Real-time Object Detection for Autonomous Vehicles using a combined Binary Classification and Deep Learning based approach

Puneet Kohli
Texas A&M University
College Station, TX USA
puneetkohli@tamu.edu

Anjali Chadha
Texas A&M University
College Station, TX USA
anjali_chadha@tamu.edu

Abstract

Improving real-time object detection systems is highly desirable, especially in the context of autonomous vehicles, where passenger safety is of utmost importance. There are various state-of-the-art Deep Learning based object detectors (YOLOv2, SSD, R-FCN), which often trade-off accuracy for speed and memory benefits to achieve real-time results. Research is being done to improve speed while still maintaining high accuracy.

We propose a combined real-time object detection system that first performs binary classification (whether there is something 'actionable' in the scene or not) followed by object detection using YOLOv2 with DarkNet. Our aim is to evaluate whether such a system provides speed and memory benefits over one solely relying on Deep Networks.

1. Introduction

Both industry and academia have been focusing attention on autonomous vehicles as the transportation means of the future. The Computer Vision task of Object Recognition is one of the most important tasks for autonomous vehicles to function well and to achieve human-like (or better) understanding of the environment. The advances in Artificial Intelligence, and specifically Deep Learning, have accelerated research in this domain. Yet, it is still an open ended challenge and improvements are still being made on top of existing literature. Most importantly, work is being done to improve speed of processing and detection, while still maintaining a high mean average precision (accuracy) [1, 2, 3, 4, 5, 6].

Our belief is that autonomous driving vision systems do not need to run computationally heavy object detection algorithm on every frame as it is not necessary that every frame will have something of relevance for decision-making purposes. Drawing a parallel to human driving, humans generally are less aware on empty highways except when another vehicle is passing by, or if a traffic sign comes by.

In this paper, we propose a combined system that runs two separate subsystems. The first subsystem is a simple binary classifier that marks the current scene as 'actionable' or not. In the context of this paper, an 'actionable' object is one which needs to be identified with finer detail, as it would be important for the autonomous vehicle's decision making system. Section 3.2 talks about the 'actionable' class in more depth. The second subsystem is a fully Deep Network based system which detects objects in the scene with a high accuracy. For this, we have identified YOLOv2 [3] as the object detection framework of choice. We further explain why we chose YOLOv2 in section 2.

We have evaluated a variety of data sets for training our system [7, 8, 9, 10, 11]. Some data sets are more general, whereas others are focused towards autonomous vehicles. There are also data sets specifically for road signs and traffic signals. We would be using a combination of some of the evaluated data sets to train our model for both 'actionable' classification as well as object detection.

Our hunch is that we would definitely get a performance boost in terms of speed, memory usage, and computation time. In terms of accuracy, we do not expect to deviate much from that which would be expected from running YOLOv2 independently. Our experimental analysis will validate whether or not this hunch is true. If we are successful, we would like to mention as future work, a more generalized approach to combined detection systems.

2. Preliminary Literature Survey

Over the last decade, we have come a long way in the field of object detection from using plain SVM for image classification [12, 5] to applying the state-of-the-art techniques like R-CNN [2], SSD [13], YOLO [4].

One of these approaches we considered for our application was R-CNN [2]. R-CNN uses region proposal methods to generate the potential regions of interest in the image and then run a classifier on these proposed boxes. This step is

followed by further processing to refine the bounding boxes in the image. However, as compared to YOLOv2 [3], this approach is harder to optimize as we are training each component separately.

YOLOv2 added various improvements to the original YOLO [4] framework to overcome YOLO's shortcomings like low recall compared to region proposed methods and significant number of localization errors.

We also examined other state-of-the-art methods like Faster R-CNN [14], R-FCN (Region based Fully Convolutional Network) [6] and SSD [13] (Single Shot Multibox Detector). But as per Redmon et al [3], YOLOv2 is superior to all of the approaches in terms of both speed (frames per second) and accuracy (mean average precision).

