Monocular 3D Object Detection by Leveraging Self-Supervised Visual Pre-training

Can Erhan
ITU Artificial Intelligence and Data Science Research Center
Istanbul Technical University, Turkey
erhanc at itu.edu.tr
*

Anil Ozturk
Eatron Technologies
Istanbul Technical University Ari Teknokent 6, Turkey
anil.ozturk at eatron.com

Burak Gunel
Eatron Technologies
Istanbul Technical University Ari Teknokent 6, Turkey
burak.gunel at eatron.com

Nazim Kemal Ure
ITU Artificial Intelligence and Data Science Research Center
Istanbul Technical University, Turkey
ure at itu.edu.tr †

Abstract

Precise detection of 3D objects is a critical task in autonomous driving. Monocular 3D object detection problem is defined as predicting 3D bounding boxes in the metric space with a single monocular image. Most 3D detectors follow the standard pre-training strategy using the supervised ImageNet dataset, which is created for a dissimilar classification task. In this paper, a simple and effective pre-training strategy is proposed for monocular 3D object detection problem, without requiring any human supervision and annotated data. A dense depth estimation pretext task is incorporated into the pre-training pipeline by taking advantage of self-supervised learning. Experiments show that transferring the pre-trained weights to the detection network increases the performance in 3D object detection and bird's eye view evaluations up to 25% improvement rate with respect to the baseline networks that are based on ImageNet pre-training. This strategy has the potential of being applicable to other 3D object detection methods without any modifications to the existing algorithm design.

*C. Erhan is with Department of Computer Engineering
†N.K. Ure is with Department of Aeronautical Engineering

Machine Learning for Autonomous Driving Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

1 Introduction

Many important tasks in autonomous driving such as prediction and planning require a robust perception pipeline in the metric 3D space. An important component of a perception stack is object detection. Thanks to improvements in deep neural networks, the accuracy of state-of-the-art 2D object detection methods almost reached to human level (Lin et al. [2020], Zhou et al. [2019]). However, these methods only provide the location of the objects in 2D image plane, which is not sufficient to represent objects in 3D world where they actually exist. Monocular 3D object detection problem relies on predicting 3D bounding boxes in the metric space with a single monocular image. Localizing the objects in 3D is an ill-posed problem, since the critical depth information cannot be measured directly from the monocular images. Despite the difficulties in algorithm design, there is an increasing trend on monocular 3D object detection research in the autonomous driving community.

Monocular 3D object detection methods mostly depend on supervised data. In recent years, there is a growing number of publicly available autonomous driving datasets (e.g. KITTI (Geiger et al. [2012]), NuScenes (Caesar et al. [2020]), A2D2 (Geyer et al. [2020]), H3D (Patil et al. [2019]), Open Waymo (Sun et al. [2020])) which contain thousands of images and annotated 3D bounding boxes. The annotation process is extremely costly since it requires accurate depth sensors like LIDAR as well as intense human supervision. In the autonomous driving context, 3D bounding boxes usually have 7 degrees of freedom (DoF) that are center location (x, y, z), dimensions (w, h, l) and heading angle (θ). On the other hand, conventional 2D objects have 4 DoF, 2D location (x, y) and sizes (w, h), which is easier to annotate. These difficulties in the data annotation process causes 3D object detection process to be of high-cost and not scalable.

Pre-training is a very common practice in many computer vision tasks in order to reduce data size as well as convergence time. In a generic pre-training process, models are first pre-trained on large-scale datasets to learn visual features, and then fine-tuned on downstream tasks such as object detection. In particular, the supervised ImageNet (Deng et al. [2009]) pre-trained model has become *de facto* standard for years. One method to obtain a pre-trained model is to utilize self-supervised learning (Jing and Tian [2020]), which is basically a subset of unsupervised learning where the input data provides the supervision. Self-supervised learning methods are proposed to learn visual features from large-scale unlabeled data through a range of pretext tasks with no human supervision.

Inspired by the idea that problem related visual features learnt by a pretext task can actually benefit the downstream task, we propose a strategy by engineering the pre-trained model with self-supervision in the context of 3D object detection for autonomous driving. In order to evaluate the strategy, we first train a dense depth network in self-supervised manner, and then use the pre-trained model to solve a monocular 3D object detection problem. The experiments are conducted on well-known KITTI dataset (Geiger et al. [2012]). It is shown that initializing the detection network with the pre-trained model obtained by the dense depth estimation network improves the performance in 3D detection and bird's eye view evaluations by a wide margin. This strategy can be applied to other 3D object detection methods without any modification in the algorithm design.

Our contributions are summarized as follows:

- We propose a simple and effective pre-training strategy to improve 3D monocular detection performance without requiring any human supervision and annotated data.
- Our strategy significantly outperforms the supervised ImageNet baseline up to 25% improvement rate when transferring the pre-trained weights to 3D object detection task.

