recognition of Wild Animals for Road Accident Prevention Using Deep Learning with Yolov4

Papa Assane Diop, Amadou Dahirou Gueye, and Malal Deme

Abstract In this paper, we address the problem of wildlife recognition for road accident prevention, where a rate of 63.17% per year of road accidents is noted in Senegal. Given that the movement of animals is unpredictable in spite of road signs, this constitutes a handicap in preventing wild animals that may cause an accident. The solution proposed in this paper allows real-time detection of wild animals crossing roads, especially in non-built-up areas. It is based on computer vision using deep learning with the Yolov4 approach which allowed the categorisation of the three types of wild animals chosen in this paper: cows, donkeys and goats. The choice of these three types of animals is justified by the fact that most wildlife-related road accidents are caused by these types of animals. To achieve this, we first collected a set of images of the three types of animals. These collected images are sorted before the model is created. The evaluation of the model was carried out using test images and also videos taken on the Niague road, more precisely in the suburbs of the Senegalese capital, with accuracy rates of 94, 32%, 98.85 and 99.96%, respectively, for cows, goats and donkeys.

Keywords Obstacle · Road · Deep learning · Detection · Yolov4 · Recognition · Wild

1.1 Introduction

Developing countries are facing an increase in the number of vehicles in regional capitals. There is a high density of vehicle movement from region to region. In Senegal, road traffic accidents account for 63.17% of all accidents, which means that [1] of all accidents, which means that the majority of accidents are related to road insecurity. These traffic accidents occur outside built-up areas, i.e. between regions of the same country or between different countries. These accidents in nonurban

P. A. Diop (✉) · A. D. Gueye · M. Deme
TIC4DEV TEAM, Alioune DIOP University of Bambey, Bambey, Senegal
e-mail: papaassane.diop1@uadb.edu.sn

C.-C. Wang, A. Nallanathan (eds.), *6th International Conference on Signal Processing and Information Communications*, Signals and Communication Technology, https://doi.org/10.1007/978-3-031-43781-6_1

areas are most often caused by wildlife species. The authors in [2] conducted research to assess the impact of wildlife-vehicle collisions along the Dakar-Bamako corridor on animal populations in the Niokolo Koba National Park.

With all these problems related to roads, video surveillance is a necessary means of ensuring road safety. Video surveillance, commonly known as video protection, is made up of cameras and everything useful to record and exploit the images in order to detect abnormal events [3]. The main objective of processing a digital image is to extract information and improve its visual quality in order to make it more interpretable by a human analyst or an autonomous machine perception.

The use of video surveillance allows us to use computer vision which includes many object detection models. In our previous work [4], we proposed a lateral road obstacle detection model based on machine learning to contribute to road safety in areas outside built-up areas. Today, with the use of deep neural networks in computer vision, deep learning is taking over machine learning in terms of video surveillance. In this paper, we have chosen YOLOV4 for the detection of three types of animals in the wildlife which are cows, donkeys and goats. The rest of the paper will be structured as follows: in Sect. 1.2, we present related work on obstacle detection systems in the field of road video surveillance. In Sect. 1.3, we propose an approach based on the YOLOv4 detection model. In Sect. 1.4, we present the training and testing results of the model as well as the performance metrics. In Sect. 1.5, we conclude with a conclusion and perspectives.

1.2 Related works

In this section, we will look at work on animal detection and monitoring using deep learning, specifically yolo (You Only Look Once).

1.2.1 Object Detection Models Based on Deep Learning

The past decade, object detection models based on deep learning have gained great importance in the research field. In [5–7], there is a good overview of the state of the art of object detection models based on deep learning. For example, in [5], the author shows us that with the advancement of artificial intelligence, neural networks such as convolutional neural networks (CNNs) were often used in image processing. Later, CNN models face many problems in execution, performance, deployment, etc. In [8], another deep learning network, namely, Faster Regional Convolution Neural Network (Faster R-CNN) for object detection and tracking, is discussed. In the literature, we note other types of algorithms such as SSD [9] and F-CNN [10]. In this paper, we choose the deep learning detection algorithm Yolov4 which is much faster and more efficient in terms of detection [11]. These algorithms have often been used in video surveillance for object detection.

1.2.2 Roadside Video Surveillance of Wild Animals

In recent years, video surveillance has been the subject of much research using deep learning. Deep learning also gives rise to detection techniques such as YOLO. A presentation of the state of the art is available in [5, 9, 12].

