Machine Learning Engineer Nanodegree

Capstone Proposal

Nicholas Condo
May 22, 2017

Proposal

Domain Background

In recent years we have seen enormous progress in the areas of artificial intelligence, machine learning, and computer vision. One of the most prevalent examples resulting from these recent advancements is the autonomous vehicle. Currently, self-driving cars rely on a number of different sensors for information about its environment, including: cameras, lidar, radar, sonar, and GPS. Object detection is one of the many critical tasks necessary for autonomous driving. One such use case is detecting traffic lights and signs in order to properly follow the rules of the road, such as obeying the speed limit and stopping at stop signs. Additionally, a common method of tracking multiple objects is tracking-by-detection [1]. Since objects in the environment act differently according to their class (car, pedestrian, cyclist, etc.), it is helpful to first classify the detected object so it can be tracked more accurately.

Object recognition and detection have seen incredible improvements with the advancements in deep learning, beginning in 2012 when Krizhevsky et al. [2] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [3] with their deep convolutional neural network (CNN). Since then, researchers wielding CNNs have improved image classification to the point that they now outperform humans ILSVRC. While this is an incredible achievement, object detection is a much more challenging task. One of the first applications of CNNs to object detection appeared in 2014 with the now popular region-based convolutional neural network (R-CNN) architecture, introduced by Girshick et al. [4]. With the added demands of real-time performance necessary for autonomous driving, there has been a lot of effort put into developing accurate *and* fast object detection networks. Some of the most popular real-time/near real-time networks for object detection include: Faster R-CNN [5], YOLO [6], and SSD [7], with researchers working feverishly on developing faster, more accurate models every day.

Problem Statement

Object detection is a critical task for autonomous driving. In addition to demanding high accuracy, real-time inference speed is required. Therefore, the goal of the proposed project is as follows: given an image as input, create a pipeline that produces an image with bounding boxes around all cars in the image in real-time. A standardized definition of real-time speed does not exist; it varies depending on the task at hand. For this work I will define real-time to be at least 10Hz, which is also the speed requirements of the Self-Driving Car Challenge [8] hosted by Udacity and DiDi Chuxing.

Datasets and Inputs

For training and evaluation, I will use the object detection dataset provided by the KITTI Vision Benchmark Suite [9], which was specifically designed for autonomous driving. The dataset consists of 7481 training images and 7518 test images, comprising a total of 80,256 labeled objects. The dataset provides bounding boxes for object classes such as cars, vans, trucks, pedestrians, cyclists, and trams. However, I will focus only on detecting cars, as there is a dedicated benchmark on the KITTI [9] website for detecting cars only.

The data was collected by driving around a mid-size city, in rural areas, and on highways with a vehicle equipped with an advanced autonomous driving platform. The platform includes two high-resolution stereo cameras (grayscale and color), a Velodyne HDL-64E laser scanner, and a localization system, which includes GPS. Ground truth coordinates were obtained by manually labeling objects in 3D point clouds produced by the Velodyne system, and projecting them back into the image. While other datasets and benchmarks exist for object detection, such as the PASCAL VOC challenge [10], the KITTI [9] dataset offers real world data collected specifically for benchmarking autonomous driving tasks.

Solution Statement

To find a solution to this challenging problem, I will look to some of the latest object detection publications for inspiration. Most of the recent state-of-the-art object detection networks are based on the popular Faster R-CNN [5], which will be a good starting point for this project. Like the authors of Faster R-CNN [5], I will leverage transfer learning to build upon a well-proven classification model: VGG-16 [11]. I will use the convolutional layers of the VGG-16 model [11] pre-trained on ImageNet [3], then experiment with additional convolutional layers and different techniques for generating bounding boxes. Faster R-CNN [5] uses a regional proposal network (RPN) with different size anchor boxes to predict “objectness” scores at each position in the image, and non-maximum suppression (NMS) to output the best bounding box per detected object. I will implement a similar approach to form a baseline model, and experiment with other recently published techniques such as those described by Redmon et al. [6] and Liu et al. [7].

Benchmark Model

In order to gauge the effectiveness of my solution, I will use Faster R-CNN [5] as a benchmark model. KITTI's [9] public leader board provides an easy way to compare results with Faster R-CNN [5], which sits in 46th place in the Car object detection category (10th place among those with a published paper). The other models I have mentioned – YOLO [6] and SSD [7] – have not been tested on the KITTI [9] dataset (at least not publically), making Faster R-CNN [5] a more relevant benchmark model. Faster R-CNN [5] shows an AP of 79.11% for this task, and the authors report a runtime of 5Hz on a GPU. My goal will be to at least match this level of accuracy while achieving real-time speeds.

Evaluation Metrics

For measuring performance, I will follow the methods of the popular KITTI Vision Benchmark Suite [9], which provides online evaluation servers for researchers to test their models and keep an up-to-date view of state-of-the-art performance on various autonomous driving related tasks. Performance is measured using the well-established average precision (AP) metric introduced by Salton and McGill [12]. Detections are iteratively assigned to ground truth labels starting with the largest overlap, measured by bounding box intersection over union. True positives are required to overlap by at least 70%, and multiple detections of the same object are counted as false positives.

The formula for finding 11-point Average Precision:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} Pinterp(r)$$

where:

$$Pinterp(r) = \ max_{r\prime : r\prime \geqslant r} \, P(r')$$

and:

$$P(r') = \ the\ measured\ precision\ at\ recall\ r'$$

Precision is defined as:

$$Precision = \frac{true\ positive}{true\ positive + false\ positive}$$

Recall is defined as:

$$Recall = \frac{true\ positive}{true\ positive + false\ negative}$$

Project Design

For this project, I will implement the neural network architecture using Keras with a TensorFlow backend. The 7481 training images will be split so that 20% of the data will be set aside for validation, and the remaining 80% will be used for training. I will begin by reconstructing the Faster R-CNN [5] architecture, using a pre-trained VGG-16 [11] model for classification. I will also experiment with other popular pre-trained classification models, such as ResNet-50 [13]. I will fine-tune the pre-trained classification model by training on just the cars from the images, cropping out the cars by using the ground truth bounding boxes. After creating a base model with the Faster R-CNN [5] architecture, I will use some of the methods proposed by Liu et al. [7], which have shown to improve inference speeds to real-time levels. Some of the approaches of SSD [7] responsible for increasing detection speeds include: using a small convolutional filter to predict object categories and offsets in bounding box locations, using separate predictors (filters) for different aspect ratio detections, and applying these filters to multiple feature maps from the later stages of a network in order to perform detection at multiple scales. By experimenting with different combinations of classification models, testing multiple techniques for producing bounding boxes, adding custom layers to the network, and parameter tuning, I hope to outperform Faster R-CNN [5] in both accuracy and speed in detecting cars on the KITTI dataset [9].

References

[1] J. Janai, F. Guney, A. Behl, and A. Geiger. Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. In *IJCV*, 2015.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[5] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. SSD: Single Shot MultiBox Detector. *arXiv preprint arXiv:1512.02325v5*, 2016.

[8] Udacity and DiDi Chuxing. Self-Driving Car Challenge. https://www.udacity.com/didi-challenge, 2017.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *NIPS*, 2015.

[12] G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. *New York: McGraw-Hill*, 1986.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.