Stanford CS230: Object Detection in Pytorch

Koen Frankhuizen, CS 230 Final Project; Stanford University; youtube link: https://youtu.be/Qcno_unfl0E

OVERVIEW

Motivation: This implementation enables us to train popular object detection & classification models on an existing, large, dataset and evaluate their performance on a dataset that reflects real-world images. Object detection and classification is an essential pre-processing step before feeding a Siamese network ([1]). This project will focus on the detection and classification and compare the performance of a YoloV3 and SSD model.

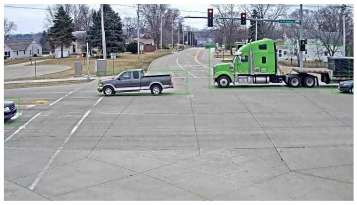
Summary: This project focusses on object detection and object classification, based on the AI City Challenge 2019 dataset. 400+ images were manually relabelled and used for training. The 2018 results are chosen as benchmark. YoloV3 and SSD model performance was compared - with a 0.67 IDF1 score for the YoloV3 model and a 0.56 IDF1 score for the SSD model.

DATA, PRE-PROCESSING AND FEATURES

Data from the AI city challenge.

- 3.25 hours of videos from 40 cameras across 10 intersections
- Three intersections selected for this project
- 400+ images hand labeled to improve train data quality

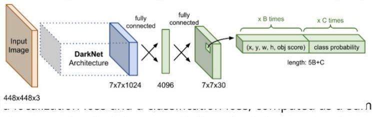
-Example raw images from each of the three scenes



- Hand labeled bounding boxes and classification added (car, van, pickup-truck, truck, and bus).
- Each of this cars will be, if known labeled with a type (in this example: [?, pick-up,]

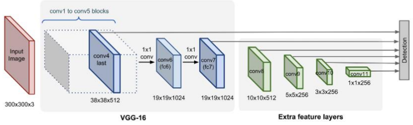
MODELS; INDIVIDUAL DISCUSSION

YoloV3: A YoloV3 was implemented using three different architectures (320, 416, 608):



Loss function: a localization loss and a classification loss, computed as a sum of squared errors with scale parameters to balance prediction of objects / no objects.

SSD: a Pytorch based SSD implementation was build for the purpose of this project. The implementation supports multiple input formats.



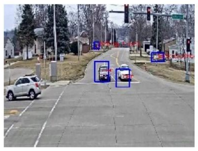
Loss function: SSD also uses two separate parts for the loss function. a L1 loss for the localization loss and softmax loss (over the multiple classes for the classification loss.

Hyper parameter tuning: After applying (1) random search and (2) grid search, I found the following optimal parameters:

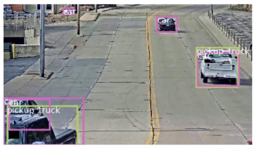
Model	Number of epochs	Input size	Learning rate
SSD-baseline	20	224	1e-4
SSD-optimized	60	608	1e-4
YoloV3-optimzed	26	416	1e-4

RESULTS (1): OUTPUT BOXES

Both the SSD and YoloV3 performed reasonable well on large vehicles, but not so good on small vehicles, where often no bounding box or low quality bounding box was presented. (figures have been cropped for clarity - original figures in report). YoloV3 overall captured objects better and more accurate .



SSD detection output



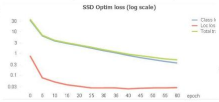
YoloV3 detection output

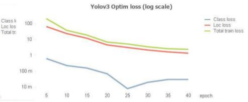
RESULTS (2): IDF1/MAP SCORE

The YoloV3 model, after hyperparameter tuning, performed better then the SSD score, in particular for the mAP score:

Model	IDF1 score (test)	mAP score (test)	IDF1 score (train)
SSD-baseline	0.44	0.03	n/a
SSD-optimized	0.56	0.22	0.62
YoloV3-optimzed	0.50	0.48	0.9

Both models converged well, shown by the loss functions (log scale) versus nr of epochs:





DISCUSSION AND FUTURE WORK

Conclusion:

- The goal was to compare SSD with YoloV3. The YoloV3 performed better.
- Overall I did not manage to 'beat' the AI City challenge baseline set by the three used models (YoloV3, SSD and Faster-RCNN). Most likely this is because my training set was much to small – a training set of 400 images is small for Computer Vision problems.
- Winning 2018 benchmark of AI City challenge scored a F1 score of 0.86, but did a lot of relabelling and data augmentation
- For the limited scope and dataset, results are reasonable good.

Future work

- Low SSD label performance should be investigated
- The YoloV3 model showed most promising results to continue with.
- A larger training set is essential to meet benchmark (AI City Challenge 2019) standards

REFERENCES

-[1] **Xinchen Liu, Wu Liu(B), Tao Mei, and Huadong Ma,** A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance

[2] SSD: **Wei Liu, Dragomir Anguelov, Dumitru Erhan,** C. zegedy. Single Shot MultiBox Detector.

[3] **Joseph Redmon, Ali Farhadi** "YOLOv3: An Incremental Improvement", 2018,

Further references see main report