For example, in [9], Haomin and He proposed a study on YOLO object detection algorithm for road scenes based on computer vision. They made a study on Yolo detection algorithms at the road level based on computer vision. The authors in [12] made an in-depth study on the progress of road object detection optimisation, which is an important part of detection and also the evaluation of detection models.

As pointed out by [13], the Yolo detection models withstand conditions such as night, rain and snow to provide fast and reliable detection. In [14], the authors presented a publication on the detection of wild animals in the forest and their use to monitor their movement. In the survey of research on detection models in road safety, we did not find any work on the Yolov4 detection model to warn the wildlife crossing roads. Thus, our paper is based on the Yolov4 approach to perform automatic wildlife detection applied on roads for accident prevention. In the following, we will present the detection approach based on YOLOV4.

1.3 Detection Approach Based on Yolov4

1.3.1 Architecture of Yolov4

In [15], the Yolov4 architecture is made up of different parts. The input comes first, and this is essentially what we have as our set of training images that will be passed to the network – they are processed in batches in parallel by the GPU. Then comes the backbone and the neck which does the feature extraction and aggregation. The sensing neck and sensing head can be referred to as an object detector assembly (Fig. 1.1).

YOLOv4 explores different backbones and data augmentation methods:

- Backbone network
- Neck
- PANet (Path Aggregation Network)
- Head

The head is the main function; it is to locate the selection frames and perform the classification.

The coordinates of the selection frame (*x*, *y*, height and width) and the scores are detected. Here, the *x* and *y* coordinates are the centre of the *b*-box expressed relative to the grid cell boundary. The width and height are predicted relative to the whole image.

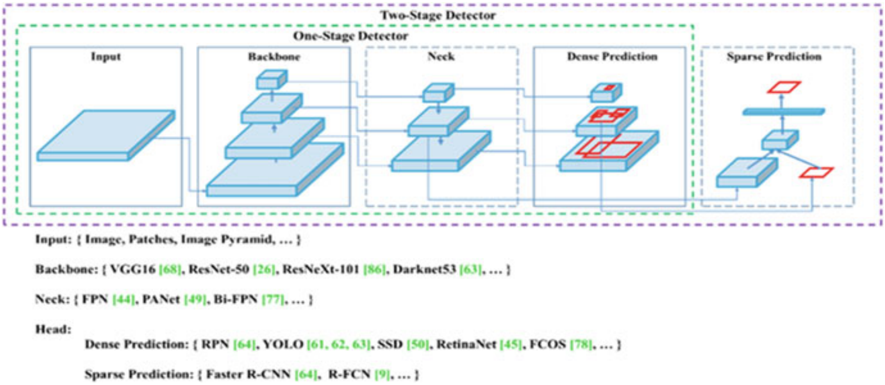

Two-Stage Detector
One-Stage Detector
Input
Backbone
Neck
Dense Prediction
Sparse Prediction
Input: { Image, Patches, Image Pyramid, … }
Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], … }
Neck: { FPN [44], PANet [49], Bi-FPN [77], … }
Head:
Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], … }
Sparse Prediction: { Faster R-CNN [64], R-FCN [9], … }

Fig. 1.1 Object detector

$$b_x = \sigma(t_x) + C_x \tag{1.1}$$

$$b_y = \sigma(t_y) + C_y \tag{1.2}$$

$$b_w = p_w e^{tw} \tag{1.3}$$

$$b_h = p_h e^{th} \tag{1.4}$$

1.3.2 Construction of Our Dataset

Our data represents a collection of images of three types of wild animal species: cows, goats and donkeys. These data were acquired through Google search sites, on a farm in Senegal, specifically in Niague, which raises cows. After the collection, we renamed the images using python code to make the renaming faster. Before renaming, we did a very essential step which is to remove the irrelevant images. After that, a problem arises, because the images acquired through the websites and the images taken through a camera on a farm were not the same size, so we have to do a resizing so that the size of all the images conforms to 671 × 480. Finally, we labelled the images. We used labelImg which is an open-source image annotation tool. We have 1000 images for each type of animal, making 3000 images in total (Fig. 1.2).

Fig. 1.2 Donkey, goat and cow

1.4 Experimentation and Validation

In this section, we will show the details of training our model to detect three (03) classes (donkey, goat and cow). Then, we present the performance measures of our model.

