Self Supervised Learning based Road Map Layout Generation and Object Detection

Sree Gowri Addepalli [1] **Amartya Prasad** [2] **Sree Lakshmi Addepalli** [3]

Abstract

This paper presents our solution for the traffic environment task where the objective is to train a model using images captured by six different cameras attached to the same car to generate a top down view of the surrounding area for road map layout and object detection. For this task we use a simple framework for contrastive learning, along with self supervised Monocular Depth estimation on the upstream task. Then we use these learned representations for our downstream tasks using state of the art model for Road Prediction like DeepLab and Object Detection using Faster RCNN to achieve our goal. Our code is publicly available at: Github

1. Introduction

2D bounding box prediction for 3D objects monocular 3D and road layout prediction(Semantic Segmentation) from purely monocular images is a challenging task. Most existing approaches depend on Lidar sensors attached to cars for getting depth data which is quite costly. Further there is a lack of quality labelled data. For the road map prediction task, we proposing using SIMCLR(Sim), a contrastive learning framework for visual representations, followed by DeepLabv3(Liang-Chieh Chen & Yuille, 2017), a SOTA semantic segmentation model. DeepLab uses atrous convolutions, or dilated convolutions, which helps extract more dense features even if objects are of varying scale which prevents loss of information with deeper networks. For the object detection task, depth information, which gets lost in 2D images, becomes all the more crucial. As a solution, we propose using Monodepth2(Godard et al., 2019), a self supervised depth estimation model, that returns a depth map, that is, a depth value for each pixel. Monodepth2 is designed to have minimum projection loss which robustly handles occlusions, a full-resolution multi-scale sampling method that reduces visual artifacts and an auto-masking loss to ignore training pixels that violate camera motion assumptions while helping us learn from temporal dynamics in the data.

The output from the trained Monodepth2 models is then passed as input to SimCLR. We expect this to help the model learn feature representations, depths in particular. Followed by this, we propose using the learned representations together with FasterRCNN (Shaoqing Ren & Sun, 2016), a SOTA Object Detection Model for predicting bounding boxes. Faster RCNN is built over the Resnet50 backbone,is a state of the art object detection that uses region proposal network and anchor generations, which acts like an attention mechanisms to tell where to look for object boundaries in the network. Finally, in order to get better performance, we also combine L1 loss along with a custom IoU based loss to directly optimize IoU.

1.1. Related Work

Recent state of the art models include traditional UNet to models like Full-Resolution Residual Networks(FRRN B) (Tobias Pohlen, 2016) which uses two residual streams,for high localisation accuracy and the pooling stream, which is responsible for high classification accuracy. As seen in the paper we see the model performs the best on road task followed by DeepLabv3. DeepLabv3 is light weight model, giving an accuracy as good as FRRN B, which propels our selection for road map prediction task.

For images that have 3D objects to predict bounding boxes we research the state of art methods that use various self supervised techniques to calculate depth per pixel in the image and help us project the top down view of these objects. We then research methods that help us understand depth and top view in the absence of ground truth. Such models are built by taking monocular images as input and projecting them on nearby view to minimize reconstruction error.

[*]Equal contribution [1]Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York [2]Department of Mathematics, Courant Institute of Mathematical Sciences, New York University, New York [3]Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York. Correspondence to: Sree Gowri Addepalli <sga297@nyu.edu>, Amartya Prasad <ap5891@nyu.edu>, Sree Lakshmi Addepalli <sla410@nyu.edu>.

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

Pseudo Lidar(Yan Wang & Weinberger, 2020) is a technique for 3D object detection that generates point clouds using Depth Images. These point clouds could serve as a proxy Lidar data and be supplied to SOTA 3D object Detection Models like (Qi et al., 2017). We however could not proceed with this due to absence of velodyne(lidar) calibration data.

MonoLayout(Mani et al., 2020) is a ResNet based Model that can be trained to predict the top down view for an image using just monocular images. These top down views could then be passed to SOTA Object Detection/Sematic Segmentation Models. Despite not being a very large model, it has proved to help improve performance significantly. Due to poor quality training labels generated using our own methods, we decided to drop this technique.

2. Methods

We review here our architecture and models used for end to end pipeline for object detection and road map prediction. Road Map prediction is the task of classifying each pixel in the 800 X 800 matrix to either true, which indicates presence of road or false which indicates the absence of road. Whereas for object detection, we need to be able to do 3D object detection of the images present in the sample.

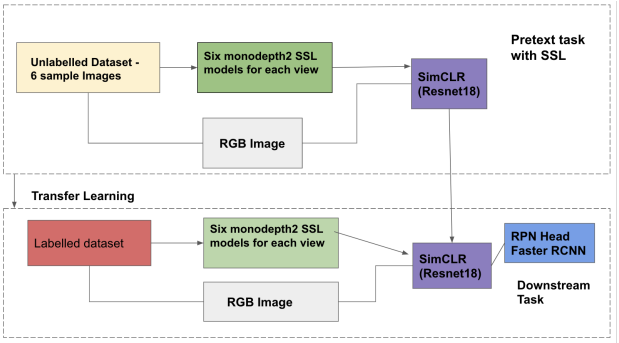

Figure 1. Architecture Diagram End to End pipeline

2.1. Baseline Through Supervised Learning

We first build our baseline for both these tasks through supervised learning to understand the model performance through only the labelled data set.

