Semi-supervised approach to DC-GAN for road map layout and object detection

Neil Menghani [*1] **Wenjun Qu** [*1] **Xulai Jiang** [*1]

Abstract

In this paper, we propose a novel semi-supervised approach to detect road layout and environment using multiple images. More specifically, given six colored images that capture the surroundings of a vehicle, we use fully convolutional layers to convert the input images into a bird's eye view to perform road detection and object detection. We developed our model based on MonoLayout, a current state-of-the-art approach in the field. We further enhance our model's performance using a self-taught learning framework. Our model achieves reasonable performance with a bounding box threat score of 0.0020 and road map threat score of 0.7951 on the validation set.

1. Introduction

An active area of research in machine learning today is the application of deep neural networks to the domain of autonomous vehicles. Researchers in both academia and industry are working to build models that interpret a car's surroundings in real time from 3D images. In this paper, we propose a novel semi-supervised approach that builds on current state-of-the-art models.

We can narrow down the question of how to build a model to interpret a car's environment by focusing on the problem of learning a mapping from a 360-degree perspective to a bird's eye view. Building this mapping of the car's surroundings is a complex task with many elements that can be broken down into smaller tasks. We focus our efforts on two sub-tasks: (1) drawing the layout of the road around the car and (2) detecting the objects and their locations within this road map.

[*]Equal contribution [1]Courant Institute of Mathematical Sciences, New York University, New York, United States. Correspondence to: Neil Menghani <nlm326@nyu.edu>, Wenjun Qu <wq256@nyu.edu>, Xulai Jiang <xj652@nyu.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

The dataset we use to learn a mapping from image space to top-down space includes 134 scenes, each consisting of 126 samples from a 25-second journey of a car. Each of these samples has 6 images captured by a camera facing a different orientation. We use the images to generate a road map layout in the form of an (800×800) binary tensor where each pixels is 0.1 meters in physical space and the car is located in the center of the map facing right. We also generate an $(n \times 2 \times 4)$ tensor representing bounding box coordinates for n objects in the surroundings of the car in the top-down space.

Our primary contributions include incorporating semi-supervised learning into state-of-the-art supervised approaches to this domain that utilize Deep Convolutional Generative Adversarial Networks (DC-GAN). We also use a unique loss functions with unlabeled data to achieve semi-supervised learning.

2. Related Work

2.1. Object Detection

Recent progress in 3D object detection generally uses LiDAR (Chen et al., 2017) or Stereo vision (Chang & Chen, 2018) to estimate the location of objects. Despite the rapid progress on monocular depth estimation, image-based 3D object detection still lags behind LiDAR-based methods. To improve the performance, researchers have proposed another method that first maps a monocular image to a bird's eye view representation. They can then use 2D object detection methods to discover 3D objects. One of these examples that has significantly inspired our work is MonoLayout (Mani et al., 2020). Unlike other works, MonoLayout neither requires pre-processing nor post-processing and it directly converts the input image to bird's eye view and estimates scene layouts.

2.2. Semantic Segmentation

Similar to modern image classification, semantic segmentation has achieved significant improvements using Fully Convolutional Networks (FCN). The mainstream models use an encoder-decoder structure. The encoder, generally implemented using other backbone networks such as ResNet,

