CVPR
#0033797411

Toward Robust and Efficient for Autonomous Driving

Anonymous CVPR submission

Paper ID 0033797411

Abstract

Due to the rapid progress of technologies and the high market potential value around the world, autonomous vehicles have recently been widely discussed and gradually become a mainstream research area. Basically, we can rely on Artificial Intelligence to deal with most of the above issues as robotics tasks. However, we note that there are two main challenges that make autonomous driving different from other robotic tasks. First, since there are many possible scenarios, it will lead to models becoming complex and computational costs becoming enormous as well. Second, they must make control and decisions more accurate and faster in very diverse conditions to realize the real-time application. This paper particularly explores the part of perception and control and introduces a more efficient, faster, and more robust algorithm to handle the above challenges. We mainly introduce three topics in this paper. First, we introduce a novel algorithm that treats the process of lane detection as a row-based selecting problem using global features to realize at extremely fast speed and challenging scenarios[32]. Second, we introduce a 3D object detection method called Stereo R-CNN, which extends Faster R-CNN for stereo inputs to simultaneously detect and associate objects in left and right images, by fully exploiting the sparse and dense, semantic and geometry information in stereo imagery[21]. Third, we introduce CURL, Contrastive Unsupervised Representations for Reinforcement Learning[17], to extract high-level features from raw pixels using contrastive learning and performs off-policy control on top of the extracted features.

1. Introduction

Autonomous vehicles are vehicles that can perceive their surroundings to make an accurate decision without any intervention from a human driver. These vehicles are also called driverless, self-driving, unmanned or robotic vehicles. It involves the integration of various techniques, including lane detection, object recognition and tracking, simultaneous localization and mapping, motion planning, vehicle control, and so on. In this paper, we will delve into the three parts of techniques, including lane detection, 3D objects detection, and control task in reinforcement learning, respectively. All of them play a major role in autonomous driving.

1.1. Lane Detection

Lane detection is a fundamental problem and has a wide range of applications in the long research history of computer vision. Recently, deep segmentation methods [11, 27, 29] can perform better results and have great success than the traditional computer vision method [1, 3, 39] which is mainly based on image processing algorithms to extract the features of lane lines. However, all current methods of lane detection for autonomous driving will encounter two main problems. The first problem of lane detection is computationally expensive. Autonomous vehicles are commonly equipped with multiple camera inputs and a lane detection algorithm is heavily executed as a fundamental component of autonomous driving. Another problem of lane detection is called no-visual-clue. Due to vehicle congestion on many roads, the lane line is blocked by the vehicles, and it is necessary to guess from the semantic information of the vehicles' location and environment. In this case, there is no visual information (e.g., color or shape of the lane) to guide the recognition of the lane line. Based on the above discussion, we introduce a novel lane detection formulation that makes a significant contribution to solve the computational cost and no-visual-clue problem [32]. This approach selects locations of lanes at predefined rows of the image using global features instead of segmenting every pixel of lanes based on a local receptive field, which significantly reduces the computational cost. It could also achieve good performance for the no-visual-clue problem because the formulation is conducting the procedure of selecting rows based on global features. In other words, compared with original deep segmentation methods, this method is selecting locations of lanes instead of segmenting every pixel and works on the different dimensions, which is ultra-fast. A lightweight version could even achieve 300+ frames per second with the same resolution, which is at least 4x faster than previous state-of-the-art methods. Besides, this method uses global features to predict, which has a larger receptive field than the segmentation formulation. In this way, the no-visual-clue problem can be addressed as well. This method achieves state-of-the-art performance in terms of both accuracy and speed on the challenging CULane dataset.

1.2. 3D Object Detection

In the previous research for 3D object detection, many approaches are regarding the LiDAR with the advantage of accurate depth information[8, 31, 45, 16, 22] or monocular camera with the advantage of low cost[6, 25, 40]. However, there are some issues for LiDAR (e.g., high cost and short perception range). On the other hand, the depth of monocular camera inference cannot guarantee the accuracy, especially for invisible scenes. Therefore, we will introduce a stereo-vision-based 3D object detection method[21]. A stereo camera can simulate human binocular vision, and therefore gives it the ability to capture three-dimensional images. Stereo vision also has the potential ability to provide a larger-range perception by combining different stereo modules with different focal lengths and baselines. Compared with the previously mentioned methods, there are several significant advantages. It provides more precise depth information than a monocular camera. It is low-cost while achieving comparable depth accuracy for objects with non-trivial disparities. Besides, this method simultaneously detects and associates objects for left and right images using the proposed Stereo R-CNN.

