Road Map Layout Generation and Bounding Box Prediction

Jatin Khilnani [*1] **Henry Steinitz** [*1] **Chady Raach** [*1]

Abstract

In this paper, we present our work for the final project of Deep Learning Spring 2020 class. The objective is to train a model using six images, each captured by a different camera attached to a car, in order to map a bird's eye view (BEV) of the surrounding area of a vehicle and detect nearby objects. This could incidentally lead to classification of detected objects and prediction of action being performed by them. We contented with the first two objectives as the scope of this project. Although we believe that these two objectives could be attained using a combined model, we preferred to have two separate architectures for organizational reasons within our team. Hence, we explore different approaches for each of the problem, and finally utilize a common feature extractor backbone with problem oriented architecture to attain the described results. We perform a survey of the self-supervision methods to learn useful representations from the provided unlabeled set, but limit the training to supervised domain.

1. Introduction

In recent years, computer vision problems such as image segmentation (semantic and instance) and object detection have been greatly advanced through deep learning, mostly by usage of Convolutional Neural Networks (CNN). These networks take advantage of the stationarity, locality and compositionality of the images to extract features in an efficient way. Convolution feature maps used by CNN can be shared by object detector and region proposal networks to detect objects in the input images which is the main constraining task of this project. The overall objective of the project is to train a model using images captured by six different cameras attached to the same car, each covering a specific view, to generate a top down view of the surrounding area. It involved two tasks: 1) Semantic segmentation: Map the road into 800×800 matrix that illustrates whether each pixel corresponds to the road in the top-down view. 2) Object Detection: Predict bounding box coordinates for each of the identified objects (cars, trucks, pedestrians etc. - labels not required) from the camera images and report their coordinates in the BEV map.

Convolutional networks seem to be a natural solution to this problem because they are based on the properties of the input (features/representations), and take advantage of enhanced computing capabilities to perform the required convolutions using many optimized modules that have been implemented in PyTorch, based on previous work related to object detection and region proposal such as RCNN (Girshick et al., 2014), Faster-RCNN (Ren et al., 2016) and YOLO (Redmon et al., 2016). Similar architecture can be used to implement a classifier that associates labels to each detected object as a task extension.

The main problem preventing the natural utilization of these models is the raw data we are given for the project. In principle, these supervised models require box coordinates for each detected object in the training set which allows to learn its features and delimiting its extent. However, the coordinates we are given are the 2D representation on the BEV map of each detected object, which makes the task of learning features from proposed regions - the way suggested by the related work and implemented in PyTorch - very difficult. To fix this, we explore combination of models with modified architecture to either make it learn conversion of given coordinates to frontal view, or the image to a top-down view for a direct mapping to target labels. Along with our simplified final architecture, which achieved decent results using a straightforward feature extraction approach through convolutions, we present another implemented approach that did not give us as good results within the project timeline, but we feel can be worked upon to improve the overall prediction performance.

2. Related Work

Semantic segmentation has a rich history in the literature and shown great results on Cityscapes Dataset (Cordts et al., 2016) which is very similar to our dataset. However the labels in semantic urban scenes understanding are available

[*]Equal contribution [1]Deep Learning Final Project Spring 2020, New York University. Correspondence to: Jatin Khilnani <jk6373@nyu.edu>, Henry Steinnitz <hjs410@nyu.edu>, Chady Raach <cr3144@nyu.edu>.

for each pixel in the input high quality images. This differs from the labeled set in the project as the images given were compressed and the pixel-wise label only corresponded to road image. More recent work on Panoptic segmentation (Kirillov et al., 2019) seemed more convenient to our problem in terms of identifying road and objects through a single architecture. Approach there is to solve a combination of semantic segmentation and instance segmentation task together (unified image segmentation), wherein each pixel i is assigned a pair of labels (l_i, z_i) where l_i represents the semantic class of pixel i and z_i represents its instance ID. Based on the same, our first understanding of the problem urged us to combine the task of road mapping and object detection. However, it requires generating such labels for all pixels/objects in the input image.

We then move forward with separate architectures for the two tasks and explore state-of-the-art models relevant to our object detection problem in detail. Pseudo-LiDAR from Visual Depth Estimation (Wang et al., 2020) requires a huge processing work including depth estimation and mapping 3D coordinates for formatting the raw data to train such model. We look into the implementation of Learning to Map Vehicles into Bird's Eye View (Palazzi et al., 2017) which is directly applicable in the context of our project yet necessitates the need for coordinates on the frontal image plane as additional input. On the other hand, Kim & Kum (2019) provide a way to map the input image into corresponding BEV by learning camera extrinsic.

We also survey self-supervision literature like Jigsaw (Noroozi & Favaro, 2016).