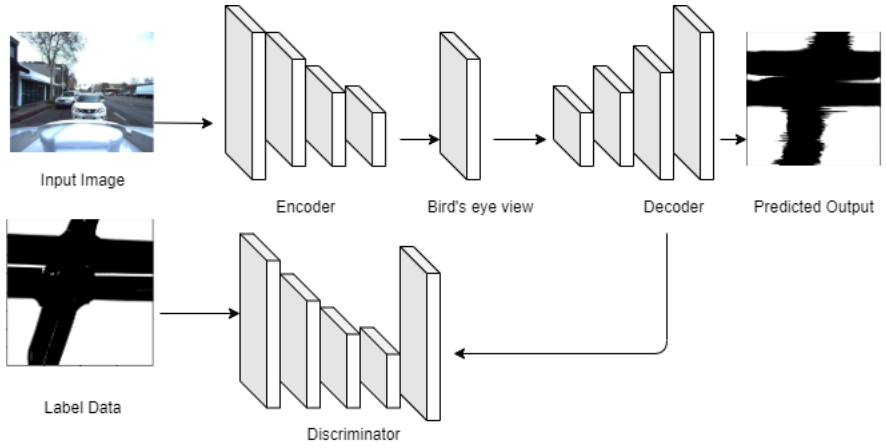

Figure 1. Overview of the model. The model takes in a set of images concatenated together to predict the layout in bird's eye view. Both road map and bounding box task uses the same architecture with different pre-processing and post-processing.

is used to reduces the input dimension and captures high level semantic representations. The decoder, takes in the result semantic representation and recovers the spatial information. Using a similar idea, Chen et al. (2018) proposed DeepLabv3+ that achieved test set performance of 89.0% on the PASCAL VOC 2012 dataset.

2.3. Semi-Supervised Learning

The aforementioned solutions rely heavily on a large amount of label data. However, it is usually expensive and difficult to label images with pixel-level annotations. To reduce the need of such heavy human effort, researchers have proposed numerous semi-supervised approaches using Generative Adversarial Networks (Goodfellow et al., 2014). For example, Han et al. (2018) uses the generator to detect objects on both labeled and unlabeled images. These images and their segmentation results are fed into the discriminator to predict if the input data are from labeled or unlabeled images. On the other hand, Hung et al. (2018) treat the output of the discriminator as the supervisory signals. The prediction maps generated from the discriminator are used as ground truth to achieve a self-taught learning scheme.

3. Semi-Supervised Road Map and Object Detection

3.1. Road Map and Object Detection Segmentation

Our model, illustrated in Figure 1 and adapted from the aforementioned MonoLayout, uses a DC-GAN to build a mapping from the image space to the top-down space for each task separately. The generator of the GAN consists of an encoder-decoder structure trained to build an output in the top-down space. The encoder uses ResNet 18 to extract features from the 6 input images, and the decoder builds a confidence map to generate a binary tensor as the output. The discriminator of the GAN is an FCN that distinguishes between "real" and "fake" (800×800) tensors generated by the generator in each task.

Unlike typical GANs in which the generator uses random noise to build "fake" images, our implementation instead uses the outputs generated from the unlabeled data as "fake". In this way, our model modifies the MonoLayout approach to work in the semi-supervised domain.

3.2. Bounding Box Detection

The object detection task requires an additional step to obtain the desired format for the bounding boxes: an $(n \times 2 \times 4)$ tensor of object coordinates representing n objects with four 2-dimensional coordinates. The output of our generator, an (800×800) coordinate map, needs to be analyzed to determine individual objects. We use the findContours function from the opencv2 package to detect a set of points identifying each contour in the confidence map generated by our trained network. This method accurately differentiates between non-overlapping shapes. We then use minAreaRect and boxPoints from opencv2 to generate the minimal rectangle enclosing the 2D point set for each object, thus providing bounding boxes. The resulting output is an $(n \times 2 \times 4)$ tensor representing the predicted bounding box coordinates for n objects.

3.3. Loss Function

Given an input image $\mathbf{X}_n$ of size $(3 \times H \times W)$, we denote the encoder by $ENC()$ and decoder by $DEC()$. Let the generator $G() = DEC(ENC())$, the output of $G(\mathbf{X}_n)$ is the predicted probability map of size $(1 \times 800 \times 800)$. The discriminatory $D()$ takes a probability map of size $(1 \times 800 \times 800)$ and outputs a confidence map of size $(1 \times$

50×50). Letting $\mathbf{Y}_n = D(\mathbf{X}_n)$ and $\hat{\mathbf{Y}}_n = D(G(\mathbf{X}_n))$, the discriminator network can be trained by minimizing the spatial cross-entropy loss $\mathcal{L}_D$:

$$\mathcal{L}_D = \sum_n^N CE(\hat{\mathbf{Y}}_n, 0) + CE(\mathbf{Y}_n, 1) \tag{1}$$

