IE 406 - ML

Final Project Report

Animal Detection and Classification

Chirayu Chaplot (201801038)
Dhanvi Shah (201801167)
Rutwa Rami (201801205)
Shabbir Murtaza (201801428)
Dishita Thaker (201801442)

November 21, 2021

Contents

1 Introduction

1.1 Motivation

With the impending threat of extinction on various species of animals, bio-diversity preservation and protection is a priority. Wildlife habitats hence have been common: natural or artificially manoeuvred. In such cases, surveillance of animals, keeping a count of the number of animals belonging to a particular species, keeping an eye on the illegal poaching and killing of animals, maintaining logs of tourists activities to restrict them to a permissible level,etc. are crucial activities that will require digital monitoring of animals via cams, drones, etc. In such a case, manually keeping counts, or even real-time identification of animals is very difficult and inefficient.
Hence, in such cases, a real-time algorithm to detect presence of animals in a given frame, can make the process smooth and efficient. Essentially an algorithm that detects the presence of animals in a given shot, and detect the animal presence can help.
Such algorithms can also be used to prevent negative human-animal interventions.

1.2 Problem Statement

The problem statement we have worked on is essentially that given an image, the presence of an animal must be detected from the image and the name of the animal, it's position by a bounding box,should be made known.
Typical Input shall be an image, the expected output is the detection of animal presence, if present; name of the animal present in the image and it's position.

2 Data-sets

We have employed the YOLO-v3 algorithm in order to detect the presence of animals in an image. We have essentially used the weights that are obtained by training on the COCO data set (the link for the same has been embedded in the references).
The COCO(Common Objects in Context) data set by MS, is a widely used data set used for deep learning problems in the field of object detection.
It has around 80 classes or labels each of which correspond to a common object in context, and has high resolution images pertaining to it. It has among these nearly 8 classes of animals, namely 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe'.
The algorithm hence will predict any of these animals, while it won't recognise those not listed here.
We can keep improving the algorithm's depth by training it on more animal classes as and when required.
In order to test the results, we use several different images of animals from the animals10 data-set(link embedded in the reference).

3 Methodology

The block schematic for the methodology is as given below:

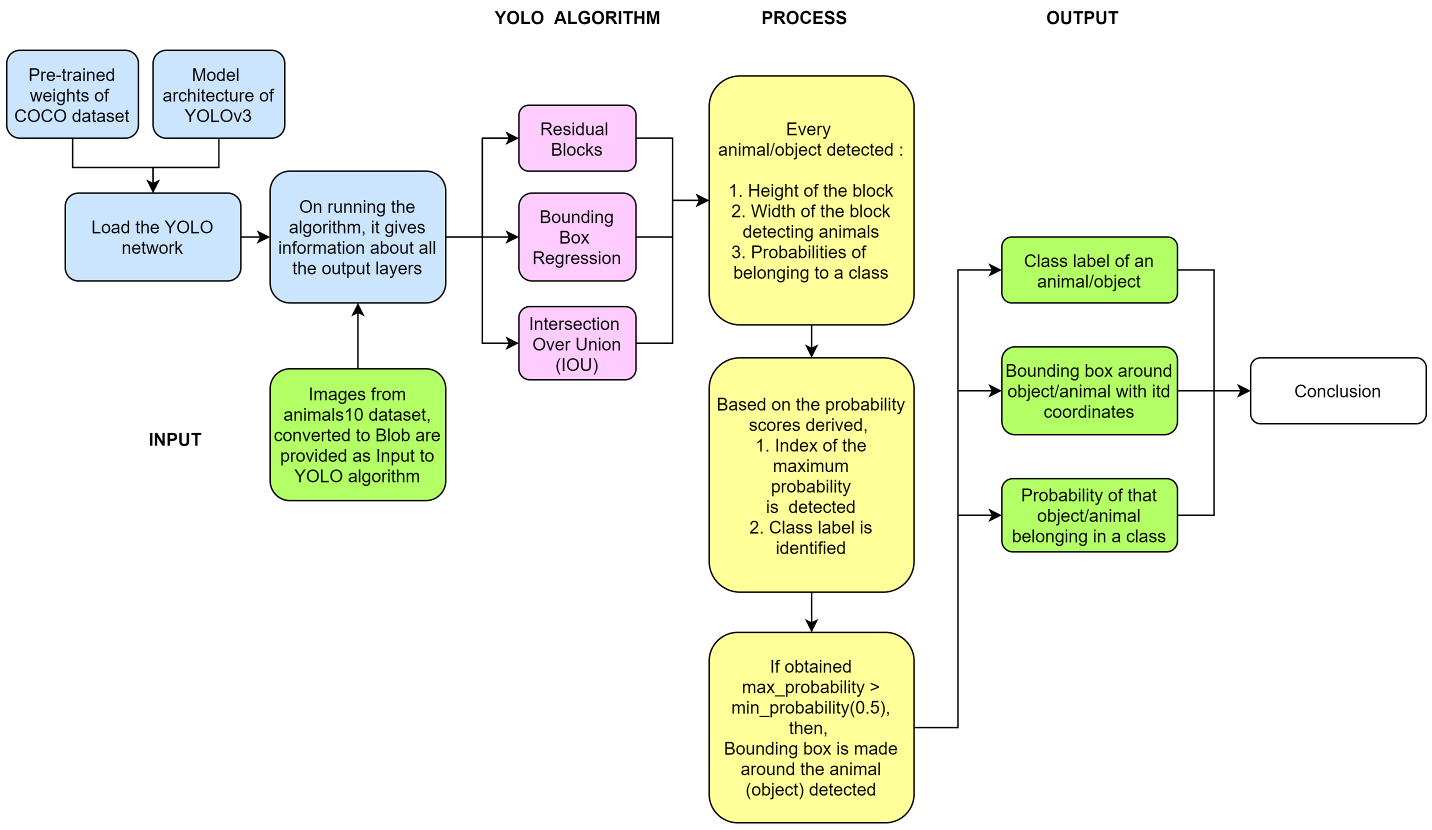
YOLO ALGORITHM
PROCESS
OUTPUT
Pre-trained weights of COCO dataset
Model architecture of YOLOv3
Load the YOLO network
On running the algorithm, it gives information about all the output layers
Residual Blocks
Bounding Box Regression
Intersection Over Union (IOU)
Every animal/object detected : 1. Height of the block 2. Width of the block detecting animals 3. Probabilities of belonging to a class
Based on the probability scores derived, 1. Index of the maximum probability is detected 2. Class label is identified
If obtained max_probability > min_probability(0.5), then, Bounding box is made around the animal (object) detected
Class label of an animal/object
Bounding box around object/animal with itd coordinates
Probability of that object/animal belonging in a class
Conclusion
INPUT
Images from animals10 dataset, converted to Blob are provided as Input to YOLO algorithm