2 Related Work

Monocular 3D object detection: Several works are based on lifting 2D detection to 3D, with assuming that perspective projection of a 3D bounding box should fit tightly with its 2D detection window (Mousavian et al. [2017], Liu et al. [2019]). The main idea of these methods are to regress the 3D bounding box parameters from the image patch enclosed by the 2D bounding box. However, they require an additional 2D object detector in order to achieve end-to-end 3D object detection. One notable approach for 3D object detection is based on detecting keypoints on the monocular images. The key assumption in this category is that vehicles are rigid bodies with distinctive common parts that can be used as keypoints for detection. Most studies (Liu et al. [2020], Tang et al. [2020], Li et al.

[2020], Chen et al. [2020a]) utilize various 2D object detection frameworks such as CenterNet (Zhou et al. [2019]) and RetinaNet (Lin et al. [2020]) to construct single-staged 3D detection pipelines in contrast to the region proposal networks in lifting 2D to 3D approaches. Other works along this line use 3D CAD models to increase the number of keypoints regarding the shape of the vehicles (Chabot et al. [2017], Ansari et al. [2018]). Although they use a semi-automatic way to label 3D keypoints by placing CAD models in the 3D bounding box ground truths, the annotation on a large scale is very complex and time-consuming. All these methods probably follow the ImageNet pre-training strategy, which is not specifically stated in their papers.

Depth estimation: In order to obtain depth straight from the images, a convolution based depth estimation network has been used by Eigen et al. [2014], which is a primordial example of this type networks. In order to enhance the information exchange between decoder and encoder parts of the neural network, substantial developments have been made by dense pixel prediction networks in time (Shelhamer et al. [2017a]). To tackle with the spatial reduction which happens in down-sampling, a concept called fractional pooling (Graham [2015]) has been proposed. Increasing the learning capabilities of the pooling in the network which resulted in better results has been introduced by Lee et al. [2015]. Apart from that, self-supervised depth estimation techniques draw attention since direct supervision is difficult and requires precise range sensors. Recently, the usage of ego-motion enables self-supervised monocular depth estimation in image sequences (Pillai et al. [2019], Guizilini et al. [2020]).

Pre-training: Transfer learning (Torrey and Shavlik [2009]) increases the performance of a deep learning model between different sample sets in the same domain or different domains. When the model with trained feature extractors from a domain starts training on another domain, the desired point in training is reached faster. In some deep learning tasks, the transfer learning method is used as pre-training phase. The pre-training phase noticeably improves the performance of most segmentation (Shelhamer et al. [2017b]) and 2D object detection (Lin et al. [2020], Zhou et al. [2019]) algorithms. If there are no ground-truth labels of the data, pre-training can be done with self-supervised methods. Reconstruction based loss functions are generally used (Doersch et al. [2015], Pathak et al. [2016], Goodfellow et al. [2014]) for self-supervised training in visual tasks. Alternatively, contrastive-loss can also be used. Contrastive learning is feeding the positive images to the model by pairing them with augmented versions of themselves and negative images with different images. The contrastive learning method is used in most studies with successful results (Chen et al. [2020b], Wu et al. [2018], Xie et al. [2020]).

3 Self-supervised Pre-training Strategy

Our proposed strategy relies on engineering the pre-trained model that is used to initialize the 3D object detection network. Most 3D detectors follow the standard pre-training strategy with the supervised ImageNet (Deng et al. [2009]) (Fig. 1a) which is actually trained to solve an image level recognition task. Even though ImageNet pre-trained model provides better convergence than the random initialization strategy, it might not be a proper initializer for dense prediction tasks such as object detection (Wang et al. [2020]). In other words, a model trained to solve a recognition task might not learn the necessary features to localize the objects in the image. Similar to this idea, we use a self-supervised dense depth estimation pretext task to force the model learn depth related features that are critical for 3D object detection.

The high level pipeline of the proposed pre-training strategy is illustrated in Fig. 1b. The dense depth auto-encoder network is first trained on top of the standard ImageNet, and then the encoder part is detached to obtain the pre-trained model which later serves the detection network. By training the depth network, it is expected that the encoder part learns a compact representation of the visual depth features. The encoded features in the latent spaces allow to make relationships between the objects present in the image and their real world depth. It is important to note that the training of the depth auto-encoder is performed by completely self-supervised manner. Neither annotated data nor human supervision is required to train the model. This strategy is simple and effective in order to improve the performance of a 3D object detection method which regresses the depth without any modification in the algorithm design.

