Object Detection for Self-Driving Cars

Tina Hajinejad THAJINEJ@UWATERLOO.CA
Department of Mathematics
University of Waterloo
Waterloo, ON N2L 3G1, Canada

Editor: -

Abstract

The development of self-driving cars has become one of the most exciting and challenging research areas in autonomous vehicles. Object detection, which is the task of identifying and localizing objects appearing in an image or video, plays a crucial role in enabling self-driving cars to perceive their surroundings and make intelligent decisions. However, achieving high levels of accuracy and speed in object detection remains a significant challenge. In this review, we aim to explore some of the proposed techniques that have been developed to improve object detection for self-driving cars. We will assess the various deep learning models, and also highlight some of the advancements and limitations in this field. This review aims to provide insights into the current state-of-the-art object detection for self-driving cars and identify promising avenues for future research.

Keywords: Object Detection, Self-Driving Cars, Perception, Deep Learning techniques

1. Introduction

Over the past few years, there has been a considerable surge in research focus on autonomous vehicles(AVs), which are automobile platforms designed to perceive and respond to their surroundings, without human intervention. This process of environmental perception is commonly referred to as perception and involves several sub-tasks such as object detection, classification, 3D position estimation, and simultaneous localization and mapping (SLAM)[1].

Object detection has long been considered one of the most interesting yet challenging tasks for environment perception, as it plays a critical role in ensuring the safe and reliable navigation of autonomous vehicles. While humans are able to perform object detection while driving quickly and with minimal errors, autonomous vehicles struggle with this task and require significant improvement. They must be able to recognize objects in real-time with a high degree of accuracy. This requires not only the ability to recognize and categorize every object in an image but also the skill to localize each one accurately by drawing a bounding box around it[2].

In the upcoming sections, we examine the fundamental techniques that underpin many of the latest methods in object detection. We assess their architecture, accuracy, speed, and overall performance, showcasing both their benefits and drawbacks. This analysis allows us to gain a better understanding of the cutting-edge in object detection and appreciate the remarkable progress made in this field over recent times.

2. Techniques

Several techniques for object detection using deep learning are proposed through the years, such as R-CNN, Fast R-CNN, Faster R-CNN, You Only Look Once (YOLO), Single Shot Detection (SSD), EfficientDet, and SPPnet. All of these techniques use Convolutational Neural Networks as their primary structure. We will discuss them more ahead based on architecture, speed in detection, and precision.

2.1 You Only Look Once

You Only Look Once[3], or for short, YOLO, is inpired by GoogLeNet for classification. They frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities.This network has 24 convolutional layers, followed by two fully connected layers, and simply use 1x1 reduction layers followed by 3x3 convolutional layers. The final layer predicts both class probabilities and and the final output of the network is the 7x7x30 tensor of predictions. Finally they use Mean Squared Error as Loss function for ease.

2.2 EfficientDet

EfficientDet[4] is a state-of-the-art object detection model that was introduced in 2020. It is based on the EfficientNet architecture, which is a family of models that have achieved state-of-the-art performance on image classification tasks while being computationally efficient. EfficientDet builds upon this architecture by incorporating a novel bi-directional feature pyramid network and an efficient object detection head.

2.3 Regions with CNN features

R-CNN [5], short for Regions with CNN features, has made room for advancement of better networks such as Fast R-CNN[6] and Faster R-CNN[7]. The object detection system presented has three main components. The first component generates region proposals, which are candidate detections for objects in the image. The second component is a convolutional neural network that extracts feature vectors from each region(consisting of five convolutional layers and two fully connected layers), and the third component is a set of linear SVMs that classify the regions into specific object categories. In essence, they have turned object detection into an image classification problem. R-CNN was very intuitive but very slow[2]. R-CNN is slow because it performs a ConvNet forward pass for each object proposal, without sharing computation [6].

2.4 Fast R-CNN and Faster R-CNN

A Fast R-CNN[6] network requires an image and a set of object proposals as input. The image is first processed with several convolutional and max pooling layers to produce a convolutional feature map. For each object proposal, a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map. These feature vectors are then passed through a sequence of fully connected layers that eventually branch into two output layers. One layer produces softmax probability estimates over K object classes,

including a "background" class, while the other outputs four real valued numbers for each of the K classes. Each set of four values encodes refined bounding-box positions for one of the K classes. The main insight of Faster R-CNN[7] was to replace the slow selective search algorithm with a fast neural net. Specifically, it introduced the region proposal network (RPN). In a sense, Faster R-CNN = RPN + Fast R-CNN [2].