The choice of YOLOv2 as our final approach was also influenced by [1], where authors investigated the pros and cons of modern convolutional object detectors in terms of speed/memory/accuracy. As per [1], SSD is faster than YOLO for real time applications. And since YOLOv2 is an improvement over YOLO, we decided to use YOLOv2.

Dollar et al[10, 15] introduces Caltech Pedestrian Dataset for detecting pedestrians on road and discusses its application for the autonomous vehicles. Although we are not focussing on pedestrian detection, it is interesting to note that work has been done on generating very specific data sets to be used for autonomous vehicles.

Saini et al [16] proposed a combination of SVM and CNN technique for the purpose of Traffic Light Detection for autonomous cars. They are using a combined approach for detecting traffic light state similar to the one we propose for general object detection.

3. Proposed Technical Plan

3.1. Data Set

We plan to use the KITTI Vision Benchmark Suite [8] data set for training our model. KITTI data is collected from driving a car fixed with a sterio camera around the city of Karlsruhe, Germany. Although, the authors used COCO [9] and PASCAL VOC [7] datasets in the original papers, we are using KITTI since our task of object detection is specifically for self-driving autonomous vehicle. As KITTI does not contain labelled traffic sign data, we will additionally train the model for traffic signs using the Belgium Traffic Sign (BelgiumTS) [11] data set.

3.2. Classification of Actionable Objects

In the scope of this project, an 'actionable' object is one that will trigger our system to switch to the YOLOv2 object detector. We have identified the following high-level set of objects to be classified as actionable by our system

- Traffic lights
- Traffic signs
- Other vehicles on the road

Essentially, any autonomous vehicle should know the state of traffic lights, be aware of signboards and their semantic meaning, and also be alert of other vehicles on the road.

We will evaluate various binary classification approaches such as Logistic Regression, Naive Bayes, and Support Vector Machine, to determine whether a scene (input frame from a camera feed) contains any 'actionable' objects. At present we are planning to use the Histogram of Oriented Gradients (HOG) as our primary feature to distinguish between actionable and non-actionable scenes. The HOG approach has generally shown good results in existing literature [17, 18] with SVM and so we are expecting SVM to be our classifier of choice.

3.3. Real-time Object Detection

Once we have identified that the scene contains actionable objects as discussed in 3.2, our system will start using the YOLOv2 object detection framework in order to detect and classify objects in the scene, which would ideally further be used by the autonomous vehicle for decision making. As discussed in 3.1, we will be training YOLOv2 on the KITTI dataset. Our system will stop using the real-time detection framework when the scene no longer contains any actionable objects. This will be easy for the system to identify as we have a relatively manageable number of actionable object classes.

3.4. Evaluation

The goal of our system is to evaluate whether using a combined Binary Classification plus Deep Network based system has performance benefits over a fully Deep Network based system for real-time object detection. Our experiment will be to run both, our system, and only YOLOv2, on the same data set of video streams from an autonomous vehicle using the same hardware (CPU, GPU, etc). We will compare both systems based on the total computation time as well as total memory usage. We will also consider mean average precision, but our expectation is that the accuracy would be roughly the same as we will be using YOLOv2 as the object detector in both cases.

References

[1] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[2] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[3] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[4] 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)*, pages 779–788, June 2016.

[5] S. S. Nath, G. Mishra, J. Kar, S. Chakraborty, and N. Dey. A survey of image classification methods and techniques. In *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pages 554–557, July 2014.

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. *CoRR*, abs/1605.06409, 2016.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[8] A Geiger, P Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: the kitti dataset. 32:1231–1237, 09 2013.

[9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[10] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, April 2012.

[11] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool. Traffic sign recognition x2014; how far are we from the solution? In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Aug 2013.

[12] O. Chapelle, P. Haffner, and V. N. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, Sep 1999.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[14] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[15] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311, June 2009.

[16] S. Saini, S. Nikhil, K. R. Konda, H. S. Bharadwaj, and N. Ganeshan. An efficient vision-based traffic light detection and state recognition for autonomous vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 606–611, June 2017.

[17] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[18] Hilton Bristow and Simon Lucey. Why do linear svms trained on HOG features perform so well? *CoRR*, abs/1406.2419, 2014.