Figure 1: Block schematic explanation of the methodology, explanation for which is provided on the next page

- The YOLO (You Only Look Once) algorithm that falls under the umbrella of Deep learning algorithms is one of the most well-known object detection algorithms. We employ the YOLO algorithm to specifically detecting the presence of animals by obtaining weights for the model that were trained on the COCO dataset.
- The YOLO algorithm is known to be very fast and efficient, hence it can help in real-time detection as mentioned in Motivation. Therefore, it has been employed by us.
- Initially, we configure the YOLO network to suit our needs i.e we load the YOLO network algorithm by using pre-trained weights of the COCO dataset; and using the architecture of YOLO v3. We employ the algorithm to run and hence obtain an output then containing the information of all layers. Now the YOLO algorithm is ready to take input and carry on detection on the same.
- Now, we consider an image from the animals10 dataset, convert it to blob format since YOLO accepts that as input. Further, taking this as input there occurs a forward propagation(using the previously obtained network layers' information) in order to obtain the network output.
- The obtained output contains an information output for each object detected; this carries information about the co-ordinates of the block detecting it, it's width and height - this is about the position information that locates the object(animal in this case) in the image.
- Following this is the set of probabilities of the object belonging to a particular class. (eg: 0.993 probability of belonging to 'cat' class, 0.001 probability of belonging to 'bicycle' class, etc.)
- From that set of information we set these probabilities' score aside, then find the maximum score and the index of that score to indicate the class to which it belongs to. Hence this finally gives us the set of detections, each with a class and the probability of object belonging to it.
- Further, we plot the bounding box by specifying a minimum probability, if the probability of the detection is more than that, then and only then will the bounding box be plotted and display the class name as well as the probability of the object belonging to it. We have defined this minimum probability as 50%
- Now, we obtain the position information of the bounding box, scale that relative to the image, make a bounding box using the width and height detected. We also print the class name to which it belongs to and the probability of it belonging to that class.
  The final image with bounding box, labels and probability is as discussed in the section below.