2.5 SPPnet

SPPnet[8] is an extension of the Region-based Convolutional Neural Network (R-CNN) architecture, which was used for object detection in images. SPPnet addresses the issue of variable input image sizes. Conventional cropping, and warping both cause unwanted issues such as losing a part of the image, or geometric distortion. SPPnet solves this by introducing a spatial pyramid pooling layer that can pool features from images of different sizes and scales. This allows the network to have a fixed-sized output vector, which can then be fed into a classifier for object detection.

The key idea behind SPPnet is to divide the input image into a pyramid of fixed-size grids, and then pool the features inside each grid using max pooling. The resulting feature maps are then concatenated into a single feature vector, which can be used as input to a classifier. By using spatial pyramid pooling, SPPnet can handle images of different sizes and aspect ratios, while still maintaining the spatial information of the image.

2.6 Single Shot Detection

Finally, The Single Shot Detection (SSD)[9] architecture consists of a base convolutional neural network (CNN) followed by a set of auxiliary convolutional layers that are responsible for detecting objects at multiple scales. These auxiliary layers are connected to different layers in the base CNN, allowing the detection of objects at different spatial resolutions. The final output of the network is a set of predicted bounding boxes, each with a corresponding class confidence score.

3. Empirical Evaluation

In order to compare different object detection techniques, we can evaluate them based on several aspects such as precision, speed, and complexity. We have already discussed the various architectures and their complexities in the previous section. In this section, we will introduce metrics to assess the performance of different methods. To evaluate speed, we will compare the time taken to detect objects in a single image. To assess precision in detection, we will use the mean Average Precision (mAP) score.

The mAP (mean Average Precision) score is a commonly used performance metric for object detection tasks. It is defined as the average of the AP (Average Precision) scores for each class. AP is a measure of how well the model is able to distinguish between positive and negative examples, and is calculated as the area under the Precision-Recall curve. The mAP score is used to evaluate the overall performance of an object detection model across multiple object classes. A higher mAP score indicates better performance.[1] We will compare the mAP score for all the techniques and comment on their precision.

Almost all of the methods were trained and evaulated on PASCAL VOC. Fast YOLO, is a variation of YOLO that uses a neural network with fewer convolutional layers (9 instead of 24) and fewer filters in those layers. Other than network size, these two models are the same. Fast YOLO is the fastest object detection method on PASCAL yet; as far as we know, it is the fastest extant object detector with an mAP score of 52.7 percent, while "YOLO pushes mAP to 63.4 percent, while still maintaining real-time performance"[3].

R-CNN achieved a mean average precision (mAP) of 53.7 percent on PASCAL VOC 2010, compared to a previous work reporting 35.1 percent mAP[5]. They also assessed all design choices and hyperparameters using the VOC 2007 dataset. To obtain their final results on the VOC 2010-12 datasets, they fine-tuned the CNN using the VOC 2012 training set and optimized their detection SVMs using the VOC 2012 trainval dataset. This paper goes on to compare their model with four strong baselines. According to this paper, the UVA system proposed by Uijlings et al. [5], which utilizes the same region proposal algorithm, only achieves 35.1 percent mAP, whereas R-CNN has a mAP score of 53.7 percent on PASCAL VOC 2010, while also being much faster. They both acheive an mAP of 53.3 percent on the PASCAL VOC 2011-12 test.[5]

To improve the efficiency of R-CNN, the authors of Fast R-CNN proposed Spatial Pyramid Pooling Networks (SPPnets[8]). The main idea behind SPPnets is to enable the sharing of computation, thereby reducing processing time. SPPnet accelerates R-CNN by 10-100 times at test time. Training time is also three times less due to faster proposal feature extraction. Compared to SPPnet, Fast R-CNN trains VGG16 three times faster, tests ten times faster, and is more accurate.[6]

Real-Time Detectors	Train	mAP	FPS
100Hz DPM	2007	16.0	100
30Hz DPM	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM	2007	30.4	15
R-CNN Minus R	2007	53.5	6
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN VGG-16	2007+2012	73.2	7
Faster R-CNN ZF	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Figure 1: Real-Time systems on PASCAL 2007. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed[3].

Fast R-CNN was the next version of R-CNN that shared many similarities with the original one. However, it improved the speed of object detection in two main ways. First, it performed feature extraction over the entire image before proposing regions, resulting in only one CNN running on the whole image instead of multiple CNNs over overlapping areas.