1.3. Control Task in Reinforcement learning

Reinforcement learning is considered to be a powerful AI paradigm that can be used to teach machines through interaction with the environment and learning from their mistakes. In other words, a reinforcement learning agent can perceive and interpret its environment and learn through trial and error with a pre-defined reward function to take appropriate actions finally. In the previous research, developing agents that can perform complex control tasks from high dimensional observations has been possible by combining the deep neural networks with reinforcement learning algorithms[24, 23]. However, it has been empirically observed that reinforcement learning from high dimensional observations such as raw pixels is sample-inefficient. It is widely accepted that learning policies from physical state-based features are significantly more sample-efficient than learning from pixels. In general, addressing the sample inefficiency of deep reinforcement learning algorithms can be classified into two streams of research. One is auxiliary tasks on the agent's sensory observations and the other is world models that predict the future. Therefore, we will introduce CURL, Contrastive Unsupervised Representations for Reinforcement Learning[17]. CURL uses a form of contrastive learning that maximizes agreement between augmented versions of the same observation, where each observation is a stack of temporally sequential frames. CURL significantly improves sample efficiency over prior pixel-based methods by performing contrastive learning simultaneously with an off-policy reinforcement learning algorithm. Based on the above discussion, we could make a little extend for this research. We can make use of this reinforcement learning algorithm to apply to the various control tasks in an autonomous vehicle with considering autonomous vehicles as an agent. From previous research in autonomous driving, we tend to separate three individual tasks, realizing recognition first and then planning and control. Besides, for the previous control problem, we tend to use some optimal control algorithm (e.g., model predictive control) to address this issue. However, model predictive control is essentially (slightly) less robust than reinforcement learning from the numerical point of view. Therefore, it is worth discussing to try to make use of CURL to perform complex control tasks from the physical state, such as continuous images as an input, to realize perception and control tasks simultaneously by combining the deep neural networks with reinforcement learning algorithms in further research.

2. Related works

In this section, we will briefly summarize recent works in relative fields.

2.1. Traditional Methods for Lane Detection

Traditional approaches usually solve the lane detection problem based on traditional computer vision algorithm to capture visual information. The main idea of these methods is to take advantage of visual clues through image processing like the HSI color model[34] and edge extraction algorithms[38, 43]. They may extract the features of lane lines, reduce the image channels, perform gray processing on the original image, and then use Canny algorithm or Sobel algorithm to edge the grayed image, extract some features of the acquired image, and then perform lane line fitting after extracting the lane.

2.2. Deep Learning Models for Lane Detection

With the advancement of deep learning, certain deep neural network-based approaches[15, 11] have demonstrated greater performance of lane detection. Typically, these strategies employ the same framework, recasting the problem as a semantic segmentation problem. For instance, VPGNet[19] proposes a unified end-to-end trainable multi-task network that concurrently tackles lane and road marking detection and recognition while being led by a vanishing point in inclement conditions. Spatial CNN[29] is a CNN-like technique for propagating information effectively at the spatial level.SCNN may be easily integrated into deep neural networks and trained end-to-end. They are particularly well suited for extended continuous shape structures or huge objects with strong spatial relationships but few visual cues, such as traffic lanes.

2.3. Object Detection

Using object detection, a computer vision approach, we can recognize and find objects in an image or video using this technique. A large number of researchers have concentrated on the prediction of 2D objects. However, 2D detection can only offer two-dimensional bounding boxes since it is limited to two dimensions. There are a plethora of applications for depth sensing and 3D information, such as collecting an object's size, location, and orientation in the world, among other things. Therefore, it's worth to delve deeper into 3D object detection.