3. Methodology

3.1. Description of the data

Two data sets were given to achieve our objectives: labeled dataset containing 28 scenes and unlabeled dataset containing 106 scenes. Each scene is described by 126 samples, representing a 25-second car journey. Each sample (Figure 1) contains 6 images captured from different orientation to give an overall view ($360°$) of the surrounding areas of the car. The annotations given with the labeled dataset are the following: 800×800 matrix to describe to the road map, bounding box coordinates to position the objects on the BEV map and two category vectors to describe the class and the action of each detected object.

There are two main challenges dealing with the labeled data. First, how can the images be fed to a network in a way that respects geometrical constraints. We account for the fact that they represent a continuous 3D space where some regions can be captured at the same time by two cameras. Second, the data is weakly labeled in the sense that bounding boxes are given on the BEV coordinates and cannot be used to point the detected objects in the captured images and extract the corresponding features the same way it is done in standard object detection architectures. We research the possibility of learning the mapping between the frontal image features to BEV coordinates using camera extrinsic, and finally rely on the model to learn this aspect as well.

Figure 1. Input images yielded by 6 cameras targeting different orientations

3.2. Alternate Approach

We experiment with an alternate approach for the bounding box prediction task as depicted in Figure 2. This architecture is divided into three separate modules as following.

- Feature Extractor: We use an encoder-decoder architecture to extract hidden representations which are then stitched together. This is done by adopting U-Net (Ronneberger et al., 2015) model for multiple images, with skip connections.
- Bird's Eye View Generation: We combine the concatenated representations with the transformed target coordinates & object labels to convert the input target image into BEV, with box coordinates mapped accordingly as required by subsequent module. We also normalize the pixel values and crop the image into 164×212 for the prediction.
- Bounding Box Prediction: We train a Faster-RCNN (Ren et al., 2016) model from scratch to utilize the transformed box coordinates for making the coordinate and label prediction. Since this architecture produces results with coordinates for only two vertices on the bounding box, we assume the edges of the box to be parallel to the axes and compute threat score accordingly.

This architecture took considerable time to code and train, and lead to issues with `NaN` loss values. Hence, we proceed with a simplified model based on our road image architecture to achieve the results.

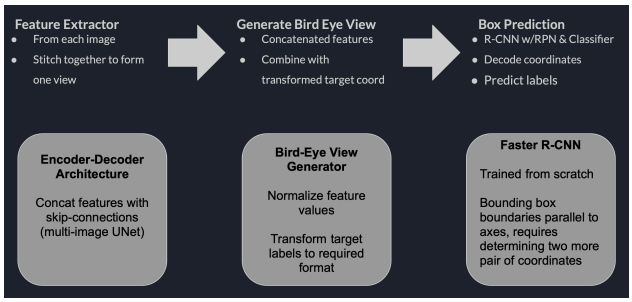

Figure 2. Alternate bounding box detection architecture

3.3. Implemented Architecture

First, our approach to deal with the 6 images in a sample that represent an overall view (360°) is to stack them in a fourth dimension (in addition to the channels, height and width dimensions). A shared feature extractor (Resnet18 (He et al., 2015)) is applied to the images, the outcomes are then concatenated to form a compressed hidden state of the input images and fed to different layers according to the task we are doing (road mapping/ box prediction). This approach is pretty natural as the 6 cameras are capturing cityscape views and likely the same objects appears in different scales and positions. Besides, it reduces the size of the model (number of parameters) and avoids unnecessary computation.

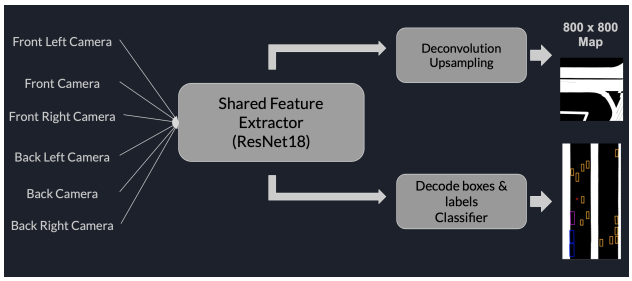

Figure 3. Overview of the implemented model

For the Road Map Layout task, then we apply layers of deconvolution and upsampling to decode the hidden state fed in from the feature extractor and form the 800×800 matrix coding the map. `ReLU` non-linearties are used in the decoding layers except for the last one where sigmoid is used in order to fit the format of the target matrix (each component is equal to 1 when it is a road and 0 otherwise).

The architecture that maps image features to bounding box coordinates consists of two fully connected layers so that it incorporates minimal prior knowledge. The output of the fully connected layer is a single box position along with its class label probabilities. The box is then matched to the single closest box in the true labels. The distance between boxes is naively measured as the sum of square distances between their corresponding endpoints. We then compute the sum of a cross-entropy loss ℓ_c between the labels and an MSE loss ℓ_m between the bounding box coordinates. The losses are relatively scaled with a hyperparameter $\lambda > 0$ so that the total loss computed is

$$\ell = \ell_m + \lambda \ell_c$$

Where, $\ell_m^{(i)} = MSE(d_i, \hat{d}_i), \ell_c^{(i)} = CE(l_i, \hat{l}_i)$.