1.4.1 Setting Up the Experiments

Implementation Details For the training of the YOLO model, we based ourselves on the Darknet framework which contains all the necessary files [16]. The training phase of YOLO requires a lot of time, which is why we use the transfer learning method. This method consists of dividing the training phase between deep artificial neural networks, which results in savings in machine resources and computing time. We need to use a pre-trained model of YOLO to do the transfer learning. Before starting the training, we need to make some settings to adapt it to our model. These modifications concern the number of classes, the number of iterations and the number of filters to be used at the layer level of the convolutional neural networks.

Training Environment The training phase of a YOLO model is rather heavy, and if you have a lot of images, you will need to have a machine with very powerful resources (GPUs, RAM) for the model to learn in a suitable time frame. This is why we use Google Colab Pro to train our data [17].

Splitting the Dataset (Training/Test) We will just split our dataset (1000 images per class) to have a training dataset (80%) and a test dataset (20%). So we will have the following:

- A training data set (80%)
- A test data set (20%)

1.4.2 Performance Measures of Our Model

Several indicators can be used to measure the performance of an object detection model. Each one has its own specificities, and it is often necessary to use several of them to have a complete view of the performance of a model. Most of these indicators depend on the parameters true positive (TP), false positive (FP), false negative (FN) and true negative (TN) [18].

- TP: These are the correctly predicted positive values, which mean that the actual class value is yes and the predicted class value is also yes.
- TN: These are the correctly predicted negative values, which means that the actual class value is no and the predicted class value is also no.
- FP: When the actual class is no and the predicted class is yes.
- FN: When the actual class is yes but the predicted class is no.

Now we will define the performance measurement indicators for the case of a YOLO model [18].

Accuracy (*P*) The accuracy is the number of objects correctly assigned to class i relative to the total number of objects predicted to belong to class *i*.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1.5}$$

Recall (*R*) Recall is the number of objects correctly assigned to class i out of the total number of objects belonging to class i.

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1.6}$$

F_1-Score (F_1) Although useful, neither precision nor recall can fully evaluate a model. The F_1-Score provides a good assessment of the performance of our model. The F_1-Score subtly combines precision and recall to make a good assessment of a model's performance.

$$F_1 = 2 * \frac{P * R}{P + R} \tag{1.7}$$

Intersection on Union (IoU) It indicates the overlap of the coordinates of the predicted bounding box with the ground truth box. A higher IoU indicates that the coordinates of the predicted bounding box closely resemble the coordinates of the ground truth box.

$$\mathrm{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{1.8}$$

Mean Average Precision The mAP is calculated by finding the average precision (AP) for each class, then averaging over the total number of classes. Interestingly, the average precision (AP) is not the average of the precision (*P*). The term AP has evolved over time. To simplify, it can be said to be the area under the precision-recall curve. The mAP incorporates the trade-off between precision and recall and considers both false positives (FP) and false negatives (FN). This property makes mAP a suitable metric for most detection applications [19].

$$\mathrm{mAP} = \frac{1}{n}\sum_{i=0}^{n} \mathrm{AP}_i \tag{1.9}$$

Loss Function This is the sum of the errors made for each example in training sets. The main objective of a learning model is to minimise the value of the loss function with respect to the model parameters by modifying the values of the weight vector using different optimisation methods, such as back-propagation in neural networks. AP_i

1.4.3 Results and Analysis

After training, a graph is generated. The graph shows us the evolution of the average accuracy (mAP) of the model and the loss function as a function of the iterations (Fig. 1.3).

This graph shows that after 1000 iterations mAP = 72% then at 1200 iterations mAP = 98% then at 2500 iterations mAP = 99%, and in all remaining iterations, mAP is equal to about 98%. We also see that the loss function keeps decreasing until the end of the training to reach 0.466. This graph shows us the results in a global way, while we have three (3) classes. The following figure will give us in detail the results obtained (Fig. 1.4).

For donkeys, we have an accuracy of 99.96% with the number of true positives (TP) =156 and the number of false positives (FP) =30.

For the cows, we have an accuracy of 94.32% with TP =308 and FP =24.

For goats, we have an accuracy of 98.85% with TP =274 and FP =18.

Averaging the accuracies for our three classes, we have 97.71%. This shows that the detection model is acceptable.