2.1.1. Road Map Layout Prediction with ResNet18

We send 6 images of the sample in our labelled data set as 18 channels which allows the network to learn the temporal dependency. We then pass this to ResNet18 with fully connected layer with binary cross entropy loss function and Adam optimizer for binary classification to achieve accuracy of 0.74 on the validation set and 0.73 on the test set.

2.1.2. Object detection and classification with Faster RCNN

We use image stitching by combining all these images into one image and sending these to this network. We stitched the 6 images into one single image and resized it to 800 X 800. Since Faster RCNN, like other SOTA Object Detection models expects bounding box values to be in pixel coordinates we transformed our x coordinate values by x*10 + 400 and y coordinate values by -y*10 +400. Also, the prediction of faster RCNN bounding boxes is of size [1*4] which expects a target output of the same size, to which we convert the target bounding box from [2*4] to [1*4]. Also, In order to get back the box coordinates in the original scale(cartesian coordinates in meters), we apply the inverse of these transforms to the output of our model.

2.2. End to End framework

We then move away from traditional supervised learning approach towards self supervision techniques to exploit the amount of unlabelled data present in comparison with labelled data. We also try to inculcate depth of each pixel in our model in an attempt to improvise which we feel we missed during supervision technique. We train two different self supervised models to generate representations which could be useful for our downstream task as presented in the architecture below.

2.2.1. Pretext task with Simple Framework For Contrastive Learning Of Visual Representations

We use SimCLR which is a framework of SOTA in Self-Supervised and Semi-Supervised Image Training, as a pretext task on our unlabelled data. It provides a model that learns representations by maximizing agreement between differently augmented views of the same data example via a contrastive loss in the latent space. It consists of the following modules:
1.Data Augmentation: This generates two correlated views of same image using random cropping, random color distortions and random Gaussian blur.
2. Base Encoder: Here, the output of last average pooling layer is used for extracting representations from images using resnet18.
3. Projection Head: These linear layers help learn embeddings of sizes 512 and 64 respectively.
4. Contrastive Loss Function: This function takes as input, a set of examples (xi and xj) and aims to identify xj in set for given xi. (called NT-Xent - the normalized temperature-scaled cross entropy loss).
5. LARS optimizer - It is called Layer wise Adaptive Rate

scaling used for optimization.

2.2.2. Self supervised Monocular Depth estimation

We use this architecture on our unlabelled data set to train six models for each camera view considering the camera intrinsic of each camera and apply to get the depth image for each image in the sample.

Figure 2. Original Image from sample

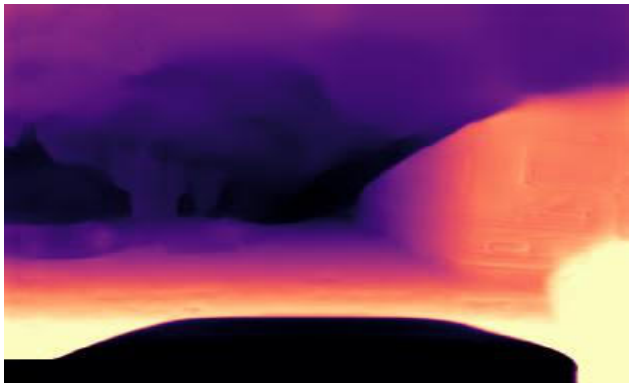

Figure 3. Depth Image from sample

2.2.3. Downstream - Road Map prediction

For the downstream task we use the representations learned from SimCLR for Road Map prediction task. Conjoined learned embeddings with ASPP (Atrous Spatial Pyramid Pooling) Head of DeepLabv3, a SOTA semantic segmentation model for road prediction. ASPP probes an incoming convolutional feature layer with filters at multiple sampling rates and effective fields-of-views, thus capturing objects as well as image context at multiple scales. We first freezed the weights of our upstream model and only trained on our DeepLab layers with batch size of 2. With this we got a 0.66 validation accuracy. We also conjoined learned embeddings with Linear classifier to achieve an validation accuracy of 0.64. We also fine tuned our entire upstream and downstream model, but couldn't train it for many epochs due to heavy model and unavailability of high end multiple GPUs.

2.2.4. Downstream - Object Detection

For object detection downstream task, we use the representations learned from SimCLR to the Regional Proposal Network (RPN) head of the Faster RCNN, a 2D object Detection model for bounding box prediction. We also combined the upstream prediction heads to RetinaNet. We faced issues while merging the upstream model with RPN head due to immutable number of input channels in Faster-RCNN/RetinaNet (Tsung-Yi Lin, 2018).