Second, it replaced the SVM with a softmax layer, which allowed for extending the neural network for predictions instead of creating a new model. Consequently, Fast R-CNN only required training one neural net, unlike R-CNN, which required training one neural net and multiple SVMs[2]. Fast R-CNN trains the very deep VGG16 network nine times faster than R-CNN, is 213 times faster at test-time, and achieves a higher mAP on PASCAL VOC 2012.[6].

At runtime, the detection network processes images in 0.3s (excluding object proposal time) while achieving top accuracy on PASCAL VOC 2012 with a mAP of 66 percent (vs. 62 percent for R-CNN[10])[6]. The original Faster R-CNN paper[7], reports a mean Average Precision (mAP) score of 73.2 percent on the PASCAL VOC 2012 dataset using the VGG-16 network architecture, which is 10 mAP more accurate than YOLO, but also 6 times slower[3]. In figure(1), you can see the comparison between the performance and speed of fast detectors.

In the final VOC 2012 leaderboard, Fast R-CNN + YOLO scored an mAP of 70.7 percent, coming in forth. YOLO, stayed at mAP of 57.9 percent, while being the only real time detector[3].

Finally, the paper on Single Shot Multibox detector[9] introduces a novel deep network object detector that achieves high accuracy without requiring pixel or feature resampling for bounding box hypotheses. This innovation translates to a remarkable boost in detection speed for high-accuracy applications, as demonstrated by its impressive performance of 59 FPS with an mAP of 74.3 percent on VOC2007 test, which surpasses the mAPs of other widely used object detectors such as Faster R-CNN and YOLO, while maintaining a much faster frame rate than Faster R-CNN and a comparable one to YOLO (7 FPS and 45 FPS respectively).

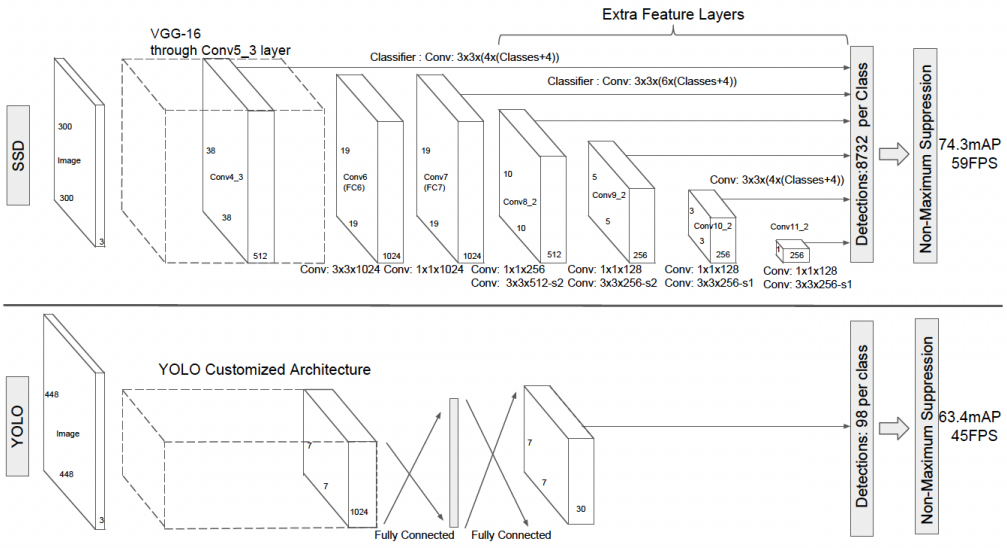

Figure 2: Comparison between YOLO and SSD structure[9].

4. Conclusion

Fast R-CNN and YOLO were groundbreaking object detection models when they were introduced, and they are still used in many applications today. Fast R-CNN is more accurate than its predecessor, R-CNN, and YOLO is known for its real-time performance. However, these models have some limitations. For example, Fast R-CNN can be slow and resource-intensive, and YOLO can have difficulty detecting small objects. SSD is another object detection model that achieved good results, but its accuracy lags behind some of the newer state-of-the-art models.

The newer state-of-the-art models have addressed some of the limitations of the older models and generally achieve higher accuracy and are more computationally efficient.

EfficientDet[4] is known for achieving state-of-the-art performance on object detection benchmarks while being computationally efficient. It has set new records on the COCO object detection dataset, achieving a mAP score of 52.2 on the test-dev set with only 66 million parameters, which is significantly fewer parameters than previous state-of-the-art models. Additionally, EfficientDet has also shown promising results on other object detection datasets such as PASCAL VOC and Open Images.

