Object detection: YOLOv3 on African Wildlife dataset

Emma Rydholm, Johanna Warnqvist

Chalmers University of Technology
Electrical Engineering Department

Introduction

Goal: In this project, our ambition was to implement YOLOv3 on a dataset with images of African Wildlife [1].

Figure 1: Example image from dataset

Our motivation: Get experience in the hot topic of object detection. Also, to gain experience in using code written by others, since this will be common in the real working life.

Key components:

- **Dataset:** 1514 images of Zebra, Elephant, Buffalo and Rhino
- **Model:** YOLOv3 with pre-trained Darknet-53 as backbone
- **Augmentations:**
  - Horizontal flip
  - Vertical flip
  - Color Jitter
  - Random cutout

Method

- Our implementation of YOLO is based on a pre-existing implementation of YOLOv3 [2], with important adjustments.
- The backbone was pre-trained on ImageNet and our network was then re-trained on the African Wildlife dataset with an 80/20 split.
- The layers where adjusted to detect only 4 classes.
- Augmentation methods where added.
- The model was trained for 60 epochs twice, with/without augmentation.
- Average precision, loss, F1-score and some other metrics where plotted after 60 epochs.

Theory

- **The YOLO approach:** one single neural network
- Predicts bounding boxes and class probabilities from a single forward pass of the image through the network
- **Faster** than other object detection methods

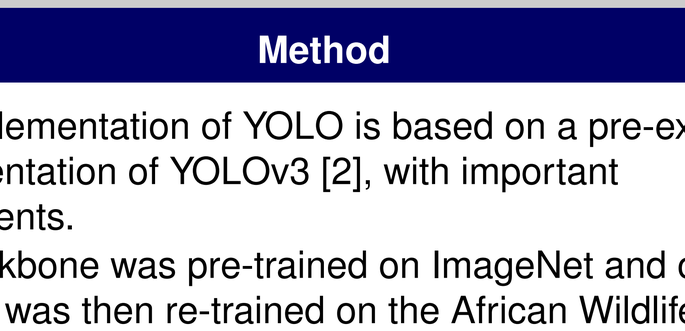

Figure 2: The YOLO architecture [3]

- **Strategy:**
  1. Divide image into SxS grid
  2. Each cell predicts a fixed number of bounding boxes and its associated confidence score and class probability
  3. If two bounding boxes has a high overlap, discard the one with lowest confidence score (non-max-suppression)
  4. All bounding boxes below a certain confidence score threshold should be removed
- **YOLOv3** is larger than YOLO and it predicts bounding boxes at three different places and at 3 different **scales**, meaning that we have 3 ”YOLO-layers”, instead of one [4].

Results

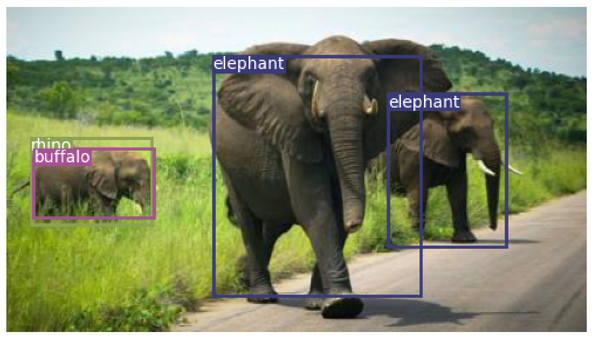

Figure 3: Detected elephants and wrong label on baby elephants

Can the model detect the animals?

- Even our best model is still not perfect
- The max mAP with augmentation is 0.7252 (epoch 57) compared to 0.67648 (epoch 58) without augmentation.
- The model had not yet fully converged (due to time shortage).

Average Precision (AP) is the area under the precision-recall function:

$$AP = \int_0^1 p(r)\,\mathrm{d}r$$

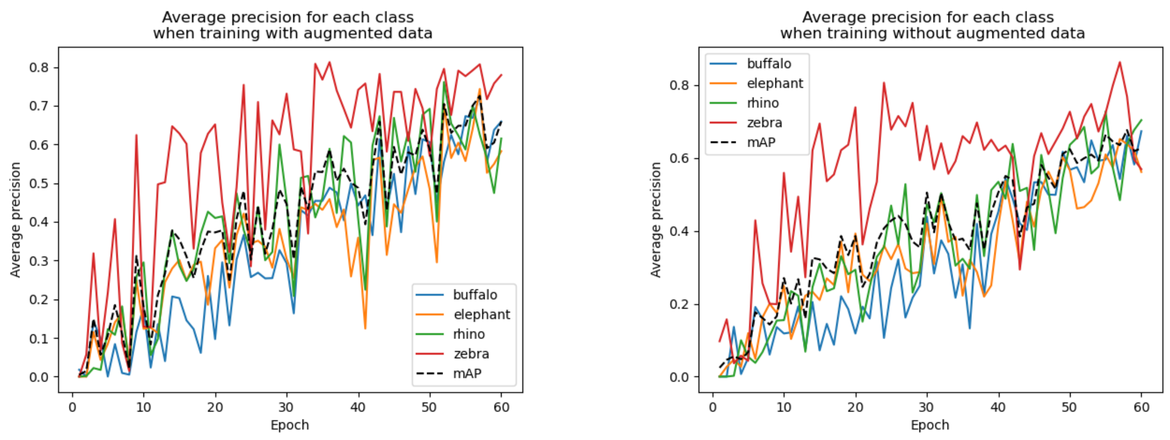

Figure 4: Mean Average Precision (mAP) and Average Precision (AP) for each class.

Detection performance over time

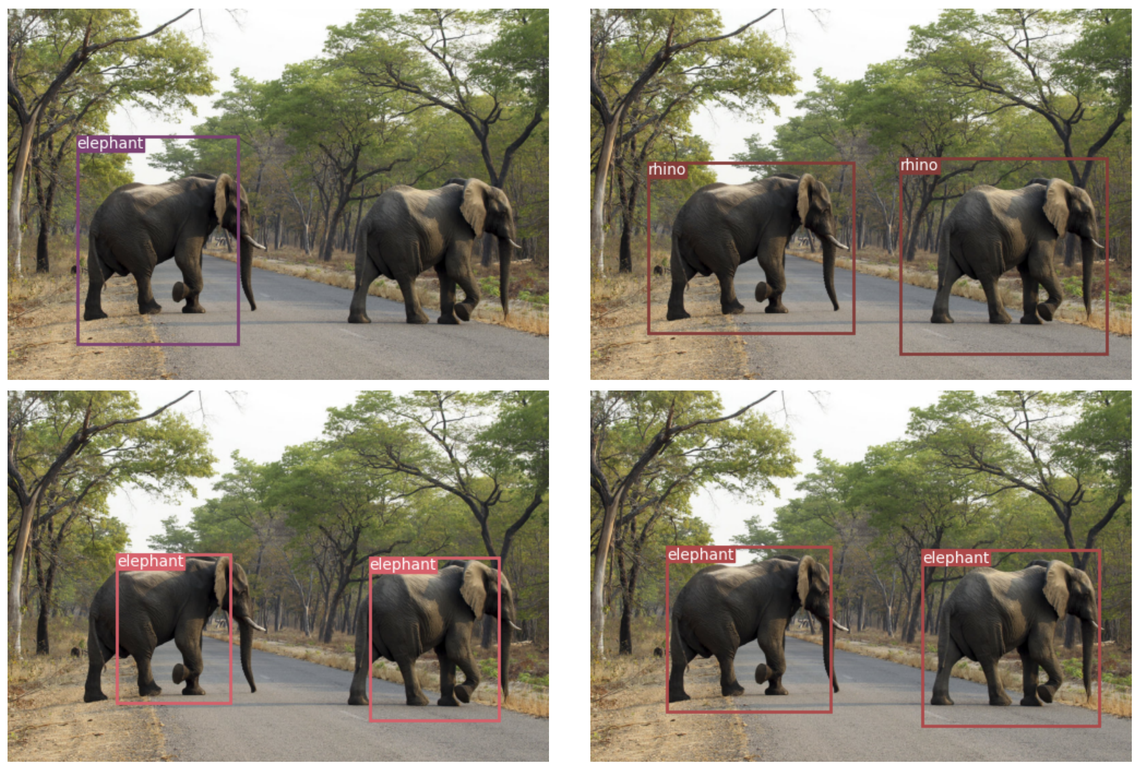

Figure 5: Detection at epoch 19, 29, 39 and 59.

Conclusion

- We have explored object detection using YOLOv3 and obtained a far from perfect model, but a model that often recognizes our objects of interest.
- A one week project is not enough to fully investigate the full potential of the YOLOv3 algorithm, if we had more time we would have done hyperparameter search, looked deeper into data augmentation, trained for longer etc.

References

[1] B. Ferreira.
African wildlife.
https://www.kaggle.com/biancaferreira/african-wildlife, 2020.

[2] E. Linder Noren.
Pytorch-yolov3.
https://github.com/eriklindernoren/PyTorch-YOLOv3, 2019.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi.
You only look once: Unified, real-time object detection, 2016.

[4] J. Redmon and A. Farhadi.
Yolov3: An incremental improvement.
arXiv, 2018.

emmary@student.chalmers.se johwar@student.chalmers.se