Predicting Object Bounding Boxes and Road Map Layouts in a Traffic Environment

Atul Gandhi Elliot Silva Ziv Schwartz
ag6776@nyu.edu egs345@nyu.edu zs1349@nyu.edu

Abstract

In today's ever-changing technological eras, one of the most exciting and hotly-debated advancements is that of self-driving cars. While the realization of such a futuristic piece of technology requires the coordination of many talents, from engineers to software developers to computer vision experts, the fundamental first step, as with any problem, is how best to represent the world around you. In this paper, we apply a pre-processing step of stitching together six 360° images into a single representation, then apply the YOLOv3 architecture for the task of object detection of nearby cars, trucks, and other vehicles. Our architecture transforms this into the two-dimensional road map representation, wherein additional AI decision-making would be applied to determine what best action the vehicle should take. We also learn a binary road map. Our results show potential, and we hope they can be expanded upon future work in the field.

1. Introduction

Creating an autonomous car system is a difficult problem, with significant repercussions for errors. An autonomous driving car must be able to fully understand the environment around it, from the road, lanes, traffic signals, and exits, to other vehicles on the road. It must know how the mechanics of braking, accelerating, and turning work, and how to space itself properly among the surrounding vehicles. Additionally, nowadays many cutting-edge self-driving car companies have explored the ethical component of self-driving cars. For example, if a man walks into the road in front of you while you're driving, do you hit the man or swerve into the car next to you?

All these decisions stem first from an ability to understand the world around you. In this project, we used a YOLOv3 architecture to produce bounding box representations of the world around it, using collections of images spanning 360 degrees around a car. The model identified bounding boxes for its prediction of where other vehicles are, based on a stitched representation of these images, and then translated these bounding boxes into the two-dimensional binary road map representation. For the road map prediction task, our model develops an average representation methodology based on labeled road images from the given dataset.

2. Literature Review

The object detection task is a well known and well-researched task in Computer Vision. Viola's (Viola & Jones) research was one of the earliest works in object detection. While it was technically described as an object detector, its primary use case was for facial recognition. The next major breakthrough came with Sermanet's (Sermanet et al., 2013) approach of using convolutional neural networks (CNNs) along with a sliding window. The proposed method classifies each part of an image as a either an object or non-object and then combines the result to get the final set of predictions.

More recent works involve approaching this problem in two steps. In the first step, called 'Region Proposal', regions with high probability of containing objects are determined. And the second step, called 'Object Detection', takes these regions as input to perform the final detection and classification of the object.

R-CNN (Girshick et al., 2013) was one of the major studies in this area. The proposed model achieved the then state-of-the-art results on the VOC-2012 and the 200-class ILSVRC-2013 object detection datasets. Despite the great performance, these models tended to be slow and future research led to faster models such as Fast R-CNN(Girshick, 2015) and Faster R-CNN (Ren et al., 2015). The most recent work in further enhancement of the R-CNN family of models is being done by Facebook's Detectron2 (Wu et al., 2019).

YOLO or "You Only Look Once" is another popular family of object detectors. While these approaches might be slightly less accurate than the R-CNN family of models, they are much faster and thus more suitable for real time object detection tasks. It was first proposed in the research by Joseph (Redmon et al., 2015). The model first splits the

input image into several grids of cells. Next, these cells independently predict a bounding box if the cell is supposed to be within a bigger bounding box (as shown in figure 1). Future research into the models have lead to development of YOLO9000 or YOLOv2(Redmon & Farhadi, 2016) and YOLOv3(Redmon & Farhadi, 2018) models.

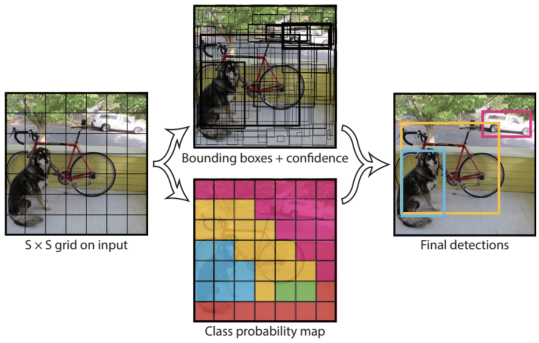

Figure 1. Original YOLO Methodology

3. Data and Intuition

This project is very relevant given that today several major technology companies are racing to be the first to deploy safe and reliable autonomous driving. However, one cannot simply jump to autonomous driving without developing several tasks along the way, such as object detection and road lane recognition. These, among other tasks, are essential components for autonomous driving to be successful and trustworthy to deploy in real world situations. The data used for this project consists of labeled and unlabeled scenes taken from a moving car. The data is organized into three levels: scene, sample, and image. A scene comprises 25 seconds of a car's journey. Within each scene, a sample is a snapshot of the scene occurring at every 0.2 seconds. Each sample contains six images, taken with cameras pointing 60 degrees apart in order to create a complete 360 degree sample of what is going around the car. Each scene contains 126 samples, each with 6 images (figure 2). The unlabeled dataset consists of 106 scenes while the labeled dataset contains 28 scenes, indicating a semi-supervised analysis. Additionally, there is extra information present for each sample, describing the current state of the car (stopped, lane change, turn, etc), that can be used to further enhance the model with particular specifications.

For intuition, our team debated at lengths how best to feed in the inputs. We considered different representations, including passing each image separately, a flat left-to-right (256x306*6) stitching, and flipping the back three images upside-down and placing them underneath the front three images. However, these representations were not the correct way to represent the image we were concerned with

Figure 2. Input images

outputting. We believed that if our inputs could as closely as possible resemble a two-dimensional top-down representation, that would work the best in predicting the bounding boxes. To accomplish this form of stitching, the images were assembled as seen below. First, sides were combined in the following order right (front back), forward behind, and left (front back). The front images are assembled on the left and the back images on the right. The front images are all rotated 90 degrees counterclockwise while the back images are all rotated 90 degrees clockwise. Our team believed this to be the most similar representation to a two-dimensional top-down view of the road since the lane is now depicted from left to right and the other objects are occupying a shifted presence in the overall layout (as shown in figure 3). There are several other permutations of the following stitching that may work better, yet our group built our model upon this representation.

Figure 3. Stitched Representation

4. Methodology

4.1. Bounding Box

One of the most fundamental requirements for a successful deployment of an autonomous car has to be its ability to identify and keep track of all the objects in its vicinity (both stationary and moving). This object detection task primarily involves identification of the object and estimation of its shape, size and position, or in other words, simply being able to draw a bounding box over an identified object in a two-dimensional or a three-dimensional space.

In this paper, we tackle the task of identifying objects such as cars, people etc and drawing a two-dimensional bounding box over a top-down bird's eye view of the road using images from six cameras placed all around the car. We approached this task as a two-dimensional object detection task combined with perspective transformation from a front view to a top down view. Our intuition was that given enough training samples, any common architecture used for object detection could also learn to translate the coordinates of the identified object into a different perspective.

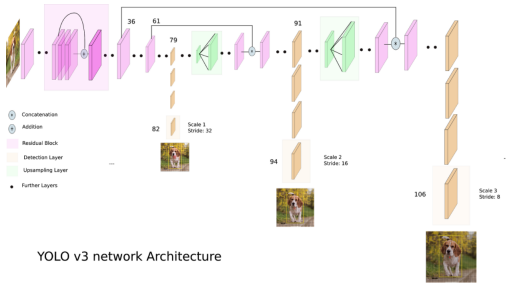

Figure 4. YOLOv3 Architecture (Kathuria, 2018)

We decided to use the YOLOv3 architecture for our task. The code was adopted from (Jocher et al., 2020). It uses a variant of Darknet-53 as its feature extractor and comprises a total of 106 convolutional layers. The object detection is performed at three different scales. For our purposes, these three scales were determined by down-sampling the dimensions of the input image by 32, 16 and 8. Finally, it takes the generated bounding boxes and applies non-maximum suppression to group boxes together. The complete architecture is shown in figure 4. As discussed earlier, the images were stitched together to get the objects as close to their position in a top-down view as possible. The annotations were also transformed to the format expected by the model i.e. for each box, we passed the coordinates of the center of the box (assuming the bottom left corner to be the origin) and the width and height of the box. We also passed the specific object category to our labels so that our model could learn to identify the type of object in addition to being able to better determine the size of the corresponding bounding box. Currently our model can only draw boxes parallel to the axes and this can be explored as a next step.

4.2. Road Map

The binary road map layout is an essential task that is needed to deploy autonomous driving in a real world situation. Without being able to analyze and anticipate what the layout of the road is, an autonomous vehicle or machine will not be able to complete the most basic functions and considerably threaten the safety of many. To complete this specific task, two possible inputs can be used to predict the layout of a random road, the stitched image road map used as the input for the bounding box task and the road images in the labeled dataset. Given the imbalance of data between labeled and unlabeled scenes (28 to 106), there will be an added benefit of using the stitched representations of the input images. Despite this, the overall reliability of these stitched images in terms of two-dimensional top-down roads is not very representative of the output we are aimed at predicting. For this reason, our model considers only the labeled road images. This allowed us to experiment with different techniques and we ended up returning the average road layout of the entire labeled dataset (as shown in figure 5). This value was thresholded at 0.5 in order to create a binary road layout.

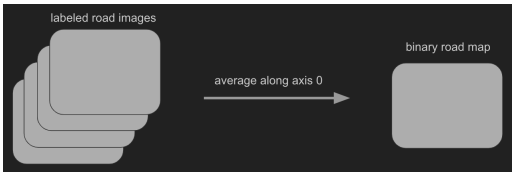

Figure 5. Road Map Prediction

5. Results

To evaluate how our group performs on the two tasks, we compute different evaluation metrics. For both tasks, threat score (ts) is the chosen evaluation metric. In the binary road map task, the model is evaluated on average threat score (ts) as defined below:

$$TS = \frac{TP}{TP + FP + FN}$$

For the bounding box, the model is evaluated across a range of thresholds (0.5, 0.6, 0.7, 0.8, 0.9). The metric is calculated as the weighted average over the union of the five thresholds.

Evaluating on the test set, our model obtained threat scores of 0.001 on the bounding box task and 0.68 on the road map task. The model does not perform well for the bounding box task with a threat score of 0.001. While it is not expected for

our model to perform this way, the task is very challenging and significant tuning is required to be able to perform well. First, the bounding boxes are represented from a top-down perspective whereas the input training images are taken from a perpendicular perspective, with images facing out from a central location on the car. Our model adapts the input images as stitched images that are rotated in order to try and mimic the top-down layout. To improve our results, further steps can be implemented to use the intrinsic parameters of the camera to project a forward-facing image to a top-down view. Coupling Computer Vision principles with deep neural network architectures has the potential to advance our model to obtain better predictions for object detection bounding boxes.

With the binary road map, it is very interesting to note that the model performs surprisingly well (over 50%) given that it is simply the average representation of the labeled dataset. Building on the assumption that the data being collected depicts real world data of road maps and layouts, the average representation is a great prediction to threshold against. Tying in statistics, the Central Limit Theorem clarifies that given a large enough sample that is representative of the domain of interest, any random unseen sample can be roughly estimated by the average of previous inputs. While this is a good model to start testing, it does not take into account the specific inputs of the given situation. This can lead to problems when the model is deployed in a new environment that perhaps has drastically different road maps and layouts, causing the model to predict average representations that are not applicable to the new environment. Future work on this task will be to try increasing the complexity of the predictions, including Gaussian blur, canny-edge detection, and non-max suppression to obtain the unique lanes and road map indicators.

References

Girshick, R. Fast r-cnn, 2015.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013.

Jocher, G., Kwon, Y., guigarfr, Veitch-Michaelis, J., perry0418, Ttayu, Marc, Bianconi, G., Baltacı, F., Suess, D., , , idow09, WannaSeaU, Xinyu, W., Shead, T. M., Havlik, T., Skalski, P., NirZarrabi, LukeAI, LinCoce, Hu, J., IlyaOvodov, GoogleWiki, Reveriano, F., Falak, and Kendall, D. ultralytics/yolov3: 43.1map@0.5:0.95 on coco2014, May 2020. URL https://doi.org/10.5281/zenodo.3785397.

Kathuria, A. What's new in yolo v3?, Apr 2018. URL https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Redmon, J. and Farhadi, A. Yolo9000: Better, faster, stronger, 2016.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement, 2018.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection, 2015.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks, 2013.

Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. doi: 10.1109/cvpr.2001.990517.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. Detectron2. https://github.com/facebookresearch/detectron2, 2019.