3.1 Detection Network

Most monocular 3D detection networks basically try to solve the inverse projection problem regressing the depth of the target object classes. According to the assumption that the encoded features provide better representation for the rigid objects and their actual depths, we select a simple and straightforward 3D detector. The SMOKE network (Liu et al. [2020]) is engineered as an extension to CenterNet (Zhou et al. [2019]) which is a keypoint-based and anchor-free object detection framework. SMOKE is also single-staged and does not contain any hand-crafted features. There are two separate prediction heads to perform the object classification and 3D bounding box regression. Estimated keypoints are classified by point-wise focal loss (Lin et al. [2020]) on the downsampled feature maps. On the other hand, the regression loss is calculated as the l_1 distance of the predicted and ground-truth 3D bounding boxes.

Similar to CenterNet, SMOKE uses a hierarchical layer fusion backbone DLA-34 (Yu et al. [2017]) to extract feature maps. Basically, DLA-34 has an encoder part to downsample the input image to different scales, and a decoder part to upsample with an iterative way to combine those layers. It enables to learn a combination of low and high level features with long skip connections. The rough detection network architecture as it is used in SMOKE is illustrated in Fig. 1. The colored shape on the left-hand side indicates the encoder part which is initialized with the pre-trained models. The weights other than the base part are initialized randomly. It is important to note that this strategy is not suitable for the 3D detection networks (e.g. Mousavian et al. [2017], Liu et al. [2019]) which are fed with the image patches cropped from their 2D detection proposals, because the depth pre-trained model accepts a complete scene image.

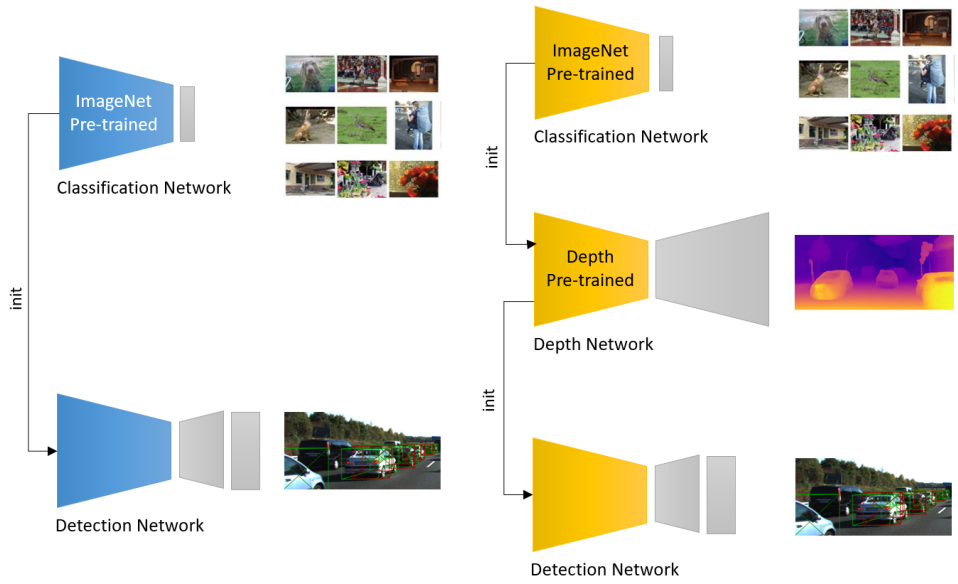


Figure 1: The high level pipeline to illustrate (a) the standard ImageNet and (b) the proposed pre-training strategies. The trapezoids on the left hand-side of the networks indicate the feature extractor backbones whereas the arrows between the backbones denote the initialization order. In the proposed strategy, a dense depth estimation network is first trained with self-supervised manner, and then the detection network is initialized with the pre-trained model obtained in order to learn critical depth features to solve 3D object detection problem.

3.2 Depth Network

The proposed pre-training strategy relies on obtaining a dense depth encoder as a pre-trained model. However, estimating dense depth maps via direct supervision is difficult for a pretext task since it requires precise range sensors and cross-calibration to collect ground-truth data. In order to obtain a pre-trained model from a dense depth network, a self-supervised methodology is employed. PackNet

(Guizilini et al. [2020]) uses geometrical constraints and camera motion on image sequences as the source of supervision. In the training pipeline, there is a pose network that is trained along with the depth network simultaneously. The self-supervision loss is constructed with the source and the target image that is synthesized from the ego-motion information. It is assumed that sequence of images are available during the training.

Instead of using packing and unpacking structures introduced in the original work, we engineer the depth auto-encoder network such that it can be integrated with the DLA-34 base part. The skip connections in the depth decoder are aligned to the ones in the detection network. In training, ResNet-50 (He et al. [2015]) is employed for ego-motion network. It is noted that the depth encoder network (i.e. DLA-34 base part) is initialized with the standard ImageNet pre-trained model as depicted in Fig. 1b.

