End-to-end Multi-Camera View Parsing to Top Down Object Detection On Road Maps

Annika Brundyn Francesca Guiso Noah Kasmanoff

Abstract

In this work we construct a top down view of a road-map and objects surrounding an ego car using six images taken from a camera mounted on top of a car. We use self-supervised learning to build a representation using a large unlabeled dataset and fine-tune on two downstream tasks, roadmap prediction and bounding box detection. Our approach achieves a roadmap threat score of 0.76, and identifies objects with a bounding box intersection over union (IoU) threat score of 0.007.

1. Introduction

Self-supervised learning (SSL) attempts to extract representations by solving pretext tasks to learn structure about a dataset without explicit labels. In this work, we leverage SSL to build representations of images taken from the top of a car to attempt to reconstruct a map of the surrounding road and to draw bounding boxes around nearby objects.

Computer vision has been used extensively to solve tasks like pose estimation, object detection and image segmentation. Object detection is commonly used in autonomous driving to build a thorough context for vehicle navigation. Translating perspective in photographs has also been used for 3D to 2D translation, image reconstruction and computer graphics.

In this work, we use SSL to learn representations that convey information about roads, objects and the environment, and fine tune in two downstream tasks, road map prediction and bounding box detection. Our approach achieves competitive results in the roadmap prediction task, but fails to generalize well to the bounding box prediction task. We hypothesize that the lack of results in this task is because the models have not been thoroughly trained given the limited time and resources we had access to.[1]

2. Approach

Our approach follows three steps: we (1) pre-train using self-supervised learning, (2) fine tune for the roadmap prediction task, and (3) fine tune for the bounding box task.

Self-supervised pre-training
Let $\mathcal{D} = \{(I_1, r, B), (I_2, r, B), ..., (I_n, r, B)\}$ define a dataset where I is a collection of images $I = \{i_1, i_2, ..., i_6\}$ taken from the top of a car, r is a binary roadmap where $r \in R^{800\times800}$ and B a set of bounding boxes $B = \{b_1, b_2, ..., b_n\}$ where each b_n has coordinates for bounding boxes and the class the object belongs to.

For SSL, we generate a target by drawing a random image target $y = i_t \sim I_n$ where $i \in R^{3\times256\times306}$. We let $x = I - \{i_t\}$ where $x \in R^{3\times256\times1836}$ with the block corresponding to i_t fully removed and set to 0. An example of this input can be seen in Figure 1.

Let f_θ define a *encoder*, g_ψ define a *decoder* and $z = f_\theta(x)$ the latent representation. We train this autoencoder using MSE.

$$MSE(x, y) = \min_{\theta,\psi} ||y - g_\psi(f_\theta(x))||_2^2 \tag{1}$$

For inference we drop g and use only f to extract the representation for a given x_i

$$z_i = f_\theta(x) \tag{2}$$

Roadmap prediction For this downstream task, let $x = I_k$ define an original collection of images without a missing image such that $x \in R^{3\times256\times1836}$, and $y = r_k$ the corresponding roadmap. We use a single MLP $h_\phi(\cdot)$ which maps $z \in R^{64} \rightarrow R^{800\times800}$, and we use this output as the target which which we optimize via binary cross entropy.

$$\min_\phi \sum_{i=1}^{N} y_i \log(h_\phi(z_i)) + (1 - y_i) \cdot \log(1 - h_\phi(z_i)) \tag{3}$$

[1] The repository with the code used is available at: https://github.com/annikabrundyn/driving-dirty

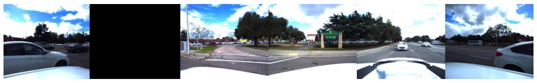

Figure 1. Input to the autoencoder

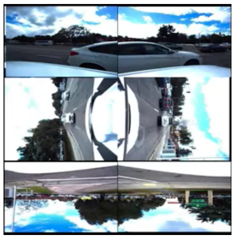

Figure 2. Spatially arranged image as one input to the bounding box model

Bounding box prediction Let $\dot{I}$ define a reshaped collection of images I where the images have been rearranged as shown in Figure 2. This is done to map the spatial locations of vehicles in image space, into the rough spatial location on the roadmap. Let C_ρ define a convolutional neural network (Yann LeCun, 1998) which maps $(\dot{I}, r) \in R^{4\times800\times800} \rightarrow R^{3\times800\times800}$ into a 3-channel image which merges the roadmap and the reshaped collection of images. This is done so that we can use a Faster-RCNN (Ren et al., 2015), F to perform bounding box detection in the image, with the constraints that the coordinates are the bounding box coordinates corresponding to the roadmap. This approach simplifies the 2D image to birds-eye view translation, into simply a linear projection of the detected bounding boxes on the image onto the roadmap. F uses the encoder f_θ as the backbone model.