Inspired by Hung et al. (2018), we define our generator loss as:

$$\mathcal{L}_G = \mathcal{L}_{ce} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{semi}\mathcal{L}_{semi} \tag{2}$$

where λ_{adv} and λ_{semi} are the hyper-parameters that control the adversarial loss and semi-supervised loss respectively. When training with label data, we set the $\mathcal{L}_{semi}$ to 0 and only minimize the cross entropy loss and the adversarial loss as proposed below:

$$\mathcal{L}_G = \sum_n^N CE(G(\mathbf{X}_n), \mathbf{Y}_n) + \lambda_{adv}CE(\hat{\mathbf{Y}}_n, 1) \tag{3}$$

We aim to train the generator to fool the discriminator by maximizing the probability of the predicted results being generated from the true distribution.

When training with unlabeled data, we set the $\mathcal{L}_{ce}$ to 0 since there is no ground truth label. We then use the trained discriminator to generate the fake label for the input data. To do so, we use the confidence map $D(G(\mathbf{X}_n))$ to infer the regions sufficiently close to the ground truth distribution. We binarize the confidence map with a threshold value to highlight the trustworthy region. The resulting generator loss becomes:

$$\begin{aligned}\mathcal{L}_G = \sum_n^N & \lambda_{adv}CE(\hat{\mathbf{Y}}_n, 1) \\ & + \lambda_{semi}CE(G(\mathbf{X}_n), \mathbf{I}(\hat{\mathbf{Y}}_n > T_s))\end{aligned} \tag{4}$$

where $\mathbf{I}$ is the indicator function and T_s is the threshold value. In addition, when perform semi-supervised training, we have to reduce λ_{adv} to prevent the adversarial loss to over-correct the prediction to fit the ground truth distribution.

4. Experiments

We train the proposed model on a single RTX 2070 GPU with 8 GB memory. For both the generator and discriminator, we use Adam as our optimizer. The learning rate is set to 2.5×10^{-4} and 1×10^{-4} respectively. We set the hyper-parameters λ_{adv} to 0.1 when training with labeled data and 0.001 when unlabeled data is used. We also set λ_{semi} to 0.1. The threat score T_s when generating fake labels is set to 0.7 and 0.5 respectively for road map detection and bounding box detection. We adopted the generator and discriminator model from the original MonoLayout code and our training process is heavily inspired by Hung et al. (2018).

We train our model over the scenes between 106 and 130 in the labeled data using supervised learning while leaving scenes between 131 and 134 for model validation purposes. Later, we implement a semi-supervised structure and train on the unlabeled data in the first 105 scenes to improve model performance.

4.1. Data Preparation

For data transformation, we first resize all six input images into six $(3 \times 256 \times 256)$ image tensors. We then use torchvision utilities to concatenate the six tensors, forming the final input tensor of size $(1 \times 512 \times 768)$. To train the bounding box generator, we use the proposed method from the previous section to convert object coordinates of $(n \times 2 \times 4)$ to a binary tensor of $(1 \times 800 \times 800)$. The interior of the bounding boxes are filled with value 1. The resultant binary tensor is used as the ground truth label for bounding box task.

4.2. Supervised vs. Semi-Supervised Learning

For both supervised and semi-supervised learning approaches, we choose ResNet 18 as our encoder to extract features from our concatenated image input. Then, the feature tensor is passed into two separate decoders with the same structure for the bounding box and road map generation tasks respectively. Both decoders generate (1) tensors as confidence maps for the bounding boxes/ road maps. Confidence maps as well as "ground truth" binary tensors are then forward through two separate discriminators, which share the same structure for each task, to compute the loss of discriminators. When adapting the semi-supervised learning method, since there is no "ground truth" for the unlabeled data, a different loss function is implemented as mentioned before.