The training data that is used for dense depth estimation network is the KITTI Raw Dataset (Geiger et al. [2012]). There are several sequences in various categories (e.g. city, residential, road, campus) with a total number of 39810 images (Eigen image splits Eigen et al. [2014], Guizilini et al. [2020][3]). The whole depth network is trained with self-supervised manner and no annotated data is used. Qualitative examples of dense depth estimation results are shown in Fig. 2.

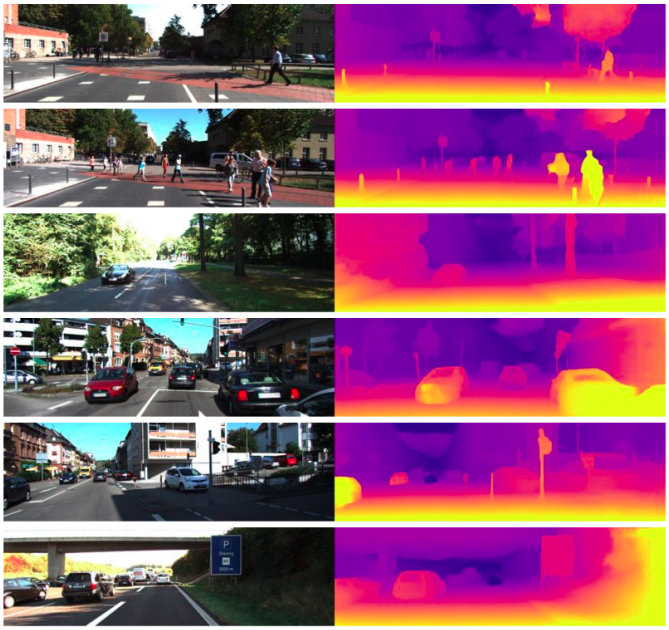


Figure 2: Qualitative examples of dense depth estimation results obtained from the depth network trained with self-supervised manner.

4 Experiments

In order to evaluate the proposed pre-training strategy, extensive experiments are conducted on the detection network. In the experiments, we first train the dense depth estimation network to obtain the pre-trained model, then the model is used to initialize the detection network. The training and evaluation results are presented as the comparison of the baseline ImageNet and the proposed pre-training strategies. For the sake of simplicity, our strategy is denoted as DepthNet for the rest of the paper.

4.1 Dataset

The detection network is trained by using KITTI Object Dataset (Geiger et al. [2012]) which is a well-accepted standard for evaluating 3D object detection performance. It contains 7481 images

[3]https://tri-ml-public.s3.amazonaws.com/github/packnet-sfm/splits/KITTI/eigen_zhou_files.txt

for training and 7518 images for testing. There are 4 types of evaluation categories which are 3D detection, bird's eye view, 2D detection and orientation. For each evaluation, they are divided into easy, moderate and hard cases based on the height of the 2D bounding box of the objects, truncation and occlusion levels. The ground truth of the test set is not released since the competition is still going on. In order to make the results comparable to the other detection methods in the literature, the actual training set is split into 3712 training and 3769 validation examples as suggested in Chen et al. [2016][4].

4.2 Training

SMOKE network (Liu et al. [2020]) is used to evaluate the proposed pre-training strategy. The training parameters are listed in Table 1. We train (i.e. fine-tune) the network which backbone is either frozen or non-frozen separately. All training parameters for both approaches are set as the same except the learning rate decay steps and number of epochs which are 50 and 100 respectively. The batch size is set to 16 with the learning rate of $1.25\text{e}{-4}$ initially and drops two times with a factor of 10. The original image resolution is used and padded to 1280×384. Flipping, shifting and scaling augmentation techniques are applied to input images. Three training sessions are carried out to provide consistent results. Due to the technical issue on the implementation platform, deformable convolution (DCNv2) and group normalization (GN) layers are switched to standard convolution and batch normalization layers.

Table 1: Detection network training parameters

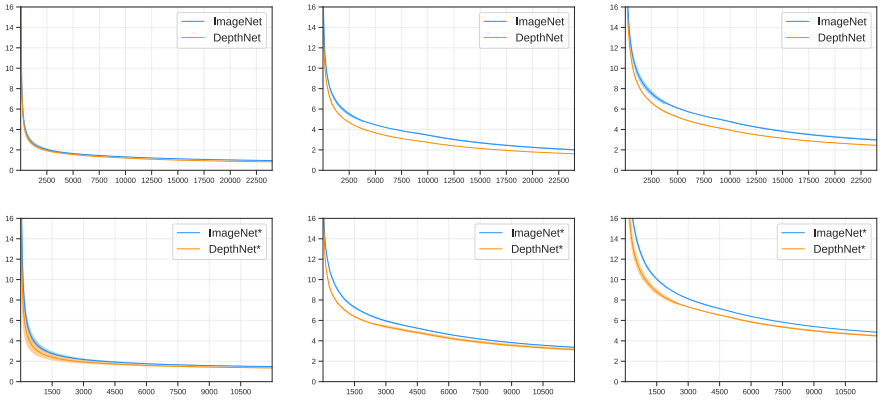
Optimizer	Adam
Initial learning rate	$1.25\text{e}{-4}$
Learning rate decay factor	0.1
Batch size	16
Confidence threshold	0.25
Input resolution	1280×384
Non-frozen backbone:	
Training epoch	100
Learning rate decay steps	40, 70
Frozen backbone:	
Training epoch	50
Learning rate decay steps	20, 35