4 Results and Discussion

4.1 Output results on the test set

As mentioned above, the expected output is detection of the animal present in the image given as input; display the class of the animal it belongs to.
We tested the algorithm on a set of images, the outputs on different images is as follows:

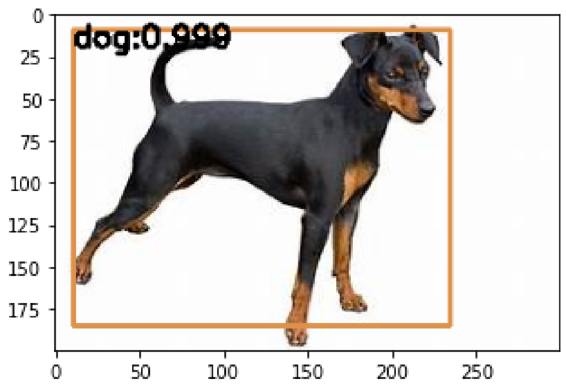

Figure 2: Detection of animals - result: dog is being detected with a bounding box and probability 0.999

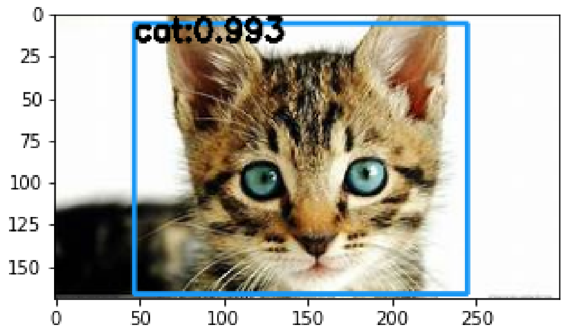

Figure 3: Detection of animals - result: cat is being detected with a bounding box and probability 0.993

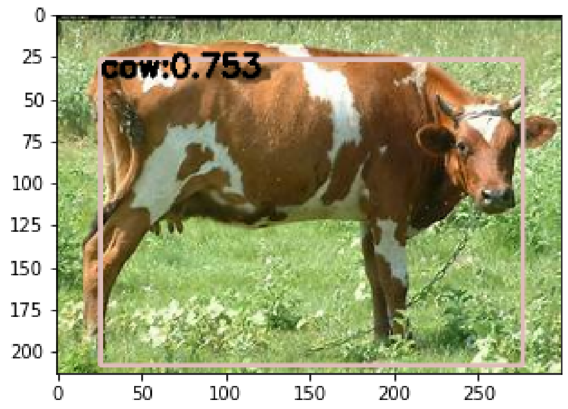

Figure 4: Detection of animals - result: cow is being detected with a bounding box and probability 0.753

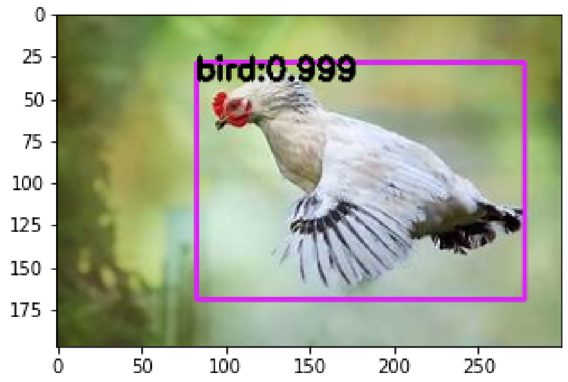

Figure 5: Detection of animals - result: bird is being detected with a bounding box and probability 0.999

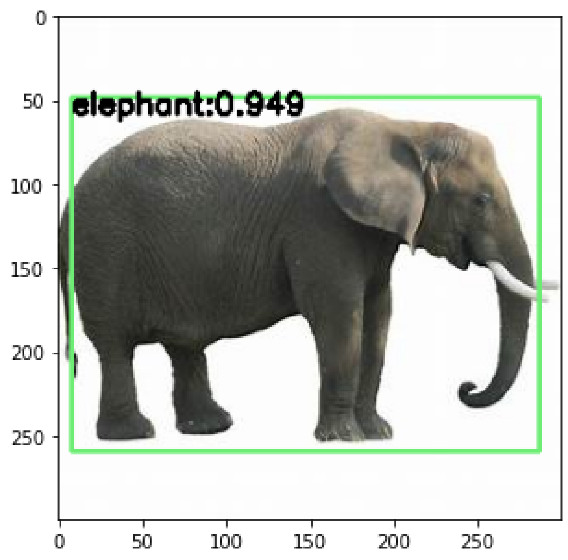

Figure 6: Detection of animals - result: elephant is being detected with a bounding box and probability 0.949

As we can observe in the figures obtained above, animals present in the image are detected. There is a single bounding box surrounds the animal to locate it's position in the image.It also displays a probability, it is the probability of the animal present there belonging to that particular class.