Figure 4.2 shows the visual result of the semi-supervised model applied to Scene 131 Sample 7. In Figure 4.2, the 2-by-3 grid on the top shows the layout of concatenation applied to the input images during preparation. The third and fourth rows compare the model prediction results (left) with the ground truth (right) of road map task and bounding box task respectively.

As we can tell from the result, the road map prediction is very good at detecting horizontal roads while much weaker to recognize the vertical branches. The reason to such "preference" of the model may be caused by the limited information of the vertical branching road. Since the vertical branching roads are only captured by the side camera while the horizontal main road is usually captured by all camera, more information of the main road are extracted and thus

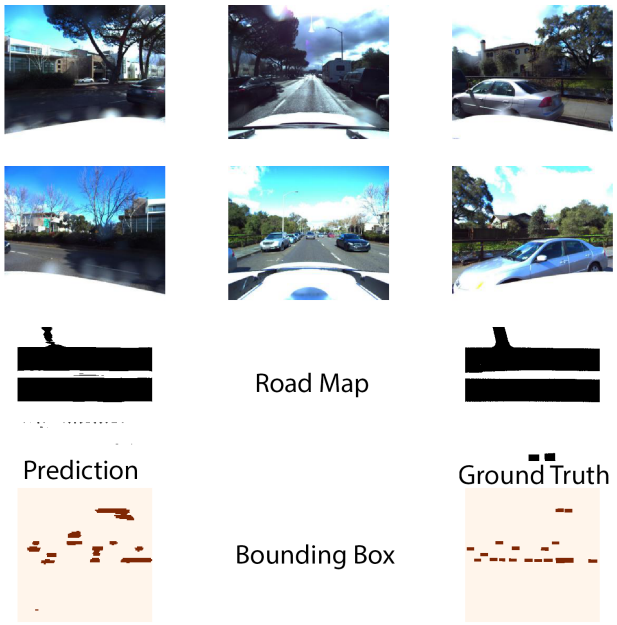

Figure 2. Model Result

better predictions can be made than the vertical branches.

The road map prediction model also struggles when the ego car is making a turn, where the features of images change dramatically between each sample. The bounding box prediction model suffers the same problem as the road map one. Although it can predict horizontal rectangular regions that contains many objects, it can barely separate each object vertically within a large region.

Table 1 shows the performance comparison between the supervised model and semi-supervised model. The performance criterion we used is call "threat score" (TS), which is similar to the intersection over union (IoU) criterion. For the bounding box task, semi-supervised learning is able to boost the performance of supervised model by approximately 185%, while only about 0.4% for the road map prediction.

	Bounding Box TS	Road Map TS
Supervised	0.0007	0.7919
Semi-supervised	0.0020	0.7951

Table 1. Performance Comparison

5. Conclusions

In this project, we are able produce quite good results for the road map prediction given surrounding images using a DC-GAN structure with semi-supervised learning. Such a good result is no surprise given how well GAN models can approximate the probabilistic distribution within the images. However, the result of the bounding box prediction model is much less satisfactory. One of the reasons causing such difference is the sparsity in the box image space: there are much fewer pixels belonging to the object bounding boxes than the pixels that belong to other things. Because our model outputs a binary tensor, the decoders are intuitively doing a classification among the pixels to two classes: "class 1 of boxes" and "class 2 of others". The imbalanced examples between the two classes force the model to perform an "imbalanced classification", which adversely affects our result.

References

Chang, J.-R. and Chen, Y.-S. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5410–5418, 2018.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Chen, X., Ma, H., Wan, J., Li, B., and Xia, T. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Han, X., Lu, J., Zhao, C., You, S., and Li, H. Semisupervised and weakly supervised road detection based on generative adversarial networks. *IEEE Signal Processing Letters*, 25 (4):551–555, 2018.

Hung, W.-C., Tsai, Y.-H., Liou, Y.-T., Lin, Y.-Y., and Yang, M.-H. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

Mani, K., Daga, S., Garg, S., Narasimhan, S. S., Krishna, M., and Jatavallabhula, K. M. Monolayout: Amodal scene layout from a single image. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 1689–1697, 2020.