We optimize the multi-part loss used in Faster-RCNN, (1) the MSE of bounding box coordinates:

$$\mathcal{L}_m = F_\pi(C_\rho(\dot{I}, r), B) \tag{4}$$

(2) the cross-entropy between the bounding box labels and the predicted log-probabilities, predicted by F_π.

$$\mathcal{L}_c = F_\pi(C_\rho(\dot{I}, r), B) \tag{5}$$

The system is trained end-to-end using both losses.

$$\min_{\pi,\rho} \mathcal{L}_c + \mathcal{L}_m \tag{6}$$

2.1. Other approaches

Initially, we attempted formulating the bounding box as a binary map prediction similar to the roadmap problem. The goal being to predict binary pixels corresponding to the object boxes on the map instead of coordinate prediction. Although our model did start to detect objects, these object appeared as long streaks of coloring on the binary map that weren't easily separable into clearly defined objects. An example of one of these predictions is included in our Appendix (section 6.1). We were curious to explore approaches that did not require significant pre-processing or post-processing to solve the problem but instead could potentially predict end-to-end.

Our next attempt was to perform regression on the bounding box coordinates directly. Our naive approach to this problem failed since the predicted bounding boxes, using MSE error on the coordinate values, were collapsing in the centre of the map. However, this motivated us to explore how we could adapt existing approaches to supervised object detection, such as Faster-RCNN, for our problem.

3. Experimental Setup

3.1. Data

Our data consist of images captured from 6 cameras attached to the ego car. [2] The dataset is organized into three levels: scene, sample, and image. A scene is 25 seconds of a car's journey. A sample is a snapshot of a scene at a given time frame. Each camera image is of shape 256 x 306 x 3. Each scene is divided into 126 samples, such that consecutive samples are about 0.2 seconds apart. Each camera captures a 70 degree view. In total, 134 scenes are provided of which only 28 have labelled information for road-maps and bounding boxes.

3.2. Compute resources

We trained on a single NVIDIA K80 GPU on the NYU Prince cluster and trained for no more than 24 hours using PyTorch Lightning (Falcon, 2019).

4. Analysis

4.1. Autoencoder

We chose a pretext task that we hoped would be hard to learn in order to extract more meaningful representations. Using all six images from a sample as input allows us to extract representations for a full scene surrounding the car. By randomly removing one of the images, it prevents the model from overfitting any one particular camera view. Figure 3 shows a sample prediction from the autoencoder. There are several distinct boundaries indicating that the autoencoder has learned to distinguish car, road and sky. In this particular example it has also hallucinated a car on it's view of the

[2] For more details on the ego car and dataset provided, please refer to the project description.

road. Although there is no way for the autoencoder to infer from the other views whether there was another car on the road in front of it, this implies that it has implicitly learned where other objects may occur.

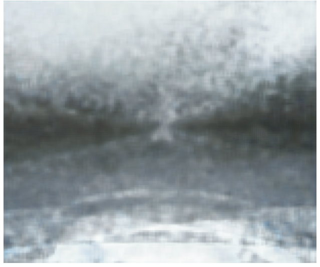

Figure 3. Reconstructed image missing from autoencoder input.

4.2. Road-map

We evaluate the roadmap predictions using a threat score metric over the predicted pixel values, defined as:

$$TS = \frac{TP}{TP + FP + FN} \tag{7}$$

Our roadmap model was very good at predicting the two horizontal roads, which appear in the large majority of samples we observed. However, it was not very confident at predicting roads that were rotated at different angles. We show an example of a prediction on a held out sample in Figure 5 that demonstrates this. Although we did not have time to explore this, one could augment the data by randomly rotating roadmaps to potentially improve the model performance.

Our final model achieved a threat score of $TS = 0.76$ on the test set.