4.1 Improvements

Single-stage object detectors, like YOLO and SSD, are faster than two-stage detectors as they predict objects on an input with a single pass. YOLOv1 learned generalizable object representations, and YOLOv2(2016) improved upon YOLOv1 by adding batch normalization and anchor boxes. YOLOv3(2018) added a 53 layered backbone-based network and independent logistic classifiers to predict overlapping bounding boxes and smaller objects. SSD models share features between classification and localization tasks on the whole image, while YOLO models generate feature maps by creating grids within an image. RetinaNet, another single-stage object detector, addresses the class imbalance problem of detecting small objects by using a focal loss function during training and a separate network for classification and bounding box regression. Although YOLO and SSD provide good inference speed, they have a lower accuracy compared to RetinaNet.[11]

4.2 State of the art techniques

An overview of the top-performing real-time models based on their mAP scores at the COCO dataset is presented in figure(3). The COCO dataset consists of 80 classes for multi-class object detection and serves as the benchmark dataset for comparing object detectors[12].

In 2020, YOLOv4 introduced 'bag of freebies' for improved data augmentation and regularization during training, and 'bag of specials' for better mAP and faster inference. YOLOv5 proposed further data augmentation and loss calculation improvements and used auto-learning bounding box anchors to adapt to a given dataset. YOLOR, introduced in 2021, used a unified network for multitask learning, including object detection and multi-label image classification. YOLOX, also introduced in 2021, uses an anchor-free, decoupled head technique for separate classification and regression networks, with reduced parameters and

increased inference speed.[11] Figure(3) shows a summary of different models' performances in terms of mAP and inference speed.

Name	Year	Type	Dataset	mAP	Inference rate (fps)
R-CNN	2014	2D	Pascal VOC	66%	0.02
Fast R-CNN	2015		Pascal VOC	68.80%	0.5
Faster R-CNN	2016		COCO	78.90%	7
YOLOv1	2016		Pascal VOC	63.40%	45
YOLOv2	2016		Pascal VOC	78.60%	67
SSD	2016		Pascal VOC	74.30%	59
RetinaNet	2018		COCO	61.10%	90
YOLOv3	2018		COCO	44.30%	95.2
YOLOv4	2020		COCO	65.70%	62
YOLOv5	2021		COCO	56.40%	140
YOLOR	2021		COCO	74.30%	30
YOLOX	2021		COCO	51.20%	57.8
Complex-YOLO	2018	3D	KITTI	64.00%	50.4
Complexer-YOLO	2019		KITTI	49.44%	100
Wen et al.	2021		KITTI	73.76%	17.8
RAANet	2021		NuScenes	62.00%	

Figure 3: 2D and 3D object detector models and their performance[11].

While improvements to real-time object detectors often come from non-real time detectors, it's still important to analyze both types. While traditional detectors use CNN or RCNN backbones, newer technologies like Transformer Neural Networks are changing the future of object detection. Swin Transformers have shown promise as a backbone for object detectors, achieving state-of-the-art results with reduced model and pre-training data size. However, their slow inference speed means they're not currently suitable for real-time applications. Despite this, it's worth mentioning them as a possible solution in the future, such as for a YOLO-type detector with a transformer backbonen (see figure (4)) [12].

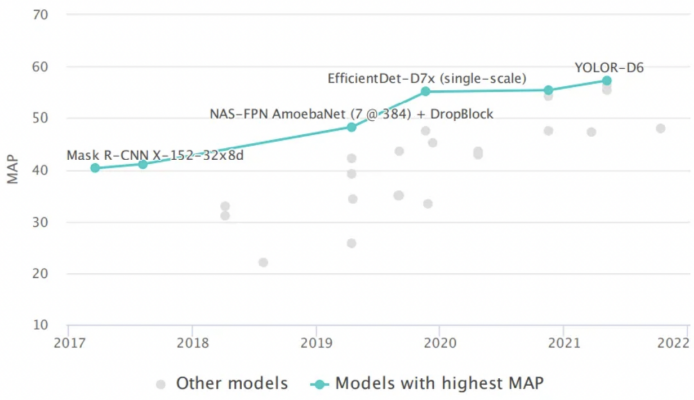

Figure 4: mAP of various models performing Real-Time Object Detection on COCO (Adapted from paperswithcode).

