SELF-SUPERVISED PRETRAINING TO IMPROVE SELF-DRIVING CARS

Brian Kelly [*] **Yada Pruksachatkun** [*] **Xiaoyi Zhang** [*] **Weicheng Zhu** [*]

Abstract

This work explores several key problems in vehicle navigation: object detection and road map layout segmentation. We use Faster R-CNN and U-Net for this problem, and show that pretraining on ResNet leads to gains in performance. We find that our lane detection model performs well in general, although there are improvements to be made in the inference of corners and turns that are not captured in full by the camera. On the other hand, our bounding box model generates too many anchor boxes, which is the cause of the low performance.

1. Introduction

Deep neural networks have been deployed in safety-critical computer vision tasks such as self-driving cars. In order to function properly, an autonomous vehicle must accurately and reliably detect both surrounding objects and the lanes in the road. While deep neural networks normally require a large amount of data to train the model properly, the amount of labeled image data is often limited and may be insufficient in many training scenarios.

To ameliorate this problem and augment the labeled data, researchers began to employ self-supervised learning. In contrast to manual labeling, in self-supervised learning, a deep neural network is trained with an objective function that acquires supervision from the data itself. By pretraining on many unlabeled images, self-supervised models provide image features that can be further fine-tuned on various down-stream tasks and achieve superior performance. Several models for self-supervised pretraining, such as AlexNet (Krizhevsky et al., 2012), VGG (Simonyan & Zisserman, 2015), DenseNet (Huang et al., 2016) and ResNet (He et al., 2016) have achieved widespread adoption, and have excellent performance in various image recognition tasks.

The dataset consists of snapshot samples, each of which contains six images captured by cameras that are mounted on a car in 70 degree intervals. Given that approximately 80% percent of the images were unlabeled, we employ self-supervised learning to augment the labeled data. For a set of six input images, this paper sought to construct models with two inference goals: to detect and classify objects surrounding the car (object detection); and to draw a binary road map layout (lane detection). We first train ResNet and AlexNet on the unlabeled data to extract image features, and then feed the features into models for the supervised down-stream tasks. For the object detection task, we implement Faster R-CNN (Ren et al., 2015), which is a two-step method that first proposes regions of interest (RoI) and then predicts the object bounding box and corresponding categories. For the lane detection task, we apply U-net (Ronneberger et al., 2015), which extracts features with convolution blocks from image and constructs segmentation map with upsampling. We find that for both down-stream tasks, self-supservised pre-training on ResNet leads to better performance.

2. Related Work

2.1. Object Detection

There has been a great deal of research in object detection models. Recent efforts can be separated into one-step and two-step methods. Two-step methods, such as Faster-RCNN (Ren et al., 2015), generate proposal regions that have a high probability of being an object, and then compute class probabilities on the proposal regions. Meanwhile, one-step methods perform classification and regression directly on an input image.

Two-step methods include methods that first detect regions that may contain objects and then compute class probabilities for each region of interest. Early approaches, such as Region of Interest Convolutional Neural Network (R-CNN) and Fast R-CNN (Girshick et al., 2014; Girshick, 2015), use a selective search for region proposal generation, a fixed greedy approach that combines the most similar regions of initial set of region proposals. In order to address the significant time required to train and to deploy these models, the Faster R-CNN model Ren et al. (2015) overcomes this challenge by using an end-to-end CNN-based region proposal network (RPN).

One-step methods, such as YOLO (Redmon et al., 2016) and SSD (Liu et al., 2016), do not include the intermediate step of region proposals. With YOLO (Redmon et al., 2016), a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes, by first dividing images into grids, and then predicting bounding boxes on the grid and class probabilities for each grid element. It has been shown to improve computational speed while still retaining performance. Single Shot Detection (SSD) (Liu et al., 2016) uses the CNN pyramidal feature hierarchy such that each feature map only attends to objects of a certain scale.

2.2. Lane Detection

Often studied as an image segmentation task in computer vision, lane detection has given rise to several different model architectures. The encoder-decoder architecture of SegNet (Badrinarayanan et al., 2017; Mamidala et al., 2019) maps the low resolution encoder features to full input resolution features for pixel-wise classification. Usually combined with a self-attention mechanism, the distillation approach transfers scene structural knowledge from a teacher model to a student model (Hou et al., 2020; 2019). In contrast to the previous architectures, U-net (Ronneberger et al., 2015) was originally designed to address the limited training data in the context of biomedical segmentation, and assigns a category to each pixel in the image.