Fig. 3 presents the moving averaged loss curves for the detection networks initialized with the proposed DepthNet and the baseline ImageNet pre-trained models. Classification loss is a metric that can be used to evaluate the success of the deep learning model in the classification task. Looking at frozen and non-frozen models, it can be seen that both types have similar convergence structures. However, for the frozen architecture, the model with the pre-trained DepthNet converges slightly faster.

Regression loss is another metric to measure the success of bounding-box regression process. In frozen and non-frozen combinations, the large gaps between the loss curves indicate the pre-trained DepthNet converges significantly faster than the baseline ImageNet. This can be also interpreted as the encoded depth features in the DepthNet improve the bounding box regression compared to classification.

Total loss represents the combination of classification loss and regression loss in this context. The characteristics of both losses can be seen in the total loss plots. The distinct performances of the models in the regression loss change are also effective in the structure of the total loss. Based on the interpretation made on the plots, the results show that the training session with the pre-trained DepthNet backbone brings more successful results in a shorter time.

[4]https://xiaozhichen.github.io/files/mv3d/imagesets.tar.gz



(a) Classification loss (b) Regression loss (c) Total loss

Figure 3: Classification, regression and total loss with respect to training iteration. Total loss refers to the combination of classification and regression loss. Three training sessions are carried out to draw the plots for consistency. The thick lines are the average of the three sessions. The shaded areas represent the minimum and maximum value ranges. Plots are created using moving average to reduce complexity. Note that $*$ denotes the pre-trained models that are frozen during the training.

5 Results

Standard KITTI evaluation script is used to calculate the mAP scores for only car class over 40 recall points with intersection over union (IoU) threshold of 0.5. The final results are provided by averaging the output of three training sessions for consistency. Table 2 lists the mAP scores obtained for different evaluation methods which can be divided into two categories: (1) 3D object detection and bird's-eye view where depth estimation is essential, and (2) orientation where visual appearance is significant. 2D detection can be ignored since it is calculated as projecting the 3D points onto the images. There is an additional row showing the improvement rate indicates the percentage that mAP scores are changed with respect to the baseline ImageNet.

Table 2: Validation split performance on $AP|_{R_{40}}$ @ $T_{IoU} = 0.5$ w.r.t. the car class. $*$ denotes frozen pre-trained models. Improvement rate indicates the percentage that the mAP scores are changed w.r.t. the baseline ImageNet.

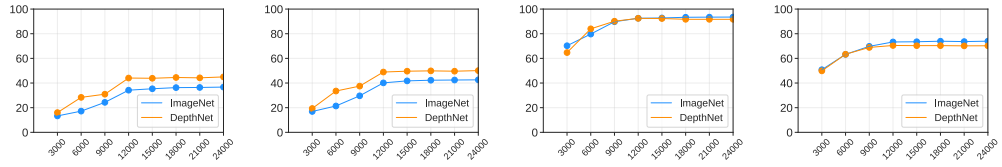
	3D object detection			Bird's eye view			2D object detection			Orientation		
	Easy	Mode	Hard	Easy	Mode	Hard	Easy	Mode	Hard	Easy	Mode	Hard
ImageNet$*$	13.03	8.98	8.19	15.91	11.41	10.18	68.46	67.26	59.58	53.87	52.94	47.41
DepthNet$*$	**21.49**	**14.21**	**13.01**	**26.40**	**17.57**	**16.60**	**77.94**	**76.37**	**69.63**	**60.16**	**59.64**	**54.97**
Improvement Rate	+64%	+58%	+58%	+66%	+54%	+63%	+14%	+13%	+17%	+12%	+13%	+16%
ImageNet	35.92	26.04	23.56	42.32	30.94	27.12	**92.67**	**87.56**	**80.15**	**73.85**	**69.50**	**63.84**
DepthNet	**45.00**	**30.40**	**26.35**	**49.94**	**33.82**	**30.00**	91.15	86.02	78.68	69.53	67.39	61.65
Improvement Rate	+25%	+17%	+12%	+18%	+9%	+11%	-2%	-2%	-2%	-6%	-3%	-3%

The comparison of frozen backbones is important for showing how the encoded features in the pre-trained models are able to improve the detection performance without any update. In this comparison, it is obvious to see that the proposed DepthNet surpasses ImageNet in all evaluations. In 3D object detection and bird's eye view, an average improvement rate over 60% is achieved, whereas an average improvement rate over 15% is achieved in 2D object detection and orientation. In both cases, the encoded depth features in the pre-trained DepthNet dramatically enhance the performance of the networks.