The paper on EfficientDet[4] proposes optimizations to improve efficiency in object detection using neural network architecture design choices. The proposed optimizations include a weighted bi-directional feature pyramid network (BiFPN) for multiscale feature fusion and a compound scaling method that uniformly scales resolution, depth, and width for all backbone, feature network, and box/class prediction networks. These optimizations have led to the development of a new family of object detectors called EfficientDet, which outperforms previous models in terms of efficiency across various resource constraints. The EfficientDet-D7, with a single model and single-scale, achieves state-of-the-art 55.1 AP on COCO test-dev with 77M parameters and 410B FLOPs, being 4x-9x smaller and using 13x-42x fewer FLOPs than the previous detectors [4].

Although there was an exponential growth of the number of papers revolving around object detection and massive improvements, there is no single technique that can be considered good enough to declare the problem of object detection completely solved. While techniques such as the YOLO series and EfficientDet series have shown significant improvements in terms of accuracy and efficiency, there is always room for improvement and new challenges that need to be addressed (see figure(5)).

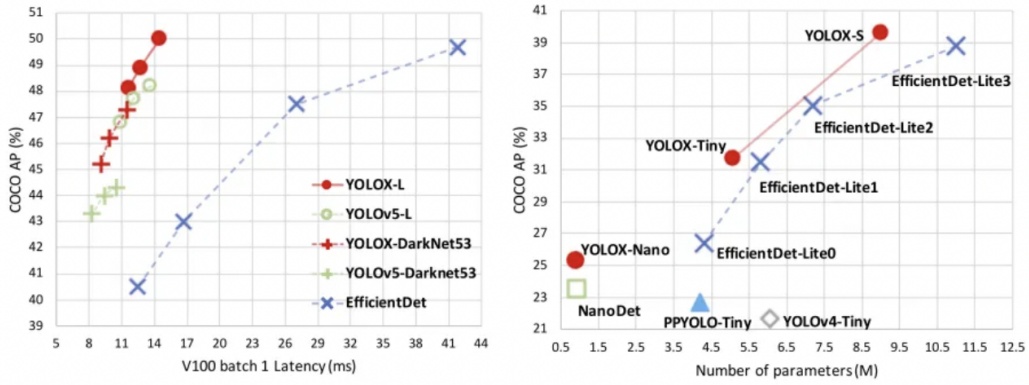

Figure 5: mAP of various models performing Real-Time Object Detection on COCO (Adapted from paperswithcode).

It is worth to mention that, the performance of these techniques may vary depending on the specific use case or application, and there may be limitations in terms of the types of objects or environments they can detect. Additionally, there are still challenges in terms of real-time processing, robustness to occlusion and clutter, and generalization to new and unseen objects.

Therefore, while these techniques represent significant advancements in object detection, there is ongoing research to further improve their performance and address the remaining challenges.

References

[1] Lewis (2016). Object Detection for Autonomous Vehicles Gene.

[2] Singh, S (2022).A complete guide on object detection its use in self-driving cars, Retrieved from https://www.labellerr.com/blog/how-object detection-works-in-self-driving-cars-using-deep-learning/.

[3]J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 779-788.

[4] Tan M, Pang R, Le QV. EfficientDet: Scalable and Efficient Object Detection. Published online 2019. doi:10.48550/arxiv.1911.09070

[5] Girshick R, Donahue J, Darrell T, Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2014:580-587. doi:10.1109/CVPR.2014.81

[6] Girshick R. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE; 2015:1440-1448. doi:10.1109/ICCV.2015.169

[7] Shaoqing Ren, Kaiming He, Girshick R, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE transactions on pattern analysis and machine intelligence. 2017;39(6):1137-1149. doi:10.1109/TPAMI.2016.2577031

[8] He K, Zhang X, Ren S, Sun J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE transactions on pattern analysis and machine intelligence. 2015;37(9):1904-1916. doi:10.1109/TPAMI.2015.2389824

[9] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector. In: Computer Vision – ECCV 2016. Springer International Publishing; 2016:21-37. doi:10.1007/978-3-319-46448-0-2

[10] Girshick R, Donahue J, Darrell T, Malik J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. IEEE transactions on pattern analysis and machine intelligence. 2016;38(1):142-158. doi:10.1109/TPAMI.2015.2437384

[11] Balasubramaniam A, Pasricha S. Object Detection in Autonomous Vehicles: Status and Open Challenges. Published online 2022. doi:10.48550/arxiv.2201.07706

[12] Azevedo, P (2022). Object Detection State of the Art 2022, Retrieved from https://medium.com/(at)pedroazevedo6/object-detection-state-of-the-art 2022-ad750e0f6003