2.2.5. Combining SimCLR with Monodepth2

Using the above model we pass images with depth images together as 24 channels (6*(3+1)) to generate 512 dimension embedding. Since this model was very heavy we could train only for 5 epochs with Adam optimizer and a batch size of 2, as internally every image had to be converted to it's depth image.

2.2.6. Custom IoU Loss

For the object detection task, we used a custom Loss function, that is, $L = L_{\text{Object Detection and Classification Losses}} + L_{IoU}$ where L_{IoU} represents - log (IoU). This would allow us to directly optimise the IoU. We expect this to lead to better performance on IoU metrics.

3. Metrics for evaluation

For the Road Map task, we use Threat score as our evaluation metric as described in the 'project description' pdf provided to us by instructors. For the Bounding Box Prediction task, we use Intersection over Union(IoU) as our evaluation metric .

4. Results

See Fig.4 and Fig.5

5. Conclusion

Supervised methods help us set decently strong baselines for these tasks. For the road map task, we trained for 100 epochs. We expect better performance if trained for more epochs. For the object detection task through supervised learning, while we did stitch the images an resized to 800*800 as well

Model	Validation Accuracy (Threat score)
Supervised Resnet18	0.74
SimCLR+ Linear classifier	0.64
SimCLR+ DeepLabv3 ASPP	0.66
Supervised Resnet18 + MonoDepth2	Trained 6 Monodepth models only
SimCLR+ MonoDepth2+ DeepLabv3 ASPP	Trained SimCLR+Monodepth models only (Very Heavy) (Upstream only)
SimCLR+ MonoDepth2+ Linear classifier	Trained SimCLR+Monodepth models only (Very Heavy)((Upstream only)

Figure 4. RoadMap Layout Results

Model	Validation Accuracy (IoU)
Supervised FasterRCNN	0.014
Supervised RetinaNet	0.012
Supervised FasterRCNN + MonoDepth2(only Depth Images)	Trained 6 Monodepth models only
Supervised RetinaNet + MonoDepth2	Trained 6 Monodepth models only
SimCLR+ FasterRCNN	Trained SimCLR and faced merging issues
SimCLR+ MonoDepth2+ FasterRCNN	Trained SimCLR+Monodepth models only (Very Heavy)
SimCLR+ MonoDepth2+ RetinaNet	Trained SimCLR+Monodepth models only (Very Heavy)

Figure 5. Bounding Box Results

as transformed boxes to pixel coordinates, we did not rotate the stitched 800*800 image by 90 degrees to the right. This would have ensured that the input has exactly the same orientation as the output. Furthermore, one other technique we ought to have tried, which we believe could have helped performance a lot is : find the minimum and maximum bounding box lengths and widths, ie $l_{min}, l_{max}, w_{min}, w_{max}$ and then constrain the bounding box coordinates(ie the box predictor part of the RPN head) in the form $[x, y, x + (1 - t_1)l_{min} + t_1 l_{max}, y + (1 - t_2)w_{min} + t_2 w_{max}]$ such that the learnable parameters become $x, y, t_1 and t_2$ where t_1 and t_2 could be sigmoid outputs to ensure they always lie between 0 and 1. This would ensure that predicted boxes aren't too large or too small and could make learning easier. For the road map task, pre-training SIMCLR followed by fine-tuning led to a drop in performance vis a vis a purely supervised approach. We believe this may have been due to overfitting. The SIMCLR part of the model may have learnt too many high level features of the unlabelled data and thus couldn't perform well on the downstream task. Moreover, the small batch size used in most of models here could have affected performance on generating better representation vectors in our upstream task, because learning good representations through contrastive approaches would naturally require the two samples to be fairly different from each other, something that would be hard to ensure with batch size of 2. A larger batch size that includes images from different scenes could lead to better representation vector, but we were unable to train on multiple GPUs due to lack of high end GPU availability.

For the object detection task, we believe that passing depth maps as a fourth channel to SIMCLR would enable the model to learn depths and thereby and lengths of cars through contrastive learning. Conjoining those learned embeddings would potentially make it far easier for Faster-RCNN, or other 2D Object Detection models to detect boxes.

References

SimCLR code. https://github.com/Spijkervet/SimCLR.

Godard, C., Mac Aodha, O., Firman, M., and Brostow, G. J. Digging into self-supervised monocular depth prediction. October 2019.

Liang-Chieh Chen, George Papandreou, I. K. K. M. and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. May 2017.

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.

Qi, C. R., Liu, W., Wu, C., Su, H., and Guibas, L. J. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017.

Shaoqing Ren, Kaiming He, R. G. and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. January 2016.

Tobias Pohlen, Alexander Hermans, M. M. B. L. Full-resolution residual networks for semantic segmentation in street scenes. December 2016.

Tsung-Yi Lin, Priya Goyal, R. G. K. H. P. D. Focal loss for dense object detection. Feb 2018.

Yan Wang, Wei-Lun Chao, D. G. B. H. M. C. and Weinberger, K. Q. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. February 2020.