By the experiments conducted on non-frozen models, the aim is to get the highest score as possible. As it can be seen in the Table 2, the improvement rates compared with frozen backbone scenarios are lower. The actual mAP scores are nearly doubled in 3D object detection and bird's eye view where

DepthNet outperforms the baseline ImageNet with up to 25% improvement rate. However, there are minor drops in 2D object detection and orientation where the features representing visual appearance is more important.

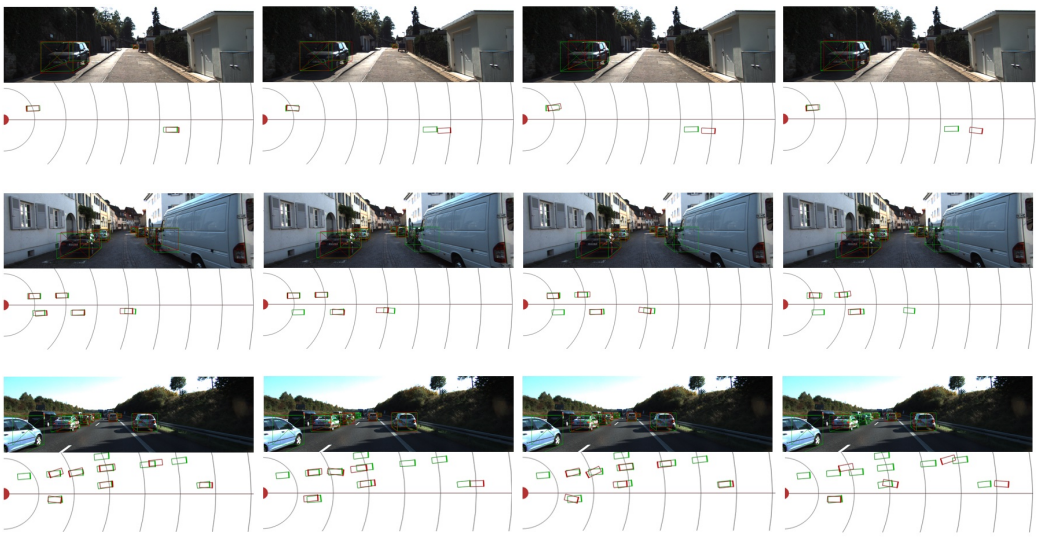
For each evaluation method, the scores with respect to training iterations are plotted in Fig. 4. For both frozen and non-frozen architectures, the biggest gap between the results is achieved in 3D object detection and bird's-eye view evaluations where depth estimation is critical. In Fig. 4a and Fig. 4b, it can be clearly seen that DepthNet approach quickly take the lead in mAP scores from the beginning of the training, and keep its score advantage over the whole training. In Fig. 4c and Fig. 4d, which are 2D object detection and orientation, DepthNet go head to head with ImageNet during the training. Since these results are obtained from a single training settings, it might be tuned that DepthNet outperforms the baseline ImageNet. To address this problem, additional experiments are performed on the effect of hyper-parameters, which is described in Sec. A.1.



(a) 3D object detection (b) Bird's eye view (c) 2D object detection (d) Orientation

Figure 4: mAP/iteration curves w.r.t. the car class for different evaluation methods. The scores are obtained with non-frozen backbones on the validation split.

Several qualitative examples on the validation split are displayed in Fig. 5. For better visualization and comparison, the bird's-eye view representation with respect to pre-training strategies is drawn. The green boxes drawn around the vehicles represent the ground-truth information whereas the red ones are the detection results for the relevant strategy. When Fig. 5c and Fig. 5d are compared, it is possible to see the difference between DepthNet and ImageNet on frozen backbones even more clear. ImageNet fails to detect most of the vehicles in the scene and when it detects, there is a huge error and the boxes do not match. However, in DepthNet the predicted boxes match nicely with the ground truth.



(a) DepthNet (b) ImageNet (c) DepthNet$*$ (d) ImageNet$*$

Figure 5: Qualitative examples on the validation split w.r.t. pre-training strategies. The green boxes drawn around the vehicles represent the ground-truth information whereas the red ones are the detection results for the relevant strategy. Note that $*$ denotes frozen pre-trained models.

6 Conclusion

In this paper, a simple and effective pre-training strategy is proposed to solve monocular 3D object detection tasks. First, a dense depth estimation network is trained with self-supervised manner in order to encode compact depth features of the scene as well as the objects. The pre-trained depth encoder is engineered to initialize the 3D object detection network. The experiments are conducted as comparing the proposed strategy with the baseline ImageNet. The results show that this strategy significantly reduces the convergence time, and it improves the detection performance by a wide margin. Furthermore, the simplicity of the strategy allows it to be applied to other 3D object detection methods without hassle in the algorithm design.