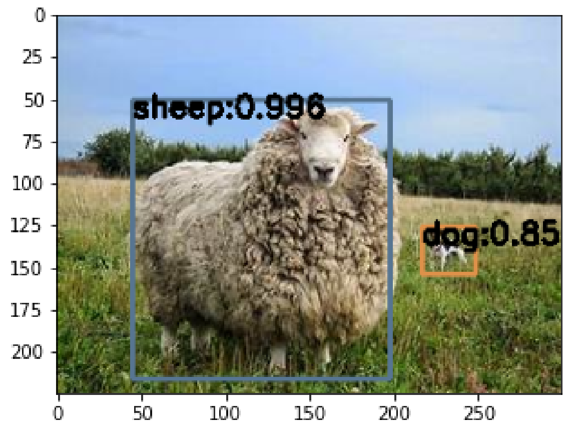

Figure 7: Detection of more than 1 animals/objects - result: sheep and dog is being detected with a bounding box and probabilities 0.996 and 0.85 respectively

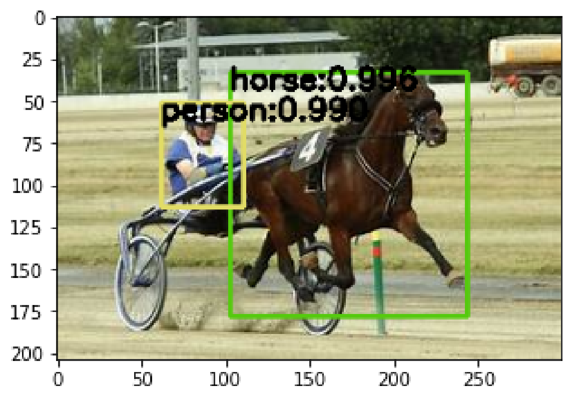

Figure 8: Detection of more than 1 animals/objects - result: horse is being detected with a bounding box and probability 0.996; person is also detected(prob: 0.990) since it belongs to the labels in COCO

As we can observe in the figures above, we tried to touch upon the corner cases too; two different animals are also detected in the image. Each comes with it's own bounding box and probability. An object: person is also detected alongside the horse; this is because we have used weights of the model trained on the COCO dataset and hence, it detects 'person' that was one of the 80 classes present in the COCO dataset.

Along with the pointing out in the image, we also extracted the co-ordinates of the bounding box. Each animal can hence be detected with a set of 2 dimensional co-ordinates. For instance, for the elephant that was detected:
Bounding Box Coordinate 1 : (8 , 48)
Bounding Box Coordinate 2 : (287 , 48)
Bounding Box Coordinate 3 : (8 , 259)
Bounding Box Coordinate 4 : (287 , 259)

For an image with 2 animals, eg: sheep and dog:
'Sheep':
Bounding Box Coordinate 1 : (45 , 51)
Bounding Box Coordinate 2 : (198 , 51)
Bounding Box Coordinate 3 : (45 , 216)
Bounding Box Coordinate 4 : (198 , 216)

'Dog':
Bounding Box Coordinate 1 : (217 , 126)
Bounding Box Coordinate 2 : (249 , 126)
Bounding Box Coordinate 3 : (217 , 154)
Bounding Box Coordinate 4 : (249 , 154)

5 Applications and Conclusion

In conclusion we can say that out output satisfies the expectations of our problem statement. It detects the animal in the image, provides it's location, it also provides the likelihood of it's detection being true effectively.
The bounding boxes and it's co-ordinates obtained can essentially be very helpful in certain number of ways:
One can not only spot animals via surveillance, but easy location detection can also be possible. One can inform about the co-ordinates of an animal lost in it's habitat when carrying on a drone search, one can spot animals being poached and also spot the person responsible for dong so using surveillance systems. These are a few of the many uses of this automation.
The detection of animals can also be useful to keep a count of the number of animals for a particular species in a habitat. Annual surveys can be carried out automatically without human involvement or only a little assistance. That would make conservation of bio-diversity easier.

6 References

1. Redmon, Joseph, and Ali Farhadi. "YOLOv3: An Incremental Improvement." ArXiv.org. April 08, 2018. Accessed October 02, 2021. https://arxiv.org/abs/1804.02767.

2. Banupriya, N., S. Saranya, Rashmi Jayakumar, Rashmi Swaminathan, Sanchithaa Harikumar, and Sukhita Palanisamy. "Animal Detection Using Deep Learning Algorithms." January 15, 2020. http://www.jcreview.com/fulltext/JCR070185.pdf

3. A. Gomez., A. Salazar, F. Vargas, towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks, 2016.

4. What Is YOLO Algorithm? | Deep Learning Tutorial 31 (Tensorflow, Keras &amp; Python). Youtube. code basics, 2020. https://www.youtube.com/watch?v=ag3DLKsl2vk.

5. Norouzzadeh, Mohammad Sadegh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S. Palmer, Craig Packer, and Jeff Clune. "Automatically Identifying, Counting, and Describing Wild Animals in Camera-trap Images with Deep Learning." PNAS. June 19, 2018. Accessed October 03, 2021. https://www.pnas.org/content/115/25/E5716.

6. https://www.kaggle.com/valentynsichkar/yolo-coco-data

7. https://www.kaggle.com/alessiocorrado99/animals10