2.3.1 LiDAR-based 3D Object Detection

The vast majority of existing 3D object recognition systems rely on LiDAR to offer exact 3D information; however, the raw LiDAR data is processed in a variety of different representations. To feed the structured convolution network, [8, 20, 41, 22, 16] convert the point cloud to a two-dimensional bird's eye view or front view representation where[8, 22, 16] gain more rich information by merging numerous LiDAR representations with the RGB images.

2.3.2 Monocular-based 3D Object Detection

The absence of precise depth information in monocular-based approaches is unavoidable. Assumption of ground-plane, a shape prior, contextual features, and instance segmentation from monocular images are used by [6] to generate 3D object suggestions. Through using geometric relationships between the 2D box edges and 3D box corners, [25] speculates a 3D box estimation method. When anticipating the series of critical points of regular-shape vehicles, [44, 4, 26] make use of the scarce information that is available.

2.3.3 Stereo-based 3D Object Detection

Using stereo vision for 3D object detection is used in just a few works. For example, it's a common practice for 3DOP[7] to utilize an energy function based on encoding object size priors, depth information (e.g., free space, point cloud density) as well as ground-plane priors to generate 3D proposals. Using the R-CNN technique, the 3D Proposals are then used to regress the object's posture and 2D boxes.

2.4. Self-supervised Learning

Self-supervised learning (SSL) is one of the machine learning techniques to train a model with labels inherently obtained from the data itself. Its goal is to develop rich representations of high-dimensional unlabeled data that may be used for a range of applications.

2.4.1 Contrastive Learning

Contrastive Learning is a method for learning representations that adhere to similarity constraints in a dataset, which is commonly structured by similar and dissimilar pairings. The objective of contrastive representation learning is to construct an embedding space in which comparable sample pairs remain near together and dissimilar sample pairs remain distant. This is frequently best described as conducting a dictionary lookup process, with the positive and negative values representing a collection of keys associated with a query (or an anchor). The choice of negatives can have a significant impact on how well the underlying representations are learned in contrastive learning.

2.4.2 Self-Supervised Learning for Reinforcement Learning

Auxiliary tasks, such as forecasting the future, are constructed on the foundation of previous observations and actions. The future prediction is made either in pixel space[12] or in latent space[28], depending on the methods. For the improvement of sample-efficiency of model-free reinforcement learning algorithms, [12, 33, 28] are some good examples of how to make better use of auxiliary tasks.

2.5. World Models for Sample-efficiency

This approach attempts to learn world models of an environment and then apply them to sample rollouts and planning. [35] proposed an early implementation of the generic principle, in which fictional samples generated from a learned world model are employed in addition to the agent's experience for sample-efficient learning. Another technique to increase sample efficiency is planning using a learned world model.

2.6. Sample-efficient Reinforcement Learning for Image-based Control

[42, 9, 18] have extensively utilized the DMControl suite[36] to evaluate sample-efficiency for image-based continuous control algorithms. [13] proposes that for Atari[2], the 100k interaction steps benchmark for sample-efficiency be used, as [14, 37] has done. CURL[17] incorporates self-supervision, contrastive learning, and the use of auxiliary tasks to enable sample-efficient reinforcement learning. It utilizes the DMControl suite and Atari Games benchmarks to determine sample-efficiency and get a good result compared with others.

3. Paper Implementation

In this section, we would like to implement one of the papers, Ultra Fast Structure-aware Deep Lane Detection[32], and attempt to introduce and evaluate its approach and experiment results in depth.

3.1. Method

The novel formulation and innovative lane structural losses are primarily introduced in this technique. A feature aggregation approach for high-level semantics and low-level visual information is also shown.

3.1.1 Definition of Formulation

From traditional segmentation methods, it is difficult to obtain effective context and global information. For this, SCNN[29] proposes a complex information strategy to greatly enhance the performance of the segmentation network, but it brings greater computational cost. Therefore, [32] introduces a new technique, utilizing global features to determine the correct position of lanes on each predetermined row. Lanes are represented in our formulation as a set of horizontal locations along predetermined rows, i.e., row anchors. Besides, gridding is the basic step for representing locations. The position of each row anchor is subdivided into several cells. Thus, lane detection may be regarded as the selection of certain cells over predetermined row anchors, as shown in Fig.1[32].