2.3. Transfer Learning and Self-Supervised Pretraining

Data augmentation in computer vision has been well-explored in the image classification domain, starting with MNIST (Zagoruyko & Komodakis, 2016). Most approaches have used self-supervision, with image mirroring, cropping, and rotation to augment images. Another approach is to exploit unlabeled data with self-supervised training with pretext tasks to learn better the image representations. Multiple pretext tasks are proposed to pretrain the convolutional networks that extracts the high-dimensional visual representations from images, including patch matching (Doersch et al., 2015), rotation classification (Gidaris et al., 2018), colorization (Zhang et al., 2016) and blank filling (Pathak et al., 2016). Recent efforts have explored pretraining objectives for a variety of computer vision tasks, including object detection (Li et al., 2019).

3. Methodology

3.1. Pretraining

We use AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2016) as the feature extracting model to obtain the feature representation of the images. AlexNet is comprised of stacked layers of convolutions followed by max-pooling, and finally a softmax function. ResNet introduces skip connections between convolutional layers, which help solve for the vanishing gradient problem.

We learn the image feature network with two pretext tasks: the first is classifying the image rotation (Gidaris et al., 2018); the second is classifying the camera angle for each picture. The self-supervision on rotation can help the feature extracting model to better learn the low-level visual features in images. Additionally, since we leverage six images from different camera angles, we propose another pretext task of camera angle classification to learn different localized feature of different views.

3.2. Bounding Box Detection

For the bounding box problem, we use Faster R-CNN, since it has been shown to be more effective than YOLO at detecting small objects. Faster R-CNN both decreases computation time while increasing effectiveness of region proposal networks by computing proposals with convolutional neural networks (CNNs) using the convolutional feature maps in region proposal networks for generation.

The Faster R-CNN model consists of several parts. First, a region proposal network (RPN) uses convolutional feature maps to generate potential anchor boxes. These boxes are passed to two parallel classification layers: a classifier that predicts whether a given anchor box is the background or foreground, and a regressor that adjusts the boundaries of the anchor boxes. The resulting anchor boxes are called *regions of interest (RoI)*. Before passing the output to the R-CNN, non-maximum suppression is used to remove overlapping RoIs. Downstream convolutional layers finally adjusts the RoI boundaries, and predicts the final object bounding boxes and corresponding categories.

Training is implemented using stochastic gradient descent. The total loss for the network is the sum of the losses in the RPN and Faster R-CNN. Moreover, the loss in each network is computed using cross-entropy for the classifier and smooth L1 loss for the regressor that are combined in a weighted sum. While the original paper trains the RPN and Fast R-CNN separately, our implementation follows recent trends (Zhang & Yang, 2017; Kendall et al., 2018) that trains both simultaneously.

3.3. Lane Detection

To infer the binary road map based on the images, we use U-net (Ronneberger et al., 2015) for segmentation, as it is a popular image segmentation model. In contrast with classical lane detection problems, the task at hand is to generate a binary road map instead of segmenting road from the original images. The U-net model, given an input image, extracts features from the image with convolution blocks and downsamples with pooling, and constructs a segmentation map with upsampling. In this work we replace the convolution block in the original paper with a ResNet backbone to improve the performance of our downstream

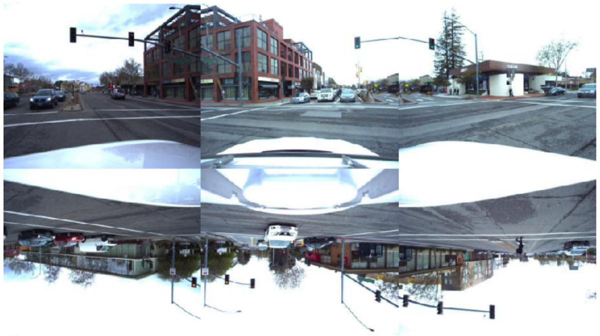

Figure 1. The self-centered view of image sample set

Table 1. Experiment Results (threat score)

Experiment	Object Detection (Dev)	Object Detection (Test)	Road Map Layout (Dev)	Road Map Layout (Test)
No pretraining	0.0168	-	0.762	-
AlexNet pretraining	-	-	0.730	-
ResNet pretraining	0.0173	0.019	0.784	0.73

model. To implement the model, we resize the image to 800×800 for input and obtained a binary map at 800×800. We train the model with Adam optimizer and cross-entropy loss.

