doi:10.3311/BMEZalaZONE2022-002

Survey on Image Based Object Detectors

Márton Cserni[1a], András Rövid[1]

[1] Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics

[a] Corresponding author: mcserni@edu.bme.hu

Abstract

Sensor fusion-based detector utilizes camera sensors to solve the problem of recognizing objects and their classifications accurately. This has been proven to increase accuracy compared to single sensor detectors and can significantly help with the 3D tracking of vehicles in the sensor system's area of interest. Even at a distance, where no lidar points are available from the target, a high-resolution camera-based detector can easily detect and classify vehicles. There is a variety of real-time capable 2D object detector convolutional neural networks, some of which are open source. This survey compiles a list of these algorithms, comparing them by precision scores on well-known datasets, and based on experimental evaluation completed on camera images taken on the ZalaZONE test-track to evaluate the distances at which the detectors first perceived the test vehicles. Additionally, inference times are also compared.

Keywords: *2D object detection, autonomous driving, camera, ZalaZONE*

1 Introduction

1.1 Context

Autonomous driving and advanced driver assistant systems (ADAS) can significantly reduce the difficulty of the complicated task of driving, and help eliminate human error, ultimately leading to a significant reduction in road traffic accidents. One such ADAS example would be the Autonomous Emergency Braking System (AEBS). In the event, where a dangerous situation arises, for example a child runs out into the road in front of the vehicle equipped with AEBS, the system shall mitigate or avoid a collision by recognizing the dangerous situation, and applying the brakes autonomously. Recently, AEBS is tested by Euro NCAP as a standard part of the safety rating [1].

A key part of any such system is an informative, perceptive and robust environment perception system. One of the most difficult, and hotly researched task in automotive environmental perception is object detection. Object detection consists of the recognition, classification and localization of relevant objects in the relevant environment of the perception system. Within object detection, 2D and 3D object detection have to be differentiated. 2D object detection outputs a two dimensional bounding box (usually) in images, in the image coordinate system. The bounding box is usually a rectangle given by the top left, and bottom right pixel points, or given by the centre point, and width and height of the box in pixels. 3D object detection is focused on outputting the 3D bounding boxes of objects in the sensor's (eventually a global) coordinate system. The 3D bounding box is given by the centre point, and width, height and length values in metres, and the heading angles (roll, pitch and yaw) of the object.

Object detection is a fundamental problem in computer vision. Camera-based detectors excel in 2D object detection, but lag behind in 3D accuracy compared to other sensors. This paper will focus on the state of the art (SOTA) in image based 2D object detection.

2 Image based object detection

The task of this paper is to compile a list of the SOTA of image-based 2D object detector algorithms in early 2022. The main focus will be on object detection, but an instance segmentation algorithm is also worth mentioning.

Image based object detection is usually approached with deep learning. These, contrarily to classical methods can produce robust and accurate detections regardless of the scenario in the image. Within the category of deep learning based methods, there exist single-stage and two stage methods. Two stage algorithms usually first create region proposals in the image, in which objects are likely, and input the regions into a second network, which regresses the exact bounding box, and performs classifications. The most notable two-stage algorithms are the R-CNN algorithm family. Two-stage methods usually are more accurate, but slower, making them hard to compute in real-time. Hence, they are not used in autonomous driving applications. Single stage algorithms (such as YOLO, SSD, and RetinaNet) on the other hand can run in real-time, but lack accuracy in certain conditions, such as groups of small objects or irregular objects.

This paper uses the Microsoft COCO (Common Objects in Context) dataset as a reference for evaluating state of the art object detectors [2].

2.1 Two-stage detectors

The most important two-stage detectors are the R-CNNs. R-CNN stands for Regions with Convolutional Neural Network features, as published in 2014 in [3]. The principle of most two-stage detectors, specifically R-CNN is shown in Fig. 1. The network extracts 2000 region proposals from the image (likely containing relevant objects), then for each proposal, a CNN computes image features, which are finally classified by an SVM.