1.4.4 Test

Testing on Images To do the detection on an image, we use a python script that takes images as input and makes a prediction with our model (Fig. 1.5).

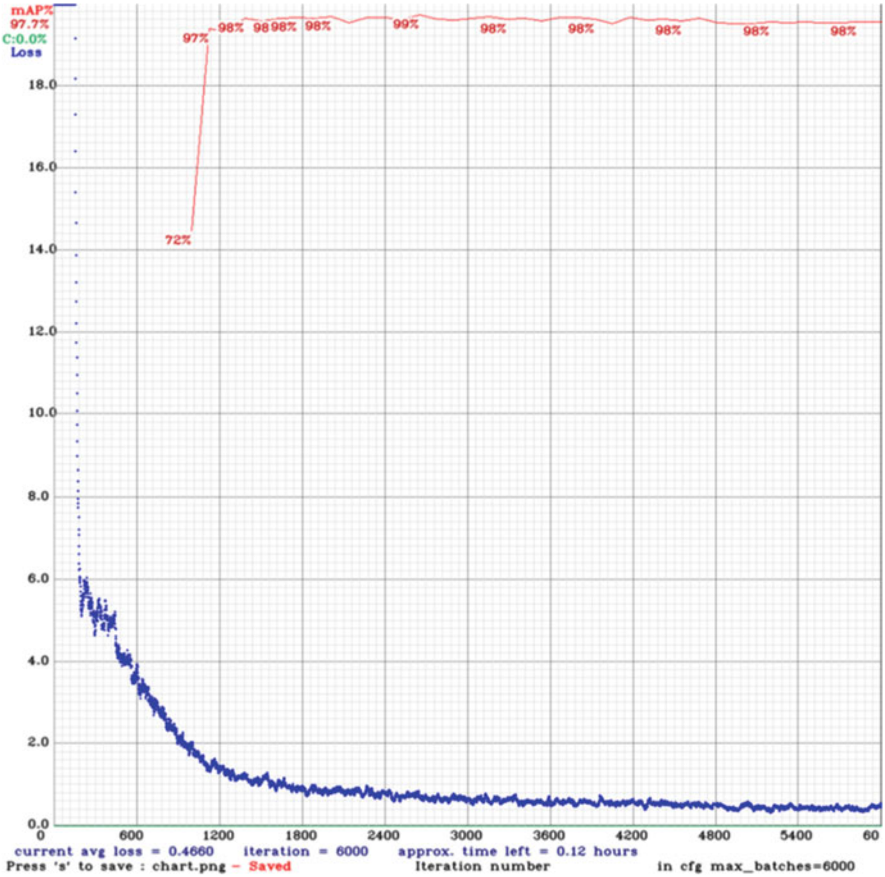
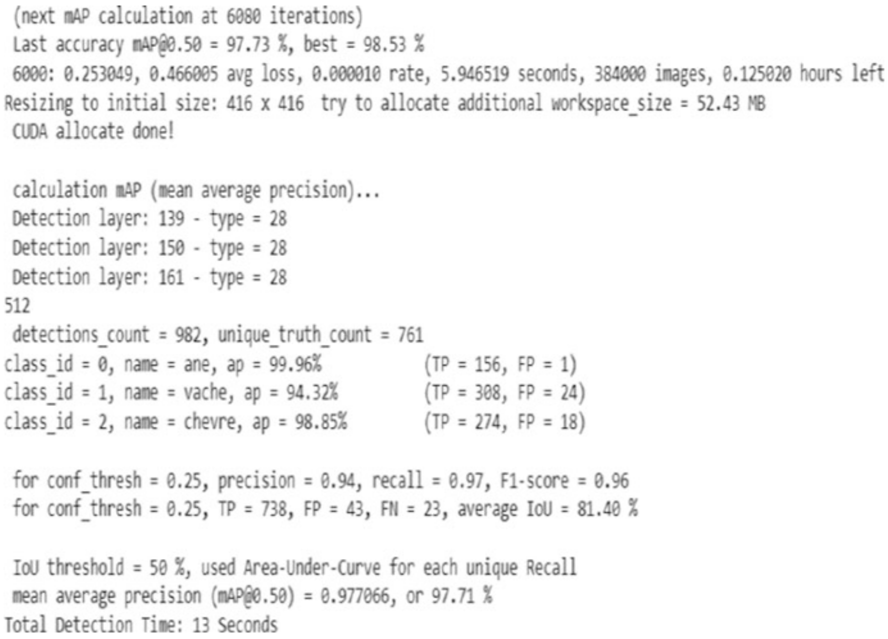


Fig. 1.3 Curve representing the evolution of the mAP and the loss function as a function of iterations

Testing on a Video To do the detection on a video, we use a python script that takes a video as input and makes a prediction with our model. This video was taken in real time on the Niague road located in Keur Massar, Senegal, as the cows are returning to the Niague farm after a day's walk (Fig. 1.6).