4. Experiments

We use the street-view images captured by cameras at 6 angles to detect the surrounding objects and segment the lane within a 80×80 square centered at the location of the car. To synthesize images pointing at 6 different angles, we concatenate each set of images into a 3×2 grid (3 for front views and 3 for rare views in Figure 1). With this alignment, we create a self-centered view of a sample set which corrects the relative position in images with respect to the bird-view ground truth. We then interpolate and resize the aligned image into 800×800 to match our desired output. For each image, we also introduce data augmentation by random cropping to adapt the slight change of angles in different scenes. In the dataset, we have 106 unlabeled scenes and 28 labeled scenes. We use all the unlabeled scenes for pretext pre-training our feature extractor, and split 23 labeled scenes for training two supervised tasks and 5 scenes for validation.

To train the model, we use the Torchvision (Marcel & Rodriguez, 2010) package for our models. For self-supervised and supervised tasks, we use ADAM optimizer at learning rate 3×10^{-4} and weight decay at 0.0005. In the training based on pretrained feature extractor, we used stochastic gradient descent with 0.005 learning rate, 0.9 momentum and 0.0005. We also use linear warm-up in the first epoch and reduce learning rate by 10 times every 3 epochs. For both tasks, we train 20 epochs and pick the model with the highest TS on validation set. Since Adam accelerates the convergence, and the featuring model is not pretrained in non-pretraining experiments, the model overfits and during training obtains the best TS in 3 epochs. The model with pretrained feature extractor trained by SGD takes longer to reach the optimal (bounding box after 8 epochs and road map after 14 epochs).

5. Results

We evaluate both tasks on threat scores **(TS)**. For lane detection, we compute the TS between output binary map and ground truth:

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

For detecting bounding boxes, we compute the weighted TS under different thresholds $t \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ of intersection over union (IOU):

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

Our results are in Table 1. After experimenting with AlexNet and ResNet, we find that ResNet outperforms AlexNet as a feature extractor. This may be because the ResNet has better ability in feature extraction due to its skip connection structures. Furthermore, we found that pretraining helps improve performance, increasing performance from bounding box from 0.0168 to 0.0173 TS for bounding box and 0.764 to 0.784 TS for lane detection.

6. Discussion

From the results we observe that the self-supervised pretraining can improve the ability of both lane detector and traffic detector. The overall performance of the threat score on lane detection is promising, while the weight threat score of traffic detection task leaves much to be improved. We visualize two samples in Figure 2. These examples demonstrate the quality of our work in detail. For the road detection, the ResNet U-net model can map the majority lane to the correct location. Although the lane in six-angle view images is twisted from its true shape, the non-linearity in

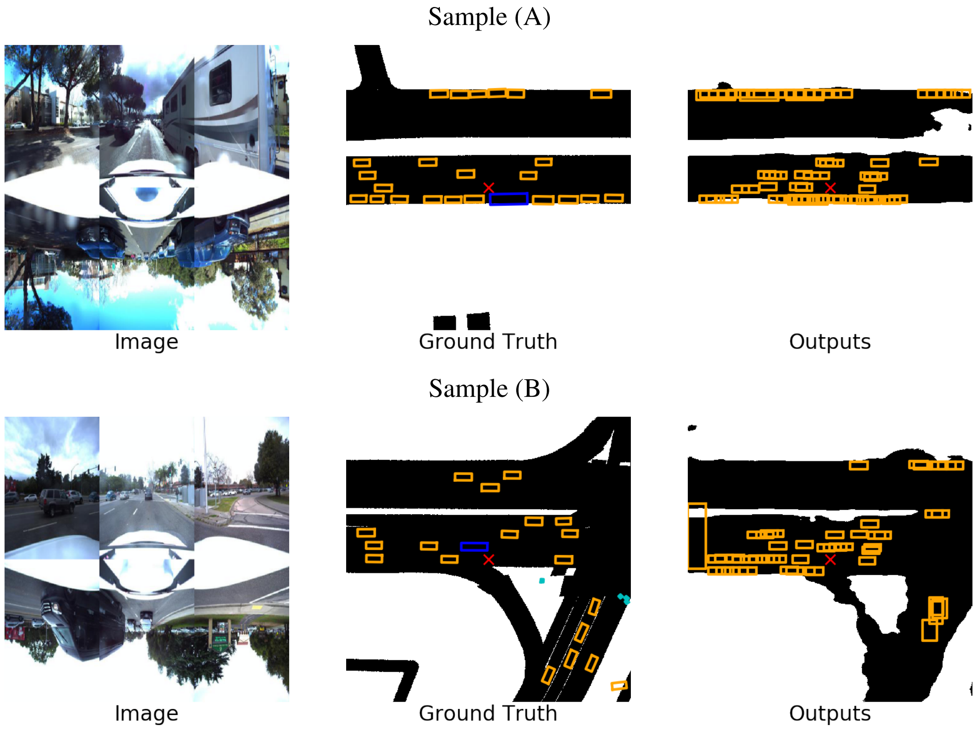

Figure 2. Examples of model inference

U-Net can implicitly perform the conformal mapping from the panoramic view to bird view. We also observe that the model struggles with some of the parts of the layout that is not captured fully by the camera such as corners and turns, which the model predicts with rough contours and white spots in the outputs. This indicates our model cannot make the a sound inference to the remote parts beyond the view of cameras, which is a limitation of the U-net.

For the task of detecting bounding boxes, we use Faster R-CNN to learn the location of the bounding boxes and its class probabilities. The nature of the region proposal networks (RPN) leads to overestimation on the anchor boxes. Comparing ground truth and our output in Figure 2, we can observe that multiple overlapping bounding boxes are inferred for the same vehicle. This shows our model is limited on distinguishing similar anchor boxes proposed from RPN, which is the main reason for sub-optimal model performance. Furthermore, our model only uses two corners of bounding boxes, shown in the sample (B) in Figure 2. This inherently limits the granularity of object detection. Although this doesn't significantly impact the resulting TS, it may cause issues in real-world utilization.

7. Conclusion

In summary, we find that ResNet outperforms AlexNet as a feature extractor for pretraining tasks. In line with previous owrk, we find that pretraining improves downstream task performance by leveraging features in the unlabeled data. Furthermore, we find that AlexNet outperforms ResNet as a pretrained feature extractor. While the downstream task for object detection using a Faster R-CNN model appears to have a low threat detection, the road layout U-Net model achieves a high performance.

Future Work While the current study provides clear evidence that pretraining is useful in bounding box and road lane detection, future studies should investigate alternative pretraining models and objectives for bounding box and/or road lane detection. Given the tepid performance of the objection detection model, other pretraining methodologies may transfer information better to this downstream task.

To increase performance on bounding box, we can first adjust the non-maximum suppression threshold. Secondly, work from other teams have demonstrated that better performance on bounding box detection with image-to-map model than Faster R-CNN styled anchor box learning. It is worth investigating why image-to-map approaches can out-perform learning anchor boxes.

For road layout segmentation, there may be better ways to perform the road image merging of the six angles that take into account the overlap of the cameras field of view on the car to better infer harder curves or turns in the road.

References

Badrinarayanan, V., Kendall, A., and Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015. URL http://arxiv.org/abs/1505.05192.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018. URL http://arxiv.org/abs/1803.07728.

Girshick, R. B. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hou, Y., Ma, Z., Liu, C., and Loy, C. C. Learning lightweight lane detection cnns by self attention distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1013–1021, 2019.

Hou, Y., Ma, Z., Liu, C., Hui, T.-W., and Loy, C. C. Inter-region affinity distillation for road marking segmentation. *arXiv preprint arXiv:2004.05304*, 2020.

Huang, G., Liu, Z., and Weinberger, K. Q. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Li, H., Singh, B., Najibi, M., Wu, Z., and Davis, L. S. An analysis of pre-training on object detection. *ArXiv*, abs/1904.05871, 2019.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector. In *ECCV*, 2016.

Mamidala, R. S., Uthkota, U., Shankar, M. B., Antony, A. J., and Narasimhadhan, A. Dynamic approach for lane detection using google street view and cnn. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pp. 2454–2459. IEEE, 2019.

Marcel, S. and Rodriguez, Y. Torchvision the machine-vision package of torch. In *ACM Multimedia*, 2010.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016. URL http://arxiv.org/abs/1604.07379.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

Ren, S., He, K., Girshick, R. B., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *ArXiv*, abs/1605.07146, 2016.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. *CoRR*, abs/1603.08511, 2016. URL http://arxiv.org/abs/1603.08511.

Zhang, Y. L. and Yang, Q. A survey on multi-task learning. *ArXiv*, abs/1707.08114, 2017.