Besides, the loss function of formulation can be written as:

$$L_{cls} = \sum_{i=1}^{C} \sum_{j=1}^{h} L_{CE}(P_{i,j,:}, T_{i,j,:}) \tag{1}$$

in which $P_{i,j:}$ is the probability of selecting $(w+1)$ gridding cells for the i-th lane, j-th row anchor, $T_{i,j:}$ is the one-hot label of correct locations and L_{CE} is the cross entropy loss. It is worth noting that it uses an extra dimension to indicate the absence of lane, so the formulation is composed of $(w+1)$-dimensional rather than w-dimensional classifications.

In this method, the computational cost can be greatly decreased. Suppose the image size is $H \times W$. In general, the number of predefined row anchors and gridding size are far less than the size of an image, $h \ll H$ and $w \ll W$. The computational cost of formulation is $C \times h \times (w+1)$ while the one for segmentation is $H \times W \times (C+1)$. Therefore, this formulation could achieve extremely fast speed.

Category	R50-Seg	SCNN	FD-50	R34-SAD	SAD	Res18-Ours
Normal	87.4	90.6	85.9	89.9	90.1	87.5
Crowded	64.1	69.7	63.6	68.5	68.8	65.7
Night	60.6	66.1	57.8	64.6	66.0	61.3
No-line	38.1	43.4	40.6	42.2	41.6	39.6
Shadow	60.7	66.9	59.9	67.7	65.9	61.9
Arrow	79.0	84.1	79.4	83.8	84.0	79.8
Dazzlelight	54.1	58.5	57.0	59.9	60.2	58.0
Curve	59.8	64.4	65.2	66.0	65.7	57.8
Crossroad	2505	1990	7013	1960	1998	1856
Total	66.7	71.6	-	70.7	70.8	67.3
Runtime(ms)		133.5		50.5	13.4	2.9
Multiple		1.0x		2.6x	10.0x	46.0x
FPS		7.5		19.8	74.6	345.0

Table 1. **Comparison of F1-measure and runtime** – on CULane testing set with IoU threshold=0.5. For crossroad, only false positives are shown, the less, the better.

3.1.2 Lane Structural Loss

Apart from the classification loss, [32] offer two additional loss functions for modeling the location relations of lane points. In this way, it is conducive to learning structural information. At the same time, because of the position information in the horizontal row direction, this information can also be used to add the prior constraints of the lane line.

The first one is derived from the fact that lanes are continuous. Lane points in neighboring row anchors should be as near as possible to one another. Thus, the similarity loss function may be written as follows:

$$L_{sim} = \sum_{i=1}^{C} \sum_{j=1}^{h-1} \|P_{i,j,:} - P_{i,j+1,:}\|_1 \tag{2}$$

Another structural loss function focuses is derived from the shape of lanes. In general, most of the lanes are straight. Even the curving lane is largely straight owing to the perspective effect. Thus, the shape loss function may be written as follows:

$$L_{shp} = \sum_{i=1}^{C} \sum_{j=1}^{h-2} \|(L_{OC_{i,j}} - L_{OC_{i,j+1}}) - (L_{OC_{i,j+1}} - L_{OC_{i,j+2}})\|_1 \tag{3}$$

in which $L_{OC_{i,j}}$ is the location on the i-th lane, the j-th row anchor. Finally, the overall structural loss can be written as follows:

$$L_{str} = L_{sim} + \lambda L_{shp} \tag{4}$$

in which λ is the loss coefficient.

3.1.3 Feature Aggregation

In the above content, the focus is on the lane line area and the positioning constraints of the lane line, but there is a

CVPR
#0033797411

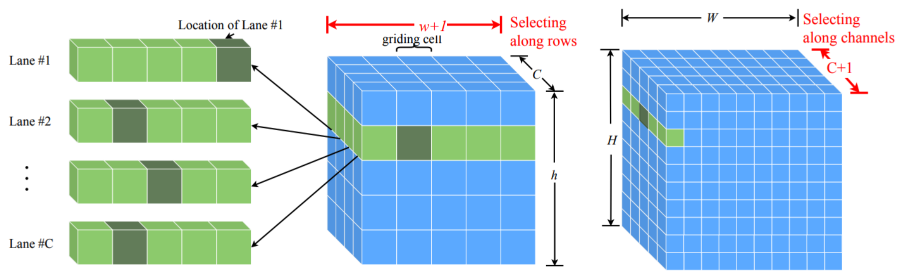

Figure 1. **Illustration of formulation (left) and segmentation (right)[32] –** The formulation selects locations on rows, whereas the segmentation classifies each pixel. The dimensions utilized for classification are also varied, as indicated by the red arrows.