Code and related artifacts are available on our `road-map-bounding-box-prediction` repository.

4. Training and Results

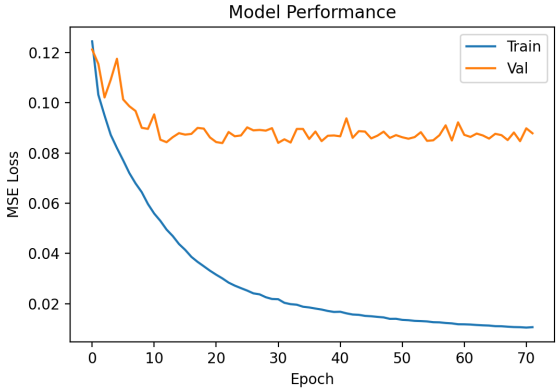

Figure 4. Epoch-wise training and validation loss for road image generation task

Both models were trained using ADAM optimizer with a learning rate of 0.001 and $\beta = (0.9, 0.999)$. We split the supplied labeled scenes into a training set containing 24 scenes and validation set containing 4 scenes. The final models were then trained using the training set and saved after each epoch. The saved models with the highest threat score on validation set were then selected as the final model (Table 1).

Figure 5. Road image generation ground truth vs prediction on a validation sample after 62 epochs

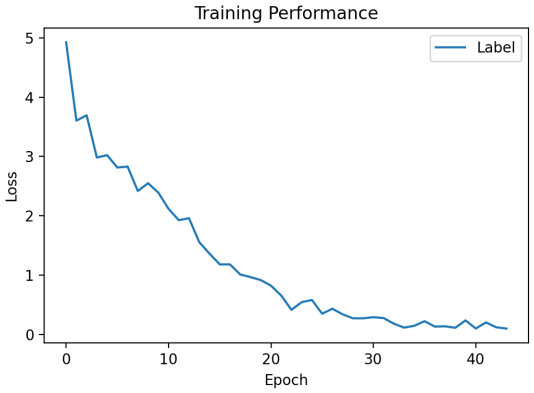

Figure 6. Epoch-wise training loss for labels in bounding box prediction task

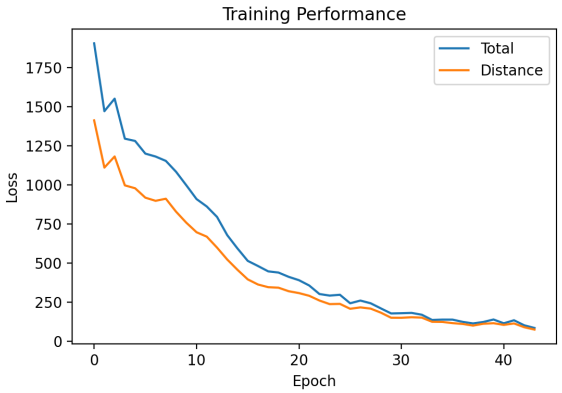

Figure 7. Epoch-wise training loss (Distance, & Total) for bounding box prediction task

Both models likely can be improved with straightforward modifications. The generated road images contain unnatural edge artifacts (Figure 5) that can be removed with different upsampling techniques. The box prediction model would perform better with support for multiple predictions.

5. Conclusion

The project provides a healthy insight into the autonomous driving domain wherein we are able to explore and learn various state-of-the-art architectures employed for solving a computer vision problem. More importantly, we are able to understand and visualize the effect of each layer in a neural network by experimenting with our custom architecture that has generated results comparable to the existing models in the current context.

Future Work. We would like to complete and test our alternate approach for bounding box detection to gauge its performance against the simplified model (after enhancing the same as well), to verify any benefit in learning representations with the bigger and more complex model. We would also like to exploit the temporal aspect of the data to utilize in a recurrent style architecture, along with self-supervised pre-training preferably in pretext invariant mode.

	Road Map	**Bounding Box**
Training	0.91	0.03
Validation	0.82	0.02

Table 1. Threat Score Results

References

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. *The Cityscapes Dataset for Semantic Urban Scene Understanding*. PhD thesis, TU Darmstadt and TU Dresden and MPI Informatics, 2016.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. 2015.

Kim, Y. and Kum, D. Deep learning based vehicle position and orientation estimation via inverse perspective mapping image. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 317–323, 2019.

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollar, P. *Panoptic Segmentation*. PhD thesis, HCI/IWR, Heidelberg University, Germany, 2019.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. *ECCV*, 2016.

Palazzi, A., Borghi, G., Abati, D., Calderara, S., and Cucchiara, R. Learning to map vehicles into bird's eye view, 2017.

Redmon, J., Divvala, S., and Farhadi, R. G. A. *You Only Look Once: Unified, Real-Time Object Detection*. PhD thesis, University of Washington, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. 2016.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015.

Wang, Y., Chao, W.-L., Garg, D., Hariharan, B., Campbell, M., and Weinberger, K. Q. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. 2020.