Computer Vision for Wildlife Conservation

Shankho Ghosh, Ketan Muddalkar, Ajay Patel, Vishnu B Nair
Balmukund Mishra, Divya Acharya, Tejalal Choudhary

Abstract

In our Ecosystem, wildlife is a very important to maintain our nature. So protection and conservation of wildlife is our responsibility, those which are at risk of being extinct. One such species Amur tiger is in danger zone. Therefore, we have implemented three object detection methods using different deep learning technique for detecting Amur tiger which can further be deployed in UAVs. Using the recently released ATWR (Amur Tiger Re-identification in the wild) dataset by Computer Vision for Wildlife Conservation that contain 2485 images along with their annotations for training and 277 images along with their annotations for validation. By using our model tiger can be tracked and detected easily. This will help to conserve wildlife better and can be later extended to different flora and fauna.

Introduction

❖In wildlife conservation number of Amur tiger decreases. Only 500 Amur tiger are left in the planet.

❖ Wildlife conservation by the object detection using deep learning is a our challenge.

❖Previous ,To save Amur tiger methods of attaching transmitters to wildlife has been used. But there are many problems associated with this like image classification, image localization, sensor failures, difficulty in scaling large population.

Fig. 1 Sample Image

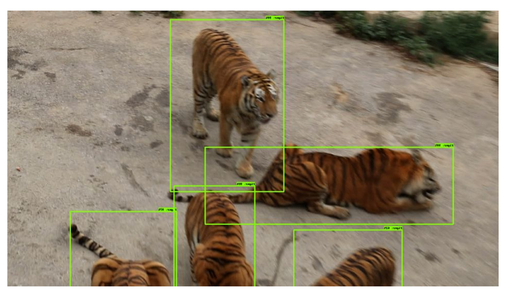

Fig. 2 Expected Output

❖By using of deep learning technique for object detection may easy to identify Amur tiger using vehicles drones and cameras.

❖ In object detection lot of poses variation, complex Background etc. Make more complex in detection.To handle this we use ATRW dataset that make our process easily .

Proposed Method

Methods - (1) *Faster R-CNN Inception v2* (2) SSD Inception v2

•Faster RCNN uses higher computation power, hence giving us higher accuracy models with low loss. This type of model is not suitable for mobile devices as the inference time would be much higher.

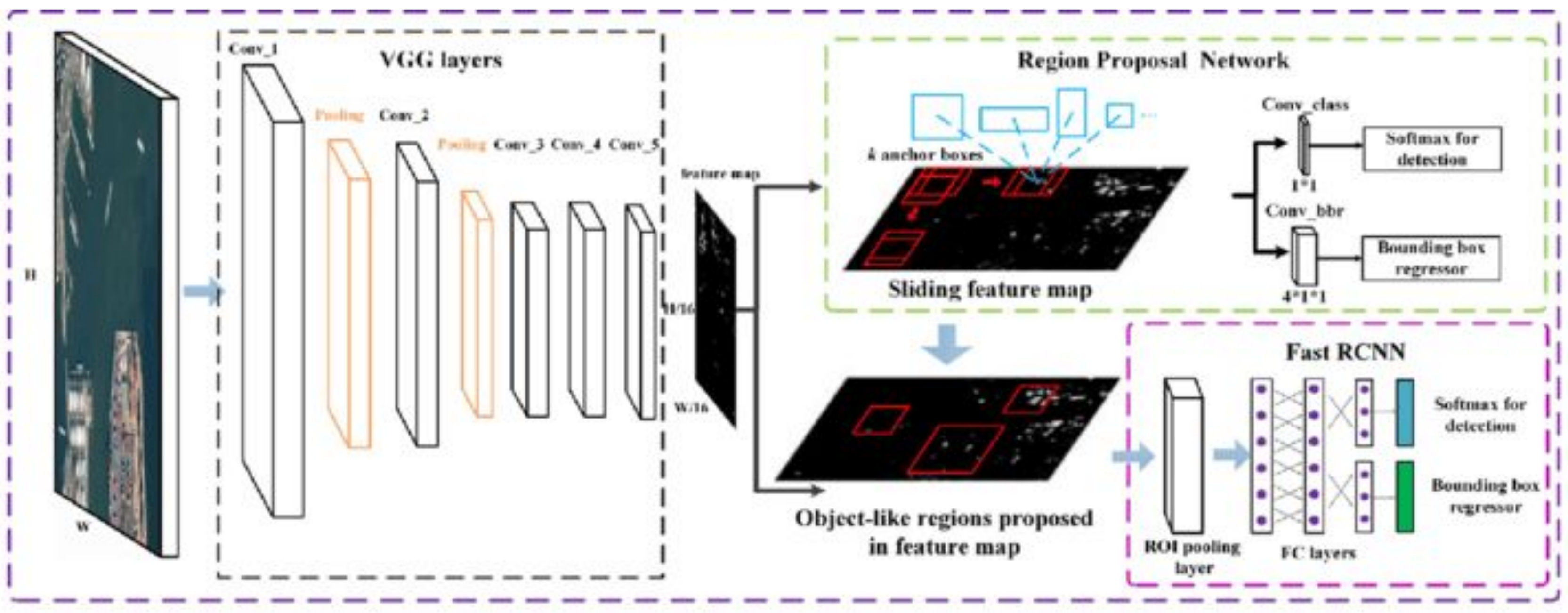

Fig. 3 Architecture of Faster R-CNN

•The architecture of SSD Inception V2 is based on single shot object detection and hence it gives lower accuracy models as compared to Faster RCNN, but with much less inference time, suitable of low computational mobile devices.

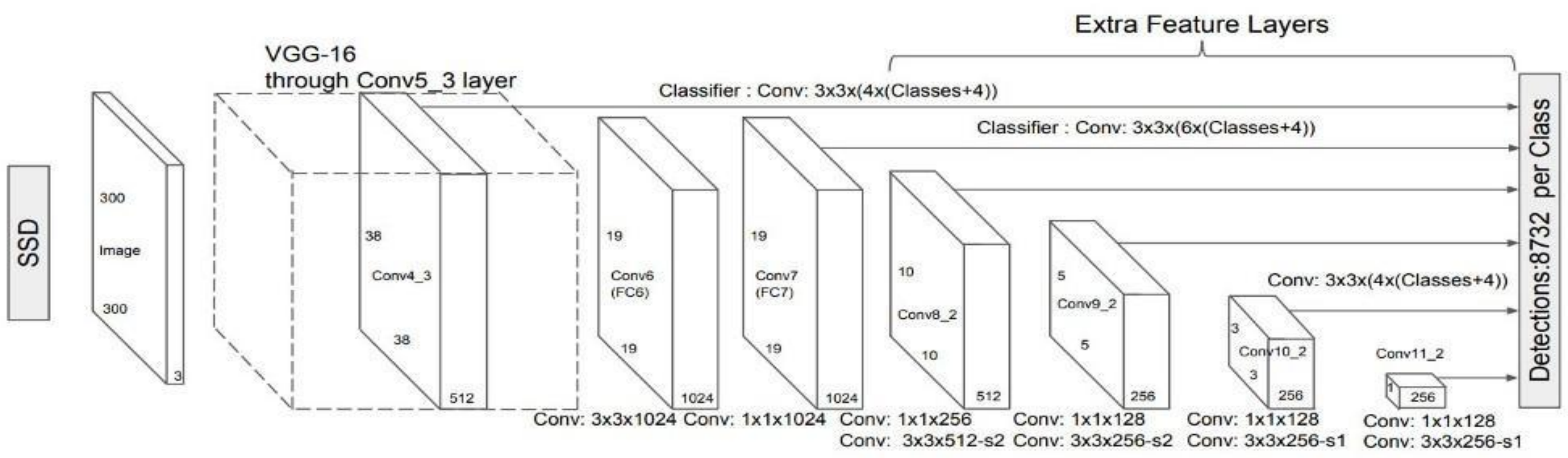

Fig. 4 Architecture of SSD

Experimental Results and Discussion

Faster RCNN Inception v2, SSD Inception v2, SSD Lite Mobilenet were the models used for training. When the dataset was trained on Faster RCNN model over 94% accuracy was observed. While using SSD, an accuracy of over 90% was observed and when SSD lite was trained over 95% accuracy was observed. This was implemented using Tensorflow 1.5 backend by tuning hyper parameters in various architecture obtained from the Tensorflow model zoo for object detection.

Models / *Parameters*	Faster R-CNN INCEPTION v2	SSD INCEPTION v2	SSD LITE MOBILENET
mAP@IoU 0.5	0.9437	0.9055	0.9554
Steps	2,00,000	1,00,000	1.30,000
LOSS (classification)	0.123126	5.8326	3.052575
LOSS (localization)	0.112125	0.662244	0.442702

Accuracy Analysis of different models

Observed Output

Fig.6 **SSD Lite MobileNet**

Fig.7 **SSD Inception v2**

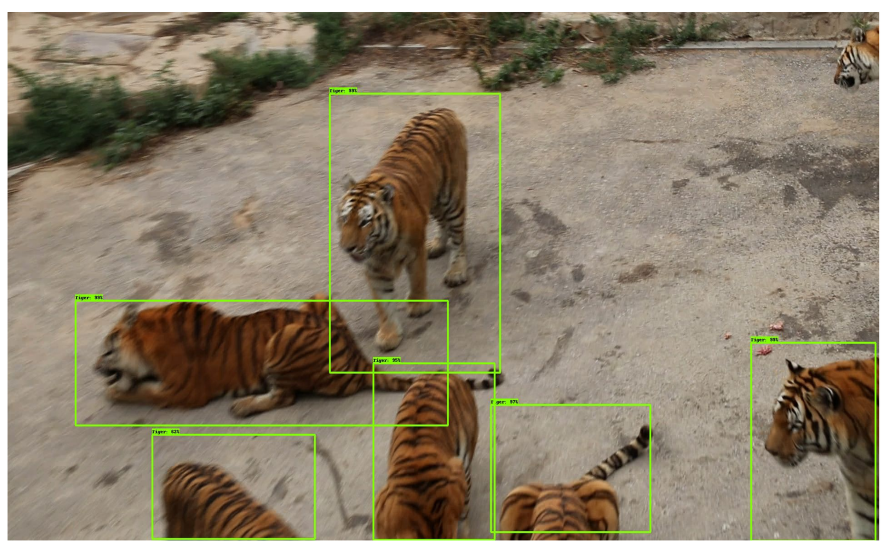

Fig. 8 **Faster R-CNN Inception v2**

Conclusions

We have presented tiger detection system. It help us to conserve tigers and which can further lead the way for other flora and fauna. Using the recently released ATRW dataset by CVWC, we experimented on 2485 images along with their annotation for training and 277 image along with their annotation for validation. The proposed system achieves an accuracy of more than 90% in three models, these models can be further applied for different applications according to system requirements. This project can be further progressed by training on tiger pose detection and Re-ID datasets by CVWC and deploying it on mobile computers like Raspberry Pi and NVIDIA Jetson Nano extending upto drones.

References

▪https://cvwc2019.github.io/challenge.html
▪https://github.com/Edje-Electronic/tensorflow-object detection
▪https://arxiv.org/pdf/1906.05586.pdf
▪https://ieeexplore.ieee.org/document/8510564

Team No. - 31