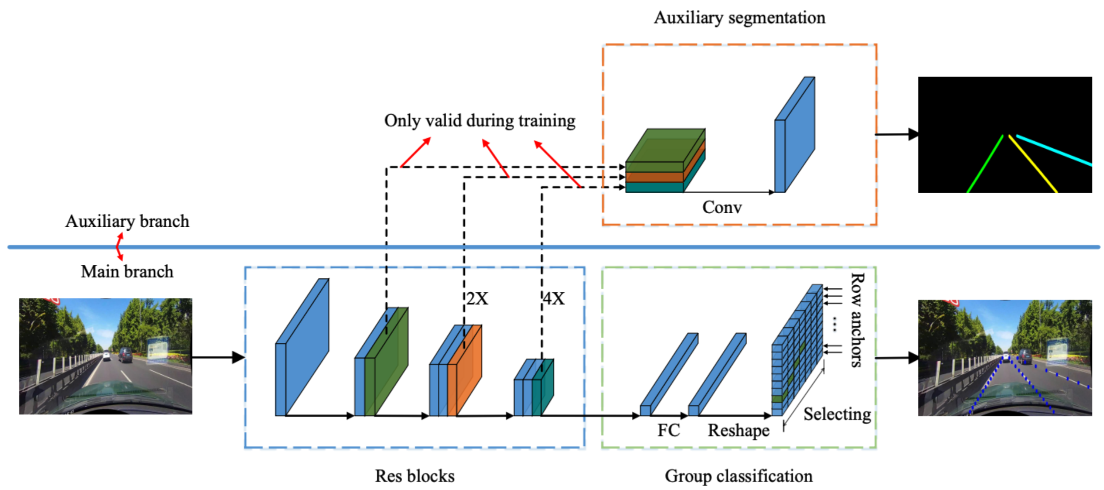

Figure 2. **Overall Network Architecture[32]**

lack of extraction of global and local information. Therefore, [32] also presents a method for aggregating auxiliary features that work with both global context and local features. To model local features, an auxiliary segmentation task based on multi-scale features is proposed. It is worth noting that this technique employs the auxiliary segmentation task only during the training phase and dismisses it during the testing phase. In this way, even if additional auxiliary segmentation tasks are added, the inference speed of the method will not be affected, which is the same as the network that does not use auxiliary segmentation tasks. Cross entropy is used as an auxiliary segmentation loss. Thus, the total loss associated with this method may be represented as:

$$L_{total} = L_{cls} + \alpha L_{str} + \beta L_{seg} \tag{5}$$

in which L_{seg} is the auxiliary segmentation loss, α and β are loss coefficients.

3.1.4 Network Architecture

The backbone of our architecture is ResNet-18. The overall architecture can be seen in Fig.2[32].

3.2. Dataset

We conduct experiments on one of the widely used benchmark datasets, CULane[29], to evaluate our method. CULane is a large scale challenging dataset for academic research on traffic lane detection. It is collected by cameras mounted on six different vehicles driven by different drivers. More than 55 hours of videos were collected and 133,235 frames were extracted. The number of lanes is equal to or smaller than 4, and the environment includes urban and highway. CULane has divided into 88880 for the training set, 9675 for the validation set, and 34680 for the testing set. The testing set is divided into normal and 8 challenging categories, which correspond to the 9 different scenarios.