1.5 Conclusion and Outlook

Wild animals are increasingly unpredictable obstacles. Related work related to road obstacle detection has been presented to propose a Yolov4-based approach to detect wild animals such as cows, goats and donkeys crossing roads especially in nonurban areas. With this approach, a performance study of the model is done to validate our



```
 (next mAP calculation at 6080 iterations)
 Last accuracy mAP@0.50 = 97.73 %, best = 98.53 %
 6000: 0.253049, 0.466005 avg loss, 0.000010 rate, 5.946519 seconds, 384000 images, 0.125020 hours left
Resizing to initial size: 416 x 416  try to allocate additional workspace_size = 52.43 MB
 CUDA allocate done!

 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
512
 detections_count = 982, unique_truth_count = 761
class_id = 0, name = ane, ap = 99.96%           (TP = 156, FP = 1)
class_id = 1, name = vache, ap = 94.32%         (TP = 308, FP = 24)
class_id = 2, name = chevre, ap = 98.85%        (TP = 274, FP = 18)

 for conf_thresh = 0.25, precision = 0.94, recall = 0.97, F1-score = 0.96
 for conf_thresh = 0.25, TP = 738, FP = 43, FN = 23, average IoU = 81.40 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.977066, or 97.71 %
Total Detection Time: 13 Seconds
```

Fig. 1.4 Detailed results of the training

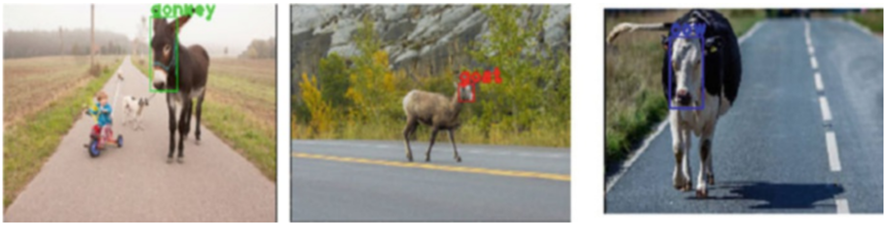


Fig. 1.5 Detection from images of the three categories of animals: donkey, cow and goat

work. In the perspective of the work, we propose an integration of several types of animals and also an evaluation of the distance of obstacles. We also plan to integrate IOT devices for the deployment of our model in a vehicle.

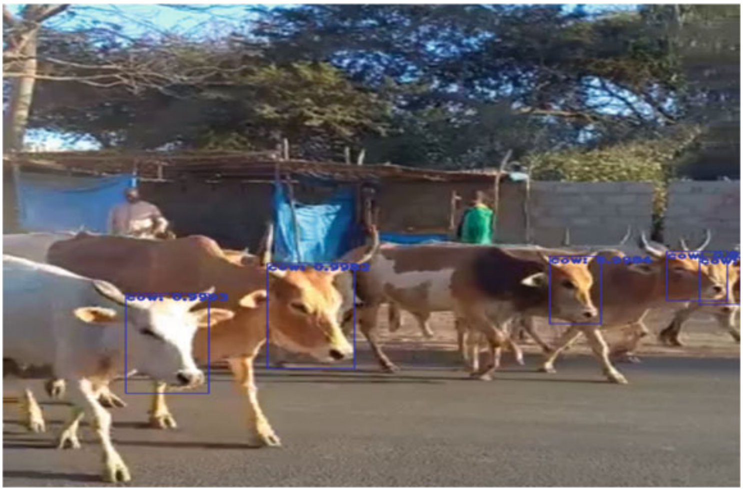


Fig. 1.6 Detection of cow on a video obtained on the road of "Niague" Senegal

References

1. So that roads no longer kill in Senegal, https://www.afro.who.int/fr/news/pour-que-les-routes-ne-tuent-plus-au-senegal. Accessed 19 Aug 2022
2. S.M. Sarr, M. Gueye, A. Aziz, Impacts des collisions avec les véhicules le long du corridor routier Dakar-Bamako sur les populations fauniques du Parc National du Niokolo Koba, au Sénégal. 12 (2022)
3. I. Global, Role of CCTV cameras: Public, privacy and protection, https://www.ifsecglobal.com/video-surveillance/role-cctv-cameras-public-privacy-protection/. Accessed 03 Oct 2022
4. P.A. Diop, A.D. Gueye, A.K. Diop, Detection of lateral road obstacles based on the haar cascade classification method in video surveillance, in *Computer and Communication Engineering. CCCE 2022*, Communications in Computer and Information Science, ed. by F. Neri, K.L. Du, V.K. Varadarajan, S.B. Angel-Antonio, Z. Jiang, vol. 1630, (Springer, Cham, 2022). https://doi.org/10.1007/978-3-031-17422-3_3
5. Z. Li, J. Wang, An improved algorithm for deep learning YOLO network based on Xilinx ZYNQ FPGA, in *2020 International Conference on Culture-oriented Science & Technology (ICCST)*, (2020), pp. 447–451. https://doi.org/10.1109/ICCST50977.2020.00092
6. P.S. Kumar, V.P. Sakthivel, M. Raju, P.D. Sathya, A comprehensive review on deep learning algorithms and its applications, in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, (2021), pp. 1378–1385. https://doi.org/10.1109/ICESC51422.2021.9532767
7. A.S. Abdullahi Madey, A. Yahyaoui, J. Rasheed, Object detection in video by detecting vehicles using machine learning and deep learning approaches, in *2021 International Conference on Forthcoming Networks and Sustainability in AIoT Era (FoNeS-AIoT)*, (2021), pp. 62–65. https://doi.org/10.1109/FoNeS-AIoT54873.2021.00023
8. K.B. Lee, H.S. Shin, An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels, in *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, (2019), pp. 7–11. https://doi.org/10.1109/Deep-ML.2019.00010

9. H. He, Yolo target detection algorithm in road scene based on computer vision, in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, (2022), pp. 1111–1114. https://doi.org/10.1109/IPEC54454.2022.9777571
10. M. Maity, S. Banerjee, S. Sinha Chaudhuri, Faster R-CNN and YOLO based vehicle detection: A survey, in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, (2021), pp. 1442–1447. https://doi.org/10.1109/ICCMC51019.2021.9418274
11. Papers with Code – COCO test-dev benchmark (object detection), https://paperswithcode.com/sota/object-detection-on-coco. Accessed 03 Oct 2022
12. YOLO, You only look once – Real time object detection, https://www.geeksforgeeks.org/yolo-you-only-look-once-real-time-object-detection/. Accessed 19 Aug 2022
13. Y. Wendi, X. Yahang, Z. Xiaoyu, L. Jiaming, W. Tianchen, Risk assessment method combining trajectory prediction and lateral obstacle monitoring, in *2021 9th International Conference on Traffic and Logistic Engineering (ICTLE)*, (2021), pp. 1–5. https://doi.org/10.1109/ICTLE53360.2021.9525701
14. C. Zhu, T.H. Li, G. Li, Towards automatic wild animal detection in low quality camera-trap images using two-channeled perceiving residual pyramid networks, in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, (2017), pp. 2860–2864. https://doi.org/10.1109/ICCVW.2017.337
15. YOLOv4 model architecture, https://iq.opengenus.org/yolov4-model-architecture/. Accessed 26 Sept 2022
16. Darknet, Open source neural networks in C, https://pjreddie.com/darknet/, Accessed 03 Oct 2022
17. I. Ali, A. Khan, M. Waleed, A Google colab based online platform for rapid estimation of real blur in single-image blind deblurring, in *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, (2020), pp. 1–6. https://doi.org/10.1109/ECAI50035.2020.9223244
18. E. Solutions, Accuracy, precision, recall & F1 score: Interpretation of performance measures, https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/. Accessed 03 Oct 2022
19. Mean average precision (mAP) explained: Everything you need to know, https://www.v7labs.com/blog/mean-average-precision, https://www.v7labs.com/blog/mean-average-precision. Accessed 03 Oct 2022