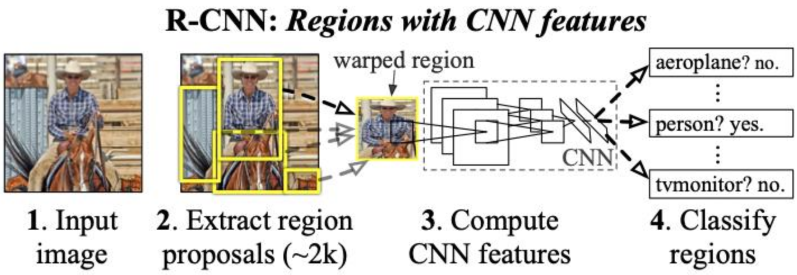

Fig. 1 *R-CNN architecture [3]*

The problem with R-CNN was that it was slow. Since 2014, the algorithm was improved in Fast-RCNN and Faster-RCNN in [4], and [5] making it much faster. Mask R-CNN was presented in 2017, implementing an instance segmentation (pixel level classification) solution alongside the object detector with minimal overhead [6]. The latest advancement is G-RCNN or Granulated R-CNN, which introduces spatio-temporal information incorporation into Fast and Faster R-CNN which improves detection speed and accuracy on videos, or image sequences [7].

2.2 Single-stage detectors

One-stage detectors focus on real-time applicability. The most popular of these deep learning models are the YOLO algorithms, the SSD, and RetinaNet. The original YOLO paper was published in [8], it's working principle can be seen in Fig. 2. Yolo predicts bounding boxes by using a single neural network by dividing the input image into a grid. Each grid cell predicts a number of bounding boxes (anchors) alongside an objectness score (the probability of an object being present in the grid cell. The objects are filtered based on this objectness score, and the final bounding box is regressed as a distance from the anchor box with the highest objectness score.

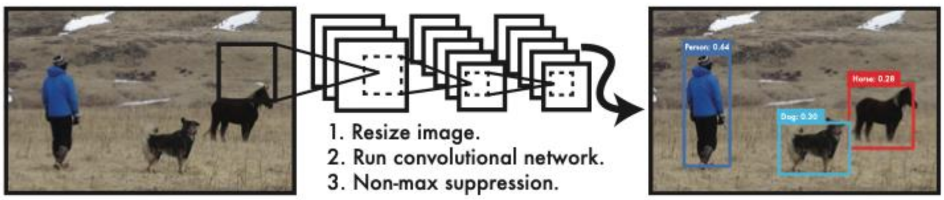

Fig. 2 *Yolo principle [8]*

Since 2016, YOLO has been updated and improved. The most significant results are Yolov3 [9], Yolov4 [10], and recently YOLOR [11]. Yolov3 improved the detection of small objects by introducing upscaling into the fully convolutional neural network model. Yolov4 presented an optimal architecture for speed and accuracy in object detection. YOLOR (You Only Learn One Representation) is a very recent discovery. The authors found a way to incorporate explicit and implicit learning into the multi-task model training, achieving increased accuracy and significant detection speed increase (88% over the scaled Yolov4).

The current state of the art in object detection on the MS Coco dataset is the Swin Transformer V2 [12]. This model, alongside the M-RCNN is an object detector and an instance segmentator.

Yolov5 is also worth mentioning [13]. This is an open-source Pytorch implementation of object detector models (based heavily on Yolov4) as implemented by Ultralytics. A simple Pytorch Hub API call is enough to download a variety of models ranging from the Yolov5n to the Yolov5x, which are all pretrained on the MS Coco dataset. The models differ in size, thus inference speed and detection accuracy.

2.3 Metric and Comparison

The most important metrics for object detectors are the inference time, and the accuracy. Inference time depends heavily on hardware and implementation efficiency. As for the accuracy, the most widely accepted metric is the AP or average precision. The latest papers publicise the COCO mAP metric of their detectors. This is explained fully in [14] and on the COCO website [15]. This metric computes the average precision of detectors with vairable Intersection over Union threshold value ranging from 0.5 to 0.95 with a step size of 0.05.

Table 1 contains the most relevant detectors and their mAP values. Inference speeds are hard to compare due to implementation differences, however there is more on this topic in the experimental evaluation section.

Table 1 *Model Coco mAP values [16]*

Name	**Coco mAP**
Swinv2-G	**63.1**
YOLOR-D6	57.3
Scaled Yolov4	55.8
Yolov5x (largest Yolov5 model)	55.4
Yolov5n (smallest Yolov5 model)	34.0
Mask-RCNN	46.1
Faster-RCNN	37.4

3 Experimental Evaluation

In this section an experimental evaluation is presented of the online available implementations of some of the state of the art 2D object detectors.

3.1 Scenario

The test scenario was recorded on the ZalaZONE test track, in Zalaegerszeg, Hungary. In the test scenario, four test vehicles, equipped with high precision GPS sensors were stationed approximately 20-50m apart, and asked to perform a Y turn, then drive away from the sensor. The camera used was a Flir PointGrey camera with 2MP image sensor and a 60° HFOV. The relatively low resolution of this camera affects network performance negatively in the case of targets at a longer distance. The detectors were evaluated based on the greatest distance they detected the retreating vehicles from, and their GPU usage and inference times were recorded. The evaluation was performed on a Nvidia RTX 2060 Super with a maximum of 175W power, 7982MiB memory and 7.2 Tflops of computing power.

3.2 Results

Firstly, the most easily accessable Yolov5 models were evaluated. These models are implemented in Pytorch, using the Tensor Cores of the GPU meaning this is a very efficient implementation. The results are surprising, as it turns out that the optimal model is the Yolov5l, which works better in this scenario compared to the larger Yolov5x. Next, an implementation of M-RCNN is evaluated as a comparison. Keep in mind, that M-RCNN precedes the Yolov5 with 4 years. The Gluon model-zoo implementation [17] uses MXNet, and the model inference also runs on the GPU. Table 2 compares the results of the Yolov5 models and the M-Rcnn. Stable distance refers to every frame being detected, and beyond that until furthest distance, the detector is noisy up to Furthest Distance, where the vehicle is lost.

Table 2 The results of the experimental evaluation

Name	Inference time [s]	GPU Power [W]	GPU Memory [MiB]	Stable Distance [m]	Furthest Detection [m]
Yolov5n	0.0084	46	1110	90	99
Yolov5s	0.0086	50	1144	118	124
Yolov5m	0.0129	55	1236	135	149
Yolov5l	**0.0153**	**111**	**1436**	**164**	**223**
Yolov5x	0.0234	130	1794	151	218
M-Rcnn	0.65	175	6065	100	120

4 Summary

In this paper, the state of the art of 2D object detection was briefly presented, along with an experimental evaluation conducted in a real-world scenario on ZalaZONE. It can be concluded, that Yolov5l is the presently most applicable implementation, however promising new results recently became available. Future research will include the evaluation of Yolor and Swinv2-G, as their code is also available online.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

[1] European New Car Assessment Programme, "Euro NCAP Rating Review 2018," June 2020. [Online]. Available: https://cdn.euroncap.com/media/58242/euro-ncap-rating-review-2018-version-12-with-appendices.pdf. [Accessed 12 02 2022].

[2] T.-Y. Lin and e. al, ""Microsoft coco: Common objects in context.," European conference on computer vision., 2014.

[3] R. Girschick, J. Donahue, T. Darell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580--587, 2014.

[4] R. Girshick, "Fast R-CNN," Proceedings of the IEEE international conference on computer vision, pp. 1440-1448, 2015.

[5] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems,* no. 28, 2015.

[6] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," Proceedings of the IEEE international conference on computer vision, pp. 2961-2969, 2017.

[7] A. Pramanik, S. K. Pal and P. Mitra, "Advances in neural information processing systems," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 6, no. 1, pp. 171-181, 2021.

[8] J. Redmond, "You only look once: Unified, real-time object detection," IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.

[9] J. Redmond and A. Fahradi, "Yolov3: An incremental improvement.," arXiv preprint , no. arXiv:1804.02767, 2018.

[10] A. Bochkovskiy, W. Chien-Yao and M. L. Hong-Yuan, "Yolov4: Optimal speed and accuracy of object detection.," arXiv preprint, no. arXiv:2004.10934, 2020.

[11] C.-Y. Wang, I.-H. Yeh and L. M. Hong-Yuan, "You only learn one representation: Unified network for multiple tasks.," arXiv preprint , no. arXiv:2105.04206, 2021.

[12] Z. Liu and e. al, "Swin Transformer V2: Scaling Up Capacity and Resolution," arXiv preprint, no. arXiv:2111.09883., 2021.

[13] Ultralytics, "github.com," 2022. [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed 13 02 2022].

[14] J. Hui, "medium.com," 2018. [Online]. Available: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173. [Accessed 13 02 2022].
[15] "Coco Detection Leaderboard," [Online]. Available: https://cocodataset.org/#detection-leaderboard. [Accessed 13 03 2022].
[16] "Papers With Code," [Online]. Available: https://paperswithcode.com/sota/object-detection-on-coco. [Accessed 13 02 2022].
[17] Gluon, "Model zoo," [Online]. Available: https://cv.gluon.ai/model_zoo/segmentation.html. [Accessed 13 02 2022].