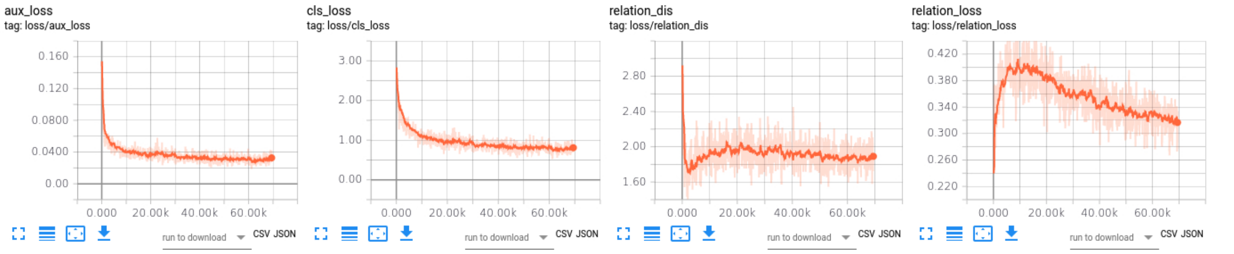

Figure 3. **Loss performance –** from left to right are L_{seg} (aux loss), L_{cls} (cls loss), L_{shp} (relation dis), and L_{sim} (relation loss)

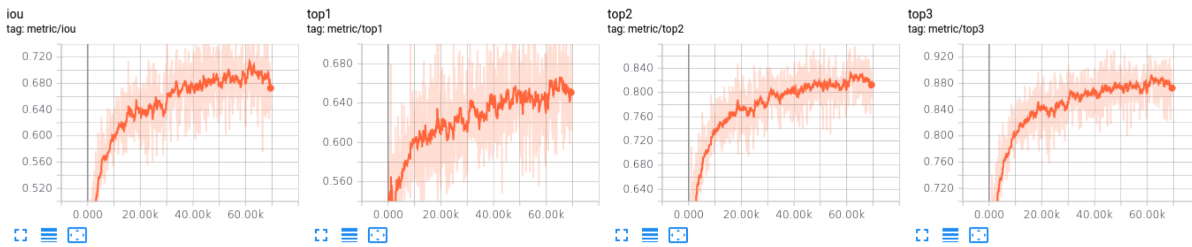

Figure 4. **IoU and accuracy**

3.3. Implementation Details

For CULane datasets, we use the row anchors that are defined by the dataset. The row anchors of CULane range from 260 to 530 and the number of gridding cells is set to 150. During the optimization procedure, images are resized to 288×800 following[29]. Adam optimizer is used to train our model. The learning rate of [32] is set to 0.1. [32] used multi-GPU training while I could only use a single GPU with a school server. Therefore, the learning rate is set to 2.5e-2 with weight decay 1e-4. Loss coefficients λ, α and β are all set to 1. The batch size is set to 32, and the total number of training epochs is set to 30 and 50 respectively for CULane dataset. All models are trained and tested with PyTorch and Scholar, a small computer cluster, Two Sky Lake CPUs @ 2.60GHz with one NVIDIA Tesla V100, at Purdue University.

3.4. Evaluation metrics

For CULane, each lane is treated as a 30-pixel-width line. The intersection-over-union (IoU) is computed between ground truth and predictions. F-measure is taken as the evaluation metric and formulated as follows:

$$\begin{aligned} F-measure &= \frac{2}{\frac{1}{R} \times \frac{1}{P}} \\ &= \frac{2 \times P \times R}{P+R} \\ &= \frac{TP}{TP + \frac{1}{2}(FP+FN)} \end{aligned} \tag{6}$$

where $P = Precision = \frac{TP}{TP+FP}$
$R = Recall = \frac{TP}{TP+FN}$
TP is true positives, FP is false positives, FN is false negatives.

4. Results

During the model training process, all loss, L_{cls}, L_{sim}, L_{shp}, L_{seg}, decrease and gradually converge. We can see Fig.3. for the visualization of loss in detail. Besides, intersection-over-union (IoU), ToP-1 accuracy, ToP-2 accuracy, and ToP-3 accuracy all gradually increase. We can see Fig.4. for the visualization of IoU and accuracy in detail. For the model testing result, we compare six different approaches, R50-seg[5] , SCNN[29], FD-50[30], R34-SAD[10], SAD, and our implementation, for F1-measure and runtime with IoU threshold=0.5. We find that our implementation has a good F1-measure with the fastest runtime, and the FPS is up to 345.0 with a resolution of 288×800 which is 46 times better than SCNN[29]. We can see the detail in Table.1. Finally, for the visualization of lane detection in the testing set, please see the Fig.5.