Acknowledgment

This work is supported by the ITU BAP grant no: MOA-2019-42321 and Eatron Technologies.

References

T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. URL `http://arxiv.org/abs/1904.07850`.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020.

Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2d2: Audi autonomous driving dataset, 2020.

A. Patil, S. Malla, H. Gang, and Y. Chen. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9552–9557, 2019. doi: 10.1109/ICRA.2019.8793925.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2020.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. doi: 10.1109/TPAMI.2020.2992393.

A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3d bounding box estimation using deep learning and geometry. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5632–5640, 2017. doi: 10.1109/CVPR.2017.597.

Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. *CoRR*, abs/1904.12681, 2019. URL `http://arxiv.org/abs/1904.12681`.

Z. Liu, Z. Wu, and R. Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4289–4298, 2020. doi: 10.1109/CVPRW50498.2020.00506.

Yunlei Tang, Sebastian Dorn, and Chiragkumar Savani. Center3d: Center-based monocular 3d object detection with joint depth understanding. *CoRR*, abs/2005.13423, 2020. URL `https://arxiv.org/abs/2005.13423`.

Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. RTM3D: real-time monocular 3d detection from object keypoints for autonomous driving. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, volume 12348 of *Lecture Notes in Computer Science*, pages 644–660. Springer, 2020. doi: 10.1007/978-3-030-58580-8_38.

Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 12090–12099. IEEE, 2020a. doi: 10.1109/CVPR42600.2020.01211.

Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep MANTA: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. *CoRR*, abs/1703.07570, 2017. URL `http://arxiv.org/abs/1703.07570`.

Junaid Ahmed Ansari, Sarthak Sharma, Anshuman Majumdar, J. Krishna Murthy, and K. Madhava Krishna. The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera, 2018.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014. URL `http://arxiv.org/abs/1406.2283`.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, Apr 2017a. ISSN 2160-9292. doi: 10.1109/tpami.2016.2572683. URL `http://dx.doi.org/10.1109/TPAMI.2016.2572683`.

Benjamin Graham. Fractional max-pooling, 2015.

Chen-Yu Lee, Patrick W. Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree, 2015.

S. Pillai, R. Ambruş, and A. Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9250–9256, 2019. doi: 10.1109/ICRA.2019.8793621.

Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2482–2491. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00256. URL `https://doi.org/10.1109/CVPR42600.2020.00256`.

L. Torrey and J. Shavlik. Transfer learning. *Handbook of Research on Machine Learning Applications*, 01 2009. doi: 10.4018/978-1-60566-766-9.ch011.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, April 2017b. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2572683. URL `https://doi.org/10.1109/TPAMI.2016.2572683`.

C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015. doi: 10.1109/ICCV.2015.167.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. 2016.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020b. URL `http://proceedings.mlr.press/v119/chen20j.html`.

Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. doi: 10.1109/CVPR.2018.00393.

Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 574–591, Cham, 2020. Springer International Publishing.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training, 2020.

Fisher Yu, Dequan Wang, and Trevor Darrell. Deep layer aggregation. *CoRR*, abs/1707.06484, 2017. URL `http://arxiv.org/abs/1707.06484`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

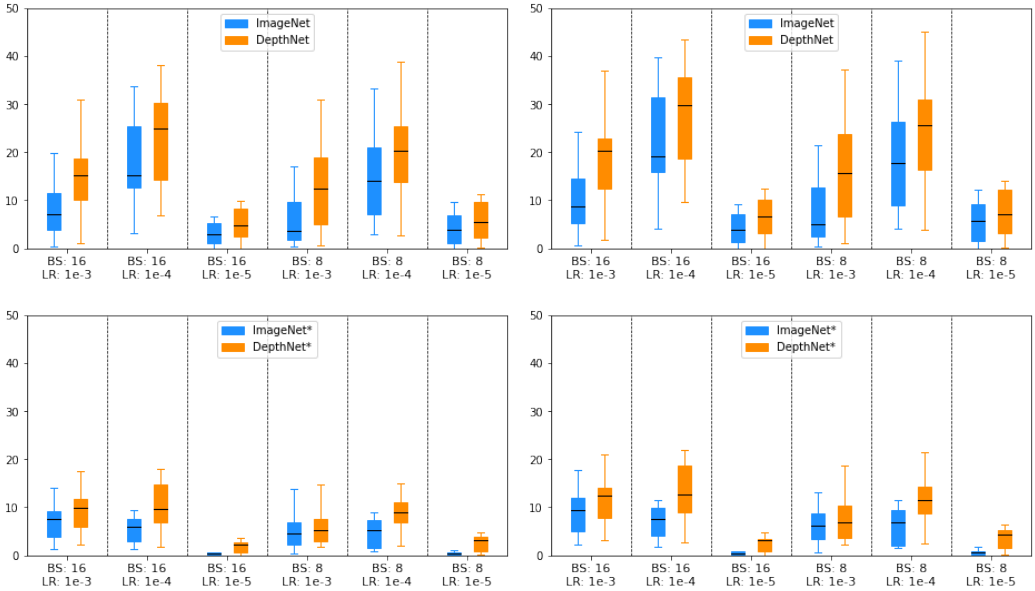
X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156, 2016. doi: 10.1109/CVPR.2016.236.

