Virtual to Real Reinforcement Learning for Autonomous Driving

Yurong You*
Shanghai Jiao Tong University
Shanghai, China 200240
yurongyou@sjtu.edu.cn

Xinlei Pan*
UC Berkeley
Berkeley, CA 94720
xinleipan@berkeley.edu

Ziyan Wang
Tsinghua University
Beijing, China 100084
zy-wang13@mails.tsinghua.edu.cn

Cewu Lu
Shanghai Jiao Tong University
Shanghai, China 200240
lu-cw@cs.sjtu.edu.cn

Abstract

Reinforcement learning is considered as a promising direction for driving policy learning. However, training autonomous driving vehicle with reinforcement learning in real environment involves non-affordable trial-and-error. It is more desirable to first train in a virtual environment and then transfer to the real environment. In this paper, we propose a novel realistic translation network to make model trained in virtual environment be workable in real world. The proposed network can convert non-realistic virtual image input into a realistic one with similar scene structure. Given realistic frames as input, driving policy trained by reinforcement learning can nicely adapt to real world driving. Experiments show that our proposed virtual to real (VR) reinforcement learning (RL) works pretty well. To our knowledge, this is the first successful case of driving policy trained by reinforcement learning that can adapt to ***real world*** *driving data.*

1. Introduction

Autonomous driving aims to make a vehicle sense its environment and navigate without human input. To achieve this goal, the most important task is to learn the driving policy that automatically outputs control signals for steering wheel, throttle, brake, etc, based on observed surroundings. The straight-forward idea is end-to-end supervised learning [3, 4], which trains a neural network model mapping visual input directly to action output, and the training data is labeled image-action pairs. However, supervised approach learns driving skills in a short-sighted manner, which means action learning is solely based on current observation. By contrast, human drivers attempt to predict what's happening in the near future and plan their actions. Recently, reinforcement learning has been considered as a promising technique to learn driving policy due to its expertise in action planing [24, 26, 16]. In other words, reinforcement learning is smarter than supervised approach, in that it acquires the driving skill by maximizing long-term reward instead of focusing on the short-term benefits.

However, reinforcement learning learns in a trial-and-error fashion and undesirable driving actions would happen at early stage. Training autonomous cars in real world will cause damages to vehicles and the surroundings. Therefore, most of research is still at the stage of simulations [18, 26, 16], which fails to meet the ultimate expectation of driving in real world, since the appearance of virtual environment is different from real world scene.

In this paper, we propose a *realistic translation network* to tackle this problem. Our proposed network (shown as Figure 1) converts virtual image rendered by simulator to a realistic one. Though virtual and realistic images have different characteristics, they share a common scene parsing representation (segmentation map of road, vehicle etc.). Therefore, our realistic translation network makes use of scene parsing representation as the interim to achieve translation from virtual to realistic image. This insight is similar to natural language translation, where semantic meaning is the interim between different languages. Specifically, our realistic translation network includes two modules. The first one is a virtue-to-parsing or virtual-to-segmentation module that pursues a scene parsing representation of input virtual image. The second one is a parsing-to-real network to translate scene parsing representations into realistic images. With realistic translation network, reinforcement learning learnt on the realistic driving video can nicely apply to real world driving.

*Indicates equal contribution

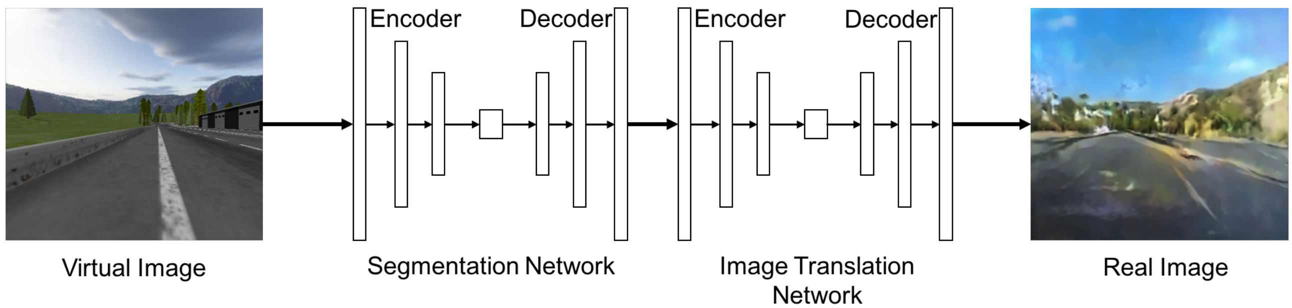

Figure 1. Network Structure for virtual to real (VR) image translation. The Virