Direct Research Journal of Agriculture and Food Science
Vol. 10(11) Pp. 254-261, November 2022
ISSN 2354-4147
DOI: https://doi.org/10.26765/DRJAFS72902107
Article Number: DRJAFS72902107

Full Length Research Paper

Towards the Conservation of Endangered Mammals using Single-stage Deep Neural Network

Ejiyi, Chukwuebuka Joseph[1], Orakwue, Chiduzie O.[2], Qin, Zhen[1]*, Diokpo, Chidinma N.[3], Nnani, Ann O.[4], Ejiyi, Makuachukwu B.[5], Goshu, Hana L.[1], and Okpara, Chidimma P.[6]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, PRC.
[2]Department of Agricultural and Bio-Resources Engineering, College of Engineering Federal University of Agriculture Abeokuta, Ogun State. Nigeria.
[3]Department of Food Science Technology, Federal University of Technology Owerri, Imo State. Nigeria.
[4]College of Environmental Science and Engineering Hohai University, Nanjing, Jiangsu PRC.
[5]Pharmacy Department University of Nigeria Nsukka, Enugu State Nigeria.
[6]Department of Biochemistry, Federal University of Technology Owerri, Imo State. Nigeria.
*Corresponding Author E-mail: ginzhen@uestc.edu.cn

Received 1 October 2022; Accepted 29 October 2022; Published 1 November; 2022

ABSTRACT: For the purpose of conserving and preserving endangered mammals in their habitat, their real-time detection is very crucial. This will help curb their involvement in accidents as well as help in tracking them in case they stray which consequently will be useful for the preservation of life and property as most of the mammals are known to destroy crops among other damages that can be caused by them. We deployed a single-stage deep neural network – You Only Look Once (YOLO) version 4 for the detection of four classes of mammals in real-time which will help in monitoring the animals as well as tracking their activities not excluding their security and protection from poisoning and being preyed upon. A system of this magnitude can be employed to replace the manual monitoring of these mammals and the efforts put in place to track their position which can be achieved using a video in this model. The choice of YOLO has been because of its applicability for the detection of the 80 classes of objects in the MS COCO dataset and the high speed it has in performing this task. To be used for our purpose, we carefully choose the components of the network and modified and retrained the model to suit our desire which is to detect 4 classes of mammals that we have in our dataset after they were annotated. This research presents a custom dataset for the detection of this class of animals in real-time with a laudable mean Average precision of 98.6% @ IoU = 0.5 and an average real-time speed of detection of 30.6 FPS on a GPU-enabled computer and over 20FPS for a computer that is only CPU-enabled. These results show that the model produced is applicable in real-time for detection both in GPU- and CPU-enabled devices. Paper code url: https://github.com/GREAT0/4--Mammals-Detection.git

Keywords: Conservation; Deep neural network; Mammals; Real-time; YOLO

INTRODUCTION

When attempts are made these days toward solving problems, on many occasions, what comes to mind is the method of deep learning which uses deep neural networks especially when the identified problem is related to computer vision. The identified challenges of computer vision lie in the region of classification, localization, and tracking as well as object detection and deep learning methods have successfully been employed in deciphering solutions to these problems (Sermanet et al., 2014). When solving classification problems, the most

conspicuous object that is present in an image is assigned a category (Melek et al., 2019), in localization, the general conception is that the position of an object is determined in an image (Ejiyi et al., 2021). When the need arises for classification to be done alongside localization, it is observed that objects and their positions are determined and categorized as well. When it comes to the problem of object detection, an image having multiple objects is passed through a network and all object classes are not only identified with labels but their locations are also determined (Russakovsky et al., 2015). This makes object detection not only more fascinating but also more tedious than classification, as well as localization when considered distinctly.

Some of the most popular techniques that are deep learning-based and have been employed in the detection of objects include; Region-based Convolutional Neural Networks (RCNN) which is a double-stage detector alongside Fast-RCNN as well as Faster-RCNN. Object detection algorithms in this category work in two steps in which the first step involves the detection of the region of the images that contain the objects of interest while the second step consists in classification which puts a label on the objects if they are really present at the proposed region (Verma and Gupta, 2018). The other category of object detectors, are You Only Look Once (YOLO), and Single Shot multi-box Detector (SSD) among others which are classified as single-stage detectors. Both the double-stage detectors as well as the single-stage detectors have been used in the task of detection and they have shown goods performance, it, therefore, follows that the purpose for which you want to employ the detector will determine to a large extent, the particular model you will use.

Some of the notable game reserves in the world have attracted the interest of tourists because of some of the animals that are found in them. Some of these animals that attracted our interest include the African elephant, the Cape buffalo, the Rhino, and Zebra which are quickly becoming very scarce. The scarcity of these mammals is attributed to the activities of men in the natural habitat of these mammals. According to some tales, when hunters especially in Africa succeed in getting hold of any of these animals, they are given certain titles and exempted from the village tax payment among others, leading to a very high hunt for these animals. Although the difficulty of hunting these animals on foot, as well as their adaptive features, has helped these animals to live and survive in contemporary ages, they are rapidly going into extinction because of how fast man is depleting their habitat owing to the imminent development needs in the society. In addition to that, some of the parts of these animals are used for some traditional festivals and for some rituals making them hunted after as high-demand commodities. To help preserve these animals, in many countries, laws have been put in place for their protection but that is not sufficient enough since people still kill these animals in disguise as an accident, and their natural habitat is destroyed because of development and technological advancements. Therefore, having a system that will be able to detect these animals in real-time will be of great benefit in conservating the animals and protecting them from unwanted or unauthorized hunting.

One of the major issues to arrest when dealing with animal detection which is key to their conservation has been identified as that of the accuracy of the detection as well as speed (Verma and Gupta, 2018) since animals are always in motion. Single-stage Deep Neural Networks like YOLO have shown exceptional performance in detection when applied to the real-time detection of objects (Lan et al., 2018). Although the likes of Faster RCNN have shown good performance for detection too but rank behind the recent versions of single-staged detectors for detection in real-time. We therefore, in this work, chose to use the YOLO network model for this task because of how effective it is in real-time detection. We use the fourth version of YOLO for this work because of several advantages it has over other detectors including its applicability in mobile devices. One of the purposes for considering the development of a model for the detection of wild animals is to foster the conservation of these animals that have almost gone to extinction owing to human's quest for economic development which is insatiable (Nguyen et al., 2017). The natural habitat of the wild animals as well as their behaviors has also been affected greatly by the increase in the human population who not only hunt these animals but also transform their habitat into what suits them best. In an attempt to conserve some of these mammals who are more endangered on exposure than other animals, they have been moved to different habitats where they are not used to which has caused some of them to elicit lots of disruptions to humans as well as the environment (Vitousek et al., 2008). The real-time detection system we put forward will therefore help to prevent accidents that may occur of the animals colliding with human-operated machines like vehicles since the occurrence of such accidents does not only affect the animals but also the humans including the property of the humans such as the involved vehicle.

The contribution of this manuscript lies in the region of developing a rich dataset alongside their annotation for the training of deep neural networks for the real-time detection of mammals within the category we used. Implemented the dataset so developed on YOLO with increased speed of detection in real-time which is better than any other seen in the literature. To the best of our knowledge, this is the first implementation of this kind of network in this field and for this purpose.

The rest of the manuscript is structured into related

works, the model used and the processes of detection, results, and conclusions

Literature review

Images from the standard camera-trap dataset were used by (Verma and Gupta, 2018) to train a model to be able to verify animals as well as background patches. They identified that the associated challenges with the trained model lie in the region of noticeable variations in the background ranging from illumination differences, and shadows as well as positional changes of objects that are not relevant to the study like the leaves from the trees and their branches (Verma and Gupta, 2018). They used various deep neural network models to actualize the classification of the wild animals and obtained an accuracy of 91% and an F1 score of 0.95. Since their target was only classification, it cannot perform detection. Camera traps have been said to be one of the efficient methods for monitoring wildlife in natural observation since it collects can foster large natural data collection (Nguyen et al., 2017). After the data is collected, biologists who are interested in using the data still find it very difficult to analyze the data in terms of determining whether an animal is in an image or not and which of the species is present. Working at lessening the burden of manually analyzing the data, classification methods based on deep convolutional neural networks have been employed for automatically classifying the wildlife. Yu *et al* (Yu et al., 2013) reported the use of sparse coding spatial pyramid matching (ScSPM) of (Jianchao et al., 2010) that is improved to achieve the classification of images that contains wildlife. They used a dataset that has 7,196 images with 18 species. In their method, in the first stage, animal objects are detected manually and also cropped out from the background, after which the extraction of the image features using ScSPM. The ScSPM tends to convert an image or the proposed bounding box to one vector which is finally used for classification by applying the linear multi-class SVM to it. They achieved an accuracy of 82% in their classification. In (Chen et al., 2014) Chen *et al* implemented a model that is CNN-based that carries out segmentation of the images automatically. They designed the network with three convolutional layers whose filter size is 9 x 9 which is comprehensively followed by a 2 x 2 kernel size maxpool layer and climaxed by a fully connected layer alongside a softmax layer. In contrast to the work of (Yu et al., 2013), Chen *et al* (Chen et al., 2014) made the animal cropping to be done automatically by applying the Ensemble Video Object Cut (EVOC) of Ren, Han, and He (Ren et al., 2013) which is an automated segmentation method. Despite the fact that it is automated and CNN-based, its performance was not good for the application since it only achieved a recognition result of 38.32% which is considered poor. It is worth noting that although it outperformed the traditional models that are based on Bag-of-visual-words used for image classification (Fei-Fei and Perona, 2005) Blei et al., 2003) it is far below human level and expectations. Gomez *et al* in their work used a deep CNN model as well as the ImageNet dataset to overcome the challenge encountered with the identification of wild animals on large scale. (Fei-Fei and Perona, 2005) alongside (Blei et al., 2003) pointed out that most of the CNN models are better re-trained on pre-trained models or weights from the ImageNet dataset using some fine-tuning techniques. This is in line with the assumption that networks pre-trained on a dataset such as ImageNet that is large would have learned features well for a good number of classification as well as detection problems. Although this is said to be true for approaches that are data-driven, it has helped to enhance performance even when the available data is small (Fei-Fei and Perona, 2005). Although their model achieved accuracies of 88.9% for top-1 and 98.1% for top-5 but did not give a solution to the problem of filtering out non-animals images in animal detection (Nguyen et al., 2017; Fei-Fei and Perona, 2005). Nguyen *et al* (Nguyen et al., 2017) got inspired by the works done previously using the deep CNN-based model and as such used that for the classification of wild animals employing the Spotter dataset. On discovering that the dataset contains an appreciable amount of blank images, that is images that animals are absent in them, they did image filtering before identifying the animals. They also worked at training the model both from scratch and from a pre-trained model from the popular ImageNet dataset by fine tuning. Their model ended up achieving above 90% recognition accuracy.

Taking a brief tour of the algorithms used for object detection as well as the dataset, we have mentioned the most commonly used ones, which are typically deep CNN-based. A convolutional neural network (CNN) is described as a subsidiary of artificial neural networks specialized with the use of perceptron for the purpose of analyzing large amounts of data under the supervised category of machine learning. Beyond object detection and image processing, CNNs are applicable for natural language processing as well as other kinds of cognitive tasks (Banupriya et al., 2020). The structure of CNN as well as its ease of building not counting out the accuracy it has achieved are some of the reasons why researchers depend on CNN for object detection and other related tasks. When YOLO was introduced in 2016 by Redmon *et al* in (Redmon et al., 2016) and (J. Redmon and Farhadi, 2017), they came up with an approach that differs from those of RCNN for object detection. The major concern in the YOLO algorithm was its implementation in real-time while achieving high accuracy. Although YOLOv1 had a good speed of

detection for real-time applications, its performances needed apt increment which was done in YOLOv2 by implementing various approaches (Redmon and Farhadi, 2017). The YOLO network comprises only feature extraction which is followed by the prediction, in the feature extraction phase, the entire image is usually sliced into *s* x *s*, then a relationship that is regression-based is initiated between these sliced parts and the number of objects that are to be detected designated by *B*. This relationship is duly initiated when the center of the object to be detected is found to fall in any of the sliced boxes. Detection is then done where the class label alongside the bounding box associated with the object class is given as the output just in a single network. Monitoring animals especially mammals that are wild or endangered is very essential as researchers use pieces of evidence from there to give information to the managers as well as the hierarchies of the conservation of the natural resources to work at maintaining the balance in the ecosystem (Nguyen et al., 2017).

The model used

The overall architectural design for YOLOv4 is depicted in (Figure 1). The figure shows that it is divided into three parts: the backbone, the neck, and the head.

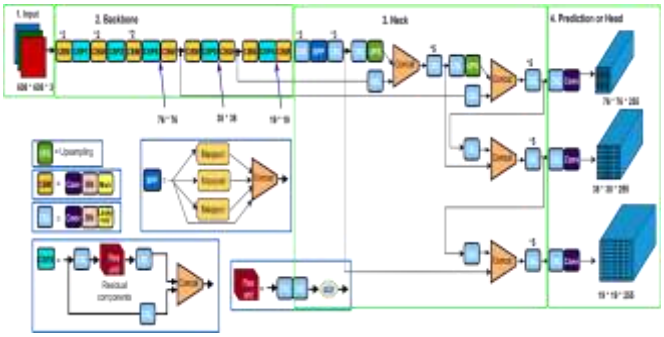

Figure 1: YOLOv4 Architecture

Each of the model's three sections must be carefully chosen in order to fully utilize the network's potential and strike the necessary balance between the input resolution, the number of convolutional layers, the number of parameters provided, and the number of output layer filters. For the backbone we used CSPDarknet, for the neck region of the network, we used Spatial Pyramid Pooling (SPP) alongside the Path Aggregation Network (PAN) and for the head, we used version 3 of YOLO. The backbone is primarily for feature extraction, the neck region brings an increment to the receptive field of the features as well as an increment to the detection accuracy, while the head is the bedrock of the detection (Ejiyi et al., 2021). The five layers of YOLO v3 are very useful for the detection of various sizes of objects in the network. These layers are the "*convolutional* layer", "*upsample* layer", "*route* layer", "*shortcut* layer", and "*yolo* layer".

The process of the detection of the mammals

The image of the mammal in question is taken as input in 416 * 416 pixels or 608 *608 which takes a much longer time for detection in comparison to the 416 input size (Li et al., 2018). A series of operations of both convolution as well as batch normalizations is carried out on the image so inputted before it is down sampled 32 times, 16 times, and 8 times – 3 times in total to obtain a feature map of multi-scale (J. Redmon and Farhadi, 2018). The detection just as the training procedure divides the input image into *n* x *n* grid cells often called S x S grid. Each of the cells takes the responsibility of predicting the possible bounding boxes designated as B, each characterized by 5 values – *x, y, w, h,* and *c*. *x* and *y* are the coordinates of the bounding boxes' center in relation to the bounds of the cell (Zou et al., 2019). *W* and *h* stand for the width and heights respectively of the bounding box in relation to the dimension of the image (Redmon, 2016). Finally, *c* stands for the confidence score showing the degree of confidence that the model has over the presence of a required object in the box. It determines also how sure the shape of the box has encompassed the object but does not have information on the class for which the encompassed object belongs (Bochkovskiy et al., 2020).

In a situation where the cell houses no object the confidence is set at 0. But if not it is set as Intersection over Union (IoU) defined by (1) below and is the IoU that is got between the generated bounding box and the ground truth. Another computable parameter is the class probability which computes the possibility of an identified object belonging to a given class. Typically, the model predicts many bounding boxes, and post-processing methodology in the form of Non-maximum Suppression (NMS) is used to ensure that the most appropriate bounding box is left under a set threshold (Kathuria, 2018).

$$c = \Pr(Obj) * IOU_{pred}^{truth} \quad (1)$$

The dataset we used consists in 1956 images of which 452 which stands for about 23% of the dataset were used for validation while the rest was used for the training of the model. The images are taken from different backgrounds and with various cameras to ensure that the model learns well having good robustness. Some of the images of the dataset are shown in (Figure 2). We made use of transfer learning utilizing a pre-trained weight from ImageNet and adjusted the configuration file of the original YOLO model to suit our needs and then started off the training (Huang et al., 2020). We used the Darknet

Figure. 2: Some images from the dataset.

framework for the training on GeForce GTX 2080Ti/PCIe/SSE2 GPU system with an Intel (R) Core (TM) i9-9900k CPU. The graphics card in the system is actually two GPUs, each with 12GB GDDR5 GPU RAM (VRAM). This permitted a computational speed that is up to 6.6 TFLOPs when running on one precision mode Ubuntu 18.04.4 LTS. The Darknet framework itself was the original concept of the developers of YOLO, it is Linux based and runs well on computers that are GPU-based. We also used OpenCV for image processing. The general flowchart for the method and implementation is shown in (Figure 3).

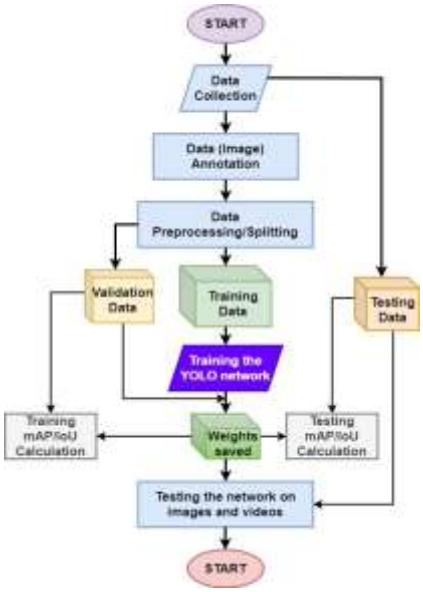

Figure 3: The flowchart of the system.

It starts with data collection from various sources and their subsequent annotation using the labeling tool. Here the images are surrounded with bounding boxes and the coordinates of the position taken are fed into the network. The data is preprocessed thoroughly and split into training, and validation sets in the ratio of 7:2. The data (training and validation) are used for the training of the YOLO v4 model with a 5-fold cross-validation strategy, and the weights are saved at the set checkpoints. The saved weights – the best one is used for testing using the test set of the data. Meanwhile, both during the training and testing, the IoU and mAP are calculated while the training loss can only be estimated during the training process. The testing set was other images found in the environment which were not annotated and preprocessed.

We used the transfer learning technique for the YOLO v4 network training, which used a pretrained weight on MS COCO as the base for the training rather than training from scratch. This reduces the training time because the weights of the layers are only updated from what it was from the previously trained data by feeding in the new data.

RESULTS AND DISCUSSION

During the training process, the mean Average Precision (mAP) was calculated as well as the training loss. The chart in (Figure 4) shows the training loss and mAP obtained which are 0.9038 and 98.6% respectively. The mAP obtained is at the 50% threshold of IoU. The maximum batch was set as well as the minimum batch at the configuration file of the model to 90 % and 80% of the number of iterations which we set at 20000. The detection results obtained are also shown below which depict that the model achieved an accuracy of above 90%, especially for real-time detection. When used for detection in real-time or video the frames processed per second is in the region of 30 without any significant drop in the accuracy of the model. The other metrics that helped in ascertaining the performance of the model are the True Positives, True negatives, false positives, and false negatives defined and used by (Ejiyi et al., 2022).

Figure 5 shows the individual result obtained from the model of the classes. The Average Precision (AP), the True Positive (TP), and the False Positive (FP) for each class of the dataset were displayed also. At the IoU set at 50%, the other values obtained such as the Precision, Recall, F1_score, and the average IoU were all displayed in (Table 1).

When we set the IoU to 0.75 upper quartiles the result obtained for each class was shown in (Figure 6). The other values of importance like the AP, TP, and FP were all shown in the same figure. We have also displayed the Precision, Recall, F1 Score as well as the average IoU in (Table 1). Finally, when the IoU was set at the upper

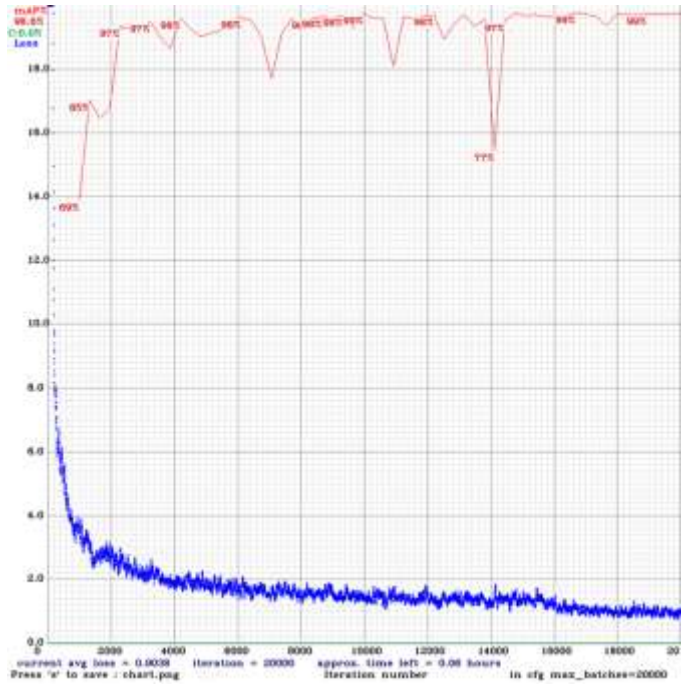

Figure. 4: The training plot.

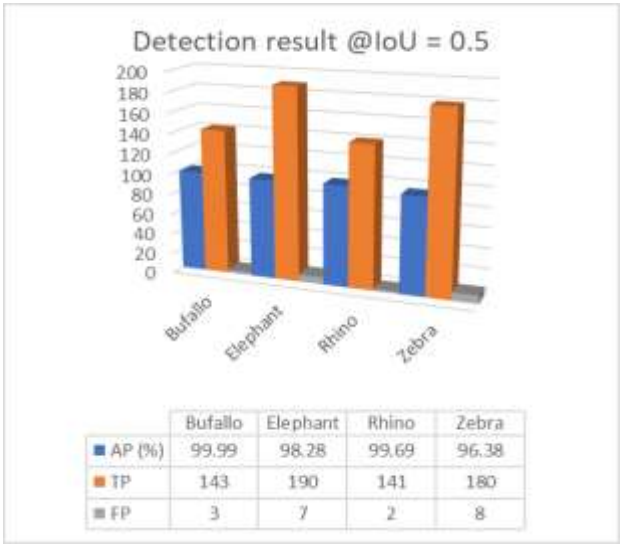

Figure 5: Detection chart at IoU = 0.5

quartile, the mAP obtained was 0.9539 (95.39%). At IoU set at 0.25 which is the lower quartile, the result obtained is what is charted in (Figure 5) with the AP, TP, and FP too. In (Table 1), we have shown the Precision, Recall, F1 Score, and Average IoU. The mAP obtained here is 0.9883 (98.83%).

The result shows that at any point that the IoU is set, the model performed very well with very high mAP. Generally, when the mAP is set at 0.75 for most models, there is a significant drop in the mAP because the IoU is

Table 1: Detection results.

	Iou = 0.25	*IoU = 0.50*	*IoU = 0.75*
Precision	0.97	0.97	0.95
Recall	0.98	0.97	0.96
F1 Score	0.98	0.97	0.95
Average IoU	87.61	87.02	86.13
mAP	0.9883	0.9858	0.9539

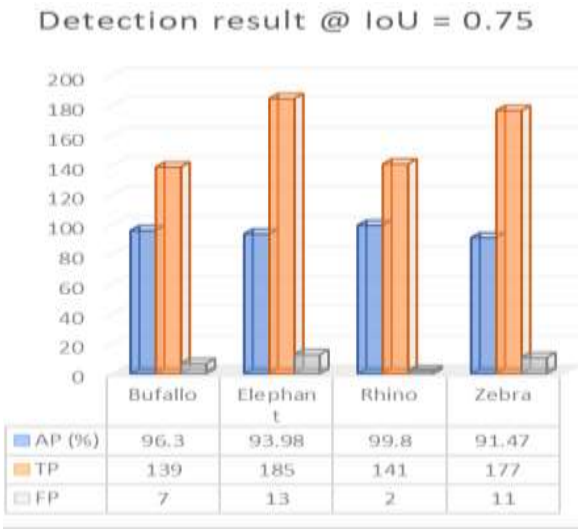

Figure 6: Detection chart at IoU = 0.75

to be at least 75% for the detection to be considered true else it is false. The robustness of the model was also shown by the result obtained when the IoU was set at 0.25, there was not much gain in mAP. This is a clear indication that the model learned well and can perform well at any quartile of mAP.

In line with the above, the average IoU in (Table 1) is the average IoU recorded for the whole detections at the set threshold of IoU. A look at table 1 shows that for the testing set, the average IoU was bigger as expected at IoU = 0.25 (Figure 7) and smallest at IoU = 0.75 but the difference is 1.48 amounting to about 1.78%.

When compared with the AP obtained from the MS COCO dataset used in the training of the YOLO model of (Bochkovskiy et al., 2020) which was 65.7% at IoU = 0.5, ours is 0.97 which is way better. Although their model was trained to detect 80 classes while ours was 4 if the number of classes is increased, our model promises to perform well since many of the hyperparameters were adjusted to suit our demand. For other models like YOLO v3 on MS COCO dataset, mAP was 57.9 for an input size of 608 and 55.3 for an input size of 416 which are not close to our model. Since this is the first application of a neural network for the real-time detection of mammals to the best of our knowledge, there are no adequate comparisons on this basis.

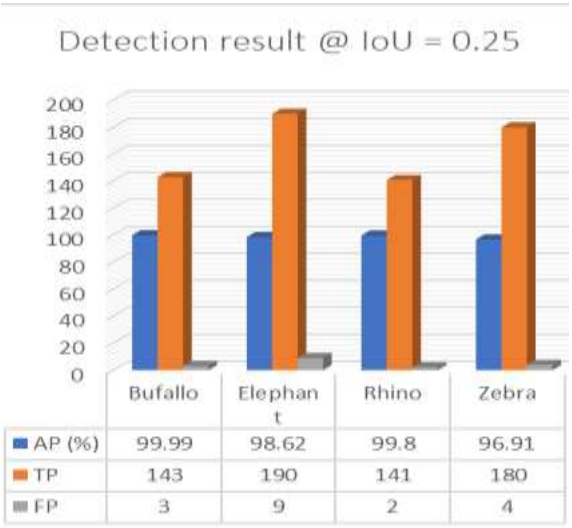

Figure 7: Detection chart at IoU = 0.25.

Some of the detection results are shown in (Figure 8) with the percentage of confidence of the model. Even in the face of low illumination or some of the patches on the animals, the model was able to detect it. This was made possible because of the data augmentation methods employed in the training set. For some animals who use adaptive mimicry type of adaptation or background that can be considered complex, the model was able to detect the animal showing that it is very effective for this task. For better functionality and applicability in this domain, a model should be able to also track the mammals so that in case they stray away, they can be traced or tracked. We will focus on that as well as on increasing the frames that are processed per second so that the model can be better fitting for real-time detection even for very high-speeding vehicles.

Because of the rate of similarities between the physical structures of Elephants, Rhinos, and Buffalos, the model usually detect them as the same from a distance but as the camera is brought closer to the image, it will distinctively detect them and appropriately classify them. This is the only identified challenge of the model. The model classifies Zebra very well in all instances because of how distinct it is from other classes additionally animals in the woods are easily tracked real-time while in motion. When other wild animals that were not contained in the dataset were presented to the model as can be seen from (Figure 8), it failed to detect them although they may be mammals. We are confident that if the number of classes of animals is increased, the model will also do a good job in detecting and tracking them.

Comparing the model, we used YOLO v4 with the popular double-stage detectors – FastRCNN and Faster RCNN, it has been ascertained that YOLO has a better speed of detection than both Fast and Faster-RCNN

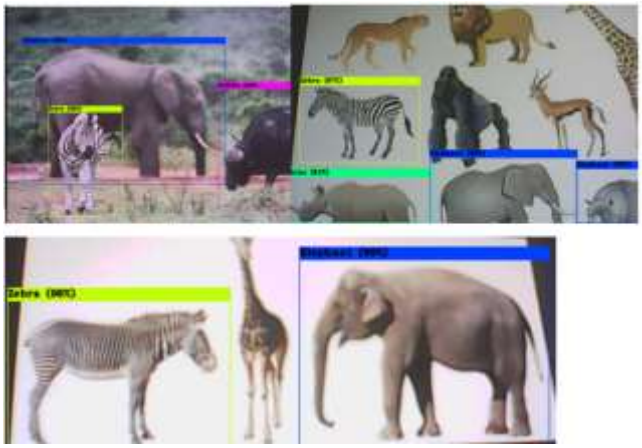

Figure 8 The detection result of the model.

which made it more suitable for real-time detection. While the double-stage detectors make use of region localization through search to find objects in the image, YOLO does not need to search the whole image but only the part of the image with a high probability of housing the object of interest. While Fast- and Faster RCNN may give better accuracy on detection, they are not ideal for application in real-time detection because of the time employed in searching the entire image for an object of interest. It is therefore more appropriate to use YOLO for this task since the accuracy and speed of detection are needed as the model is geared toward the detection of animals in real-time.

Conclusion

This paper successfully worked at annotating a custom dataset containing 4 mammals of which 3 of them have very close physical characteristics. The animals in the dataset are Buffalo, Rhino, Elephant, and Zebra. The YOLO model (version 4) was used to train the dataset for the purpose of implementation for real-time detection. We used some data augmentation methods to increase the robustness of the model and achieved a mean Average Precision (mAP) of 98.6% during training at IoU threshold of 0.5. When the model was employed for real-time detection it achieved an average FPS 30.6 FPS on the computer it was trained on and a little above 20FPS on CPU computers. This can be applied for real-time detection in many scenarios like video surveillance, monitoring of traffic especially with respect to these animals, and possibly tracking. The experiment showed that our model works well even in a situation where the lighting is poor.

Funding

This work was supported in part by the National Natural

Science Foundation of China (No.62072074, No. 62076054, No. 62027827, No. 61902054), the Sichuan Science and Technology Innovation Platform and Talent Plan (No. 2020JDJQ0020), the Sichuan Science and Technology Support Plan (No. 2020YFSY0010), and the Natural Science Foundation of Guandong Province (No. 2018A030313354).

REFERENCES

Banupriya, N., Saranya, S., Jayakumar, R., Swaminathan, R., Harikumar, S., and Palanisamy, S. (2020). Animal detection using deep learning algorithm. In *Journal of Critical Reviews*. https://doi.org/10.31838/jcr.07.01.85

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*. https://doi.org/10.1016/b978-0-12-411519-4.00006-9

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. http://arxiv.org/abs/2004.10934

Chen, G., Han, T. X., He, Z., Kays, R., and Forrester, T. (2014). Deep convolutional neural network based species recognition for wild animal monitoring. *2014 IEEE International Conference on Image Processing, ICIP 2014*. https://doi.org/10.1109/ICIP.2014.7025172

Ejiyi, C. J., Bamisile, O., Ugochi, N., Zhen, Q., Ilakoze, N., and Ijeoma, C. (2021). Systematic Advancement of Yolo Object Detector For Real-Time Detection of Objects. *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 279–284. https://doi.org/10.1109/ICCWAMTIP53232.2021.9674163

Ejiyi, C. J., Qin, Z., Adetunji Salako, A., Nkanta Happy, M., Ugochi Nneji, G., Chima Ukwuoma, C., Amuche Chikwendu, I., and Gen, J. (2022). Comparative Analysis of Building Insurance Prediction Using Some Machine Learning Algorithms. *International Journal of Interactive Multimedia and Artificial Intelligence*, *7*(Special Issue on Artificial Intelligence in Economics, Finance and Business), 75–85. https://doi.org/10.9781/ijimai.2022.02.005

Fei-Fei, L., and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*. https://doi.org/10.1109/CVPR.2005.16

Huang, Z., Wang, J., Fu, X., Yu, T., Guo, Y., and Wang, R. (2020). DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. *Information Sciences*. https://doi.org/10.1016/j.ins.2020.02.067

Jianchao Yang, Kai Yu, Yihong Gong, and Huang, T. (2010). *Linear spatial pyramid matching using sparse coding for image classification*. https://doi.org/10.1109/cvpr.2009.5206757

Kathuria, A. (2018). *What's new in YOLO v3?* Towards Data Science.

Lan, W., Dang, J., Wang, Y., and Wang, S. (2018). Pedestrian detection based on yolo network model. *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*. https://doi.org/10.1109/ICMA.2018.8484698

Li, G., Song, Z., and Fu, Q. (2018). A New Method of Image Detection for Small Datasets under the Framework of YOLO Network. *Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2018*, *Iaeac*, 1031–1035. https://doi.org/10.1109/IAEAC.2018.8577214

Melek, C. G., Sonmez, E. B., and Albayrak, S. (2019). Object Detection in Shelf Images with YOLO. *EUROCON 2019 - 18th International Conference on Smart Technologies*, 1–5. https://doi.org/10.1109/EUROCON.2019.8861817

Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., Ritchie, E. G., and Phung, D. (2017). Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. *Proceedings - 2017 International Conference on Data Science and Advanced Analytics, DSAA 2017*. https://doi.org/10.1109/DSAA.2017.31

Redmon, J. S. D. R. G. A. F. (2016). (YOLO) You Only Look Once. *Cvpr*. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., and Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., and Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *Computer Vision and Pattern Recognition (Cs.CV) ArXiv:1804.02767 [Cs.CV]*. http://arxiv.org/abs/1804.02767

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.91

Ren, X., Han, T. X., and He, Z. (2013). Ensemble video object cut in highly dynamic scenes. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2013.254

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-015-0816-y

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*.

Verma, G. K., and Gupta, P. (2018). Wild Animal Detection from Highly Cluttered Images Using Deep Convolutional Neural Network. *International Journal of Computational Intelligence and Applications*. https://doi.org/10.1142/S1469026818500219

Vitousek, P. M., Mooney, H. A., Lubchenco, J., and Melillo, J. M. (2008). Human domination of Earth's ecosystems. In *Urban Ecology: An International Perspective on the Interaction Between Humans and Nature*. https://doi.org/10.1007/978-0-387-73412-5_1

Yu, X., Wang, J., Kays, R., Jansen, P. A., Wang, T., and Huang, T. (2013). Automated identification of animal species in camera trap images. *Eurasip Journal on Image and Video Processing*. https://doi.org/10.1186/1687-5281-2013-52

Zou, Z., Shi, Z., Guo, Y., and Ye, J. (2019). *Object Detection in 20 Years: A Survey*. 1–39. http://arxiv.org/abs/1905.05055.