5. Conclusions

In this paper, we introduce the problems faced by the development of autonomous driving. The first one is most AI models and computational costs are complicated and large for autonomous driving. The second one is a model should

Figure 5. **Visualization of inference results –** including 8 different scenarios, which are crowd, hlight, arrow, curve, shadow, noline, curve, and night

have good robustness and generalization to face diverse scenarios. Therefore, we introduce three topics that try to solve the above problems from different papers. We also implement one of paper[32] and further discuss the method which considers the lane detection problem as a row-based selecting problem using a global feature and compares the experiment result with other methods. Finally, we achieve an extremely fast inference time at various challenging scenarios. We also think it's worth further discussing the issue of the combination of CURL framework and one reinforcement learning algorithm to realize the control and perception part for autonomous driving.

References

[1] M. Aly. Real time detection of lane markers in urban streets. In *2008 IEEE Intelligent Vehicles Symposium*, pages 7–12. IEEE, 2008. 1

[2] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458. PMLR, 2017. 3

[3] M. Bertozzi and A. Broggi. Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE transactions on image processing*, 7(1):62–81, 1998. 1

[4] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017. 3

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 6

[6] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. 2, 3

[7] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1259–1272, 2017. 3

[8] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d

object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2, 3

[9] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 3

[10] Y. Hou, Z. Ma, C. Liu, and C. C. Loy. Learning lightweight lane detection cnns by self attention distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1013–1021, 2019. 6

[11] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. In *arXiv preprint arXiv:1504.01716*, 2015. 1, 2

[12] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016. 3

[13] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019. 3

[14] K. P. Kielak. Do recent advancements in model-based deep reinforcement learning really improve data efficiency? 2019. 3

[15] J. Kim and M. Lee. Robust lane detection based on convolutional neural network and random sample consensus. In *International conference on neural information processing*, pages 454–461. Springer, 2014. 2

[16] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2, 3

[17] M. Laskin, A. Srinivas, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020. 1, 2, 3

[18] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019. 3

[19] S. Lee, J. Kim, J. Shin Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. Seok Hong, S.-H. Han, and I. So Kweon. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1947–1955, 2017. 2

[20] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016. 3

[21] P. Li, X. Chen, and S. Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7644–7652, 2019. 1, 2

[22] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 2, 3

[23] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 2

[24] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. 2

[25] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 2, 3

[26] J. K. Murthy, G. S. Krishna, F. Chhaya, and K. M. Krishna. Reconstructing vehicles from a single image: Shape priors for road scene understanding. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 724–731. IEEE, 2017. 3

[27] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 286–291. IEEE, 2018. 1

[28] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[29] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 4, 5, 6

[30] J. Philion. Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11582–11591, 2019. 6

[31] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2

[32] Z. Qin, H. Wang, and X. Li. Ultra fast structure-aware deep lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 276–291. Springer, 2020. 1, 4, 5, 6, 7

[33] E. Shelhamer, P. Mahmoudieh, M. Argus, and T. Darrell. Loss is its own reward: Self-supervision for reinforcement learning. *arXiv preprint arXiv:1612.07307*, 2016. 3

[34] T.-Y. Sun, S.-J. Tsai, and V. Chan. Hsi color model based lane-marking detection. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 1168–1172. IEEE, 2006. 2

[35] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990. 3

[36] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, et al.

Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 3

[37] H. P. van Hasselt, M. Hessel, and J. Aslanides. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32:14322–14333, 2019. 3

[38] Y. Wang, D. Shen, and E. K. Teoh. Lane detection using spline model. *Pattern Recognition Letters*, 21(8):677–689, 2000. 2

[39] Y. Wang, E. K. Teoh, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22(4):269–280, 2004. 1

[40] B. Xu and Z. Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2345–2353, 2018. 2

[41] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 3

[42] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019. 3

[43] B. Yu and A. K. Jain. Lane boundary detection using a multiresolution hough transform. In *Proceedings of International Conference on Image Processing*, volume 2, pages 748–751. IEEE, 1997. 2

[44] M. Zeeshan Zia, M. Stark, and K. Schindler. Are cars just 3d boxes?-jointly estimating the 3d shape of multiple objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3678–3685, 2014. 3

[45] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 2