02456: Real-time object detection and tracking of beer & cola cans

s173901 Andreas Grønbech Petersen[1], s173941 Philip Hoedt Karstensen [2], s182521 Holger Christian Nyeland Ehlers[2], and s173952 Jonas Brøndum[1]

1 DTU Electric, Technical University of Denmark; 2 DTU Health Tech, Technical University of Denmark

Introduction

Detecting and tracking objects in real-time is still a rather demanding task on computationally limited systems. In recent years the speed of object detection networks have become faster and faster given more data and better GPUs to train with. For various industrial applications a package of such a machine and a video camera could be the replacement of an on-site human eye and so a more convenient solution on smaller computers might be desirable. We take advantage of progress in deep learning, by using existing state-of-the-art convolutional neural networks (CNNs) to propose an end-to-end fast object detection and tracking model.

Key points

- We use existing neural network architectures to further develop and infer on a new data set created from rolling beer and cola cans on a hardwood floor.
  - Single Shot Detection (SSD) [1]
  - Faster R-CNN network architecture [2].
  - Different feature extractor networks
- We perform pre- and post-processing of data
- We implement a near real-time object detection and tracking algorithm
- We compare the performance of different backbone networks in the context of real time inference

General approach and methods

We collected video of beer (Grøn Tuborg) and cola (Coca Cola) cans as the data for both training and testing. The data set consists of 3389 frames from two video recordings.
The video was partitioned into frames that were manually labelled with boxes according to presence of cans. These annotations are processed, such that frames without annotations are disregarded to decrease subsequent processing time. Followingly the processed data can be fed into either a Faster R-CNN network with a variation of backbones (ResNet50, MobileNetV3, etc.) or a Single-Shot-Detection CNN (YOLOv5s).
This network is then trained and tested on a randomized 80/20 partition of the data set. Subsequently the performance of the network is evaluated on a video for its accuracy and real-time performance both on CPU and GPU to conclude on its industrial feasibility.

Model

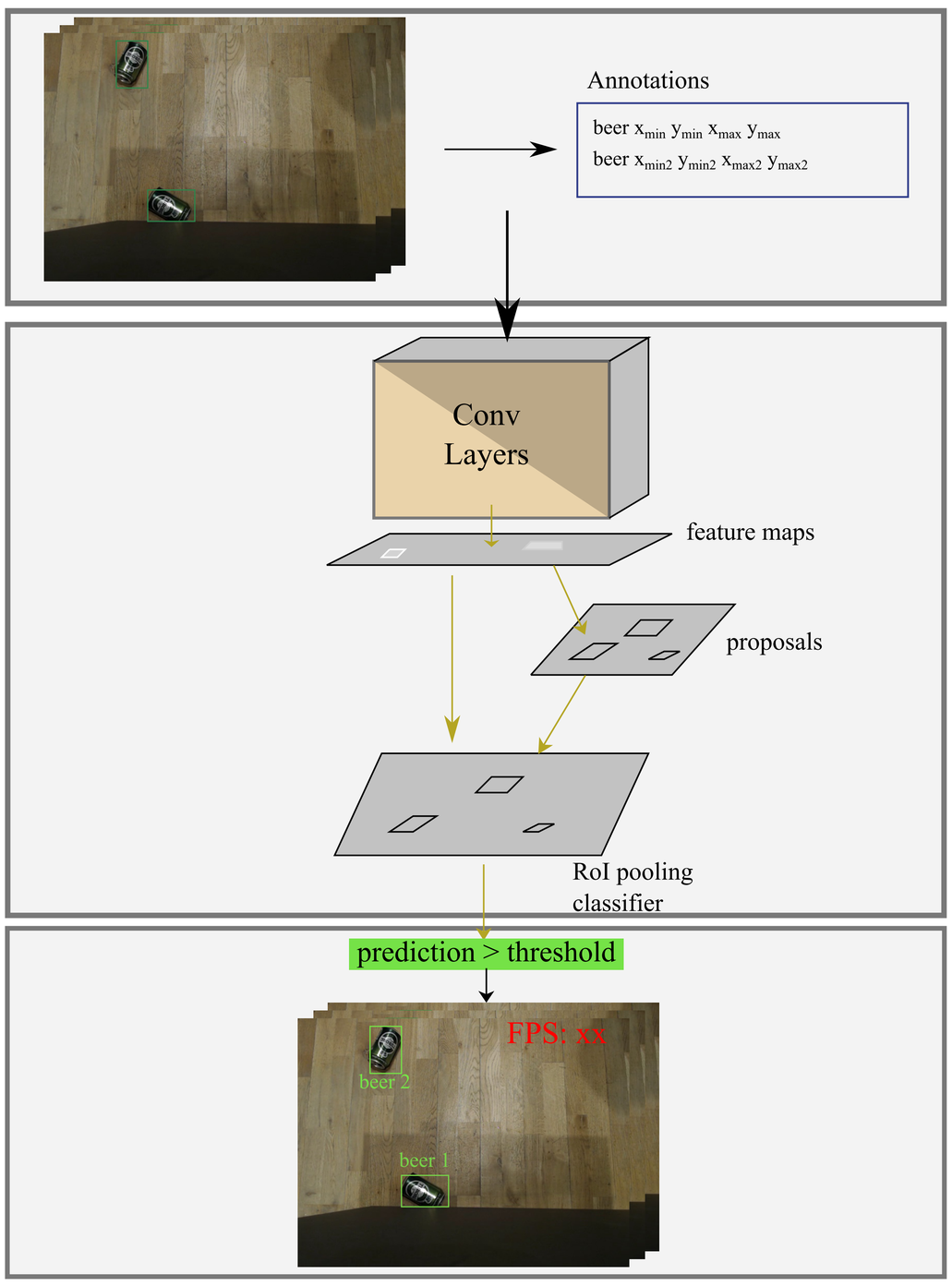

Figure 1: Faster RCNN model architecture representation.

Inference and accuracy

The accuracy and inference time of the different CNN models used for training on the entire dataset. As seen from both fig. 2 and table 1, YOLOv5s performs the best overall with a compromise on both accuracy and inference time.
On the other spectrum is the ResNet50-network which lags behind in inference time performance, since it's generally a larger network.

Table 1: Accuracy and inference time achieved by pre-trained CNN models

Backbone	Accuracy (mAP)	Inference Time [s]
MobileNetv3Large	87.4%	0.0874
MobileNetv3Large_320	27.3%	0.0495
ResNet50	86.8%	0.3238
YOLOv5s	85.1%	0.0333

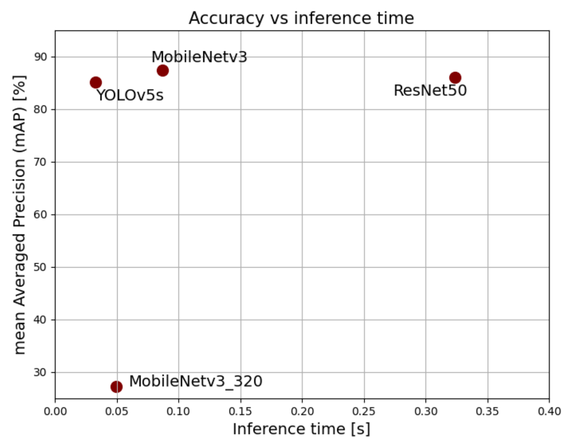

Figure 2: Accuracy vs inference time for the different trained models.

Model performances

Table 2: Processing performance, both averaged and interval, of the different models using either CPU or GPU. The processing is performed on a separate test video with various types of cans: Grøn Tuborg, Coca Cola, Heineken, Royal Eksport and Fanta. $[FPS]$ = minimal and maximum FPS captured, $\overline{FPS}$ = average FPS for entire video and t = elapsed time for tracking of entire video.
The CPU tests were run an Intel i5-4460K
The GPU tests were run on a Nvidia GTX 1050Ti

Model	$\overline{FPS_{CPU}}$	$[FPS_{CPU}]$	t[s]	$\overline{FPS_{GPU}}$	$[FPS_{GPU}]$	t[s]
MobileNetv3Large	1.34	0.76 - 1.55	933.25	9.88	0.39 - 14.21	126.95
MobileNetv3Large_320	5.5	1.63 - 7.08	228.16	19.57	0.39 - 29.59	64.08
ResNet50	0.23	0.12 - 0.24	5416.15	2.17	0.05 - 2.65	577.29

Object tracking

The proposed algorithm uses centroid tracking to maintain an overview of the current objects in a given frame. Centroid tracking is an easy way to implement object tracking, while not being the fastest or most reliable. For objects that have intersecting trajectories with near contact the centroid tracking algorithm can fail to detect the objects correctly as it uses euclidean distances.

Discussion and Conclusion

Limitations of the project

Relatively complicated networks that were not easily adjustable which meant that models achieved sub optimal performance. The limitations include

- Quantization
- Tracking algorithms

Future Directions

Implement quantization, testing alternative tracking algorithms as well as testing the YOLOv5s in more depth.

Acknowledgements

Thanks to Peter Jensen for supervision and offering us the opportunity to work with this challenging and exciting project.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. 2015. URL http://arxiv.org/abs/1506.02640.

[2] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. URL http://arxiv.org/abs/1506.01497.