A Appendix

A.1 Hyper-parameter Analysis

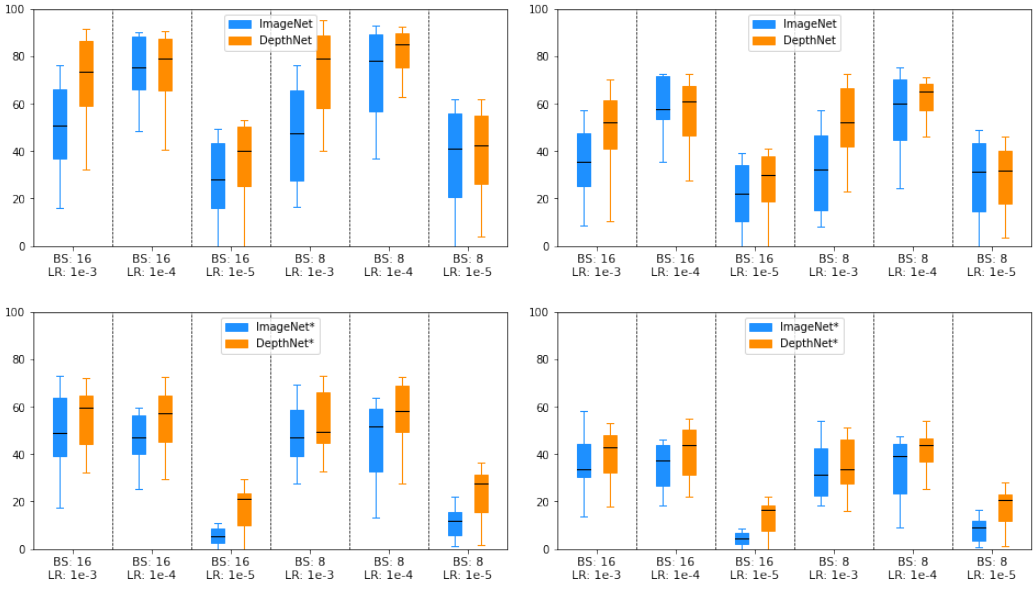
In this section, additional experiments are performed in order to analyze the effect of hyper-parameters on different evaluation strategies. The same dataset and splits are used. There are 6 different parameter sets including batch sizes (i.e. 8 and 16) and learning rates (i.e. $1e-3$, $1e-4$ and $1e-5$). The maximum iteration size remains the same for each parameter set for a fair comparison. The evaluation scores are taken at fixed intervals during the training. The trainings are repeated 3 times with different seeds. Fig. 6 is created by processing the evaluation scores seen along the 3 training sessions as a set for each parameter combination, and it illustrates the box plots constructed by these evaluation scores.

It can be seen that the trainings that include the pre-train phase with DepthNet give similar or better results when compared with ImageNet in most task types. The mentioned difference is particularly evident in the 3D object detection and bird's-eye view tasks where the depth information is crucial. The minimum points are very close or equal in the two networks is due to the closeness of the scores at the beginning of the training to 0. The fact that the median points of the box-plots of DepthNet are mostly higher than ImageNet indicates that DepthNet gives higher scores during the trainings. In other words, it experiences convergence faster. Furthermore, varying batch size has less effect to the evaluation scores compared to the effect of learning rate. Despite the parameter changes, it is obvious that the results obtained with different parameter sets present similar characteristics with each other, and DepthNet outperforms the baseline ImageNet in most cases.



(a) 3D object detection

(c) Bird's eye view



(b) 2D object detection

(d) Orientation

Figure 6: mAP/hyper-parameters plots w.r.t. the car class for each evaluation method. The scores are obtained with different parameter sets on the validation split ($AP|_{R_{40}}$ @ $T_{IoU} = 0.5$). Only the *easy* part of scores are plotted for simplicity. Each box is constructed with the evaluation scores taken at fixed intervals. Three training sessions are carried out for consistency. While the median is represented by the horizontal line inside the boxes, the upper edge of the boxes indicates the highest score obtained. Note that $*$ denotes frozen pre-trained models.