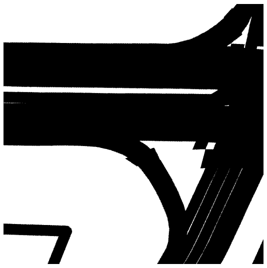

Figure 4. Target roadmap

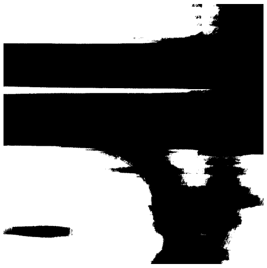

Figure 5. Predicted roadmap

4.3. Bounding Boxes

The bounding box model is evaluated using a threat score metric averaged over a number of different intersection over union (IoU) thresholds (0.5, 0.6, 0.7, 0.8, and 0.9). The final score is given by:

$$TS(t) = \sum_{t} \frac{1}{t} \cdot \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \tag{8}$$

Our final model achieved an average threat score of $TS(t) = 0.007$ on the test set. This model was only trained for about 5 hours and was far from converging when it was tested, but showed promising early predictions. Notice in Figure 7 that the model has learned to place car objects only on roads. These predictions could clearly be improved by including an angle prediction in our model since by default faster-RCNN only outputs rectangular coordinates.

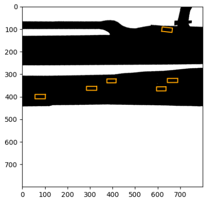

Figure 6. Target bounding boxes

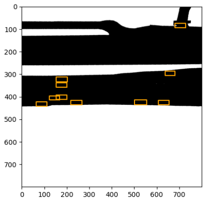

Figure 7. Predicted bounding boxes

5. Related Work

This work presents a generalized approach to self-supervised learning, view transformation and object detection; one clear application area is in autonomous driving. The most significant strides have been made with the use of light detection and ranging (lidar) systems. In this paper, we discuss an alternative method for building an environment model that could be used as input to autonomous driving vehicles.

Results presented in (Henaff, 2019) is a natural extension to this work. Both of these papers create a world model based on data collected by camera, as opposed to lidar. This camera however, was placed above a highway, only providing temporary information of a single vehicle. Our approach suggests that one could possibly predict a top down road-map, using only cameras attached to the agent vehicle. Additionally, we have avoided pre-processing the inputs to our system which facilitates real-time prediction.

6. Conclusion

In this work we trained an autoencoder on an image infill pretext task to build representations from the large unlabeled dataset for the road map prediction task, and the bounding box task. For the roadmap prediction task we achieved a threat score of 0.76. For the bounding box prediction task we achieved an IoU threat score of 0.007. We anticipate more improvement to come with more training time and hyper-parameter tuning to improve both models.

6.1. Future direction

In future works we hope to explore different approaches to the self-supervised pretext task. Our final bounding box model was severely under-trained, but the initial results were very promising and we would like to evaluate it's performance given more time and hyper-parameter tuning. In addition, we intend to would like to explore other extensions to this model such as predicting an angle on our rectangular bounding boxes and customizing the default faster-RCNN architecture. Finally, we have not explored any pre-processing or post-processing, we think that these tasks could greatly improve the performance of our models.

Acknowledgements

We thank Yann Lecun, Alfredo Canziani, Mark Goldstein, Jiachen Zhu, and the rest of the DS-GA-1008 teaching staff. We also thank William Falcon for insightful discussion.

References

Falcon, W. Pytorch lightning. *GitHub. Note: https://github.com/williamFalcon/pytorch-lightning Cited by*, 3, 2019.

Henaff, M., C. A. . L. Y. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. *arXiv preprint arXiv:1901.02705.*, 2019.

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

Yann LeCun, Léon Bottou, Y. B. P. H. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 1998.

Appendix

6.2. Binary map approach to bounding box prediction

Our initial attempt at bounding box detection was to convert this to a segmentation problem. An example prediction is given in Figures 8, 9

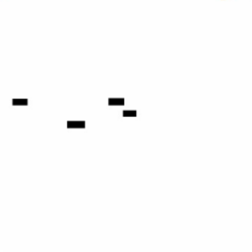

Figure 8. Target bounding boxes

Figure 9. Predicted bounding boxes

6.3. Hyperparameter tuning results

All models were trained using `Adam` at various learning rates.

Table 1. Tested autoencoder configurations.

HIDDEN DIM	LATENT DIM	LR	VAL LOSS (MSE)
128	64	0.0001	0.07265
64	32	0.0001	0.07336
128	32	0.001	0.08357

Table 2. Examples of tested roadmap model configurations. (Note: in all configurations, the weights of the encoder module were extracted from the top two performing autoencoder networks.) A learning rate of 0.0001 was used for all models.

LOSS	UNFREEZE EPOCH	PRETRAINED MSE	TS
MSE	20	0.07336	.8408
BCE	0	0.07265	.8442
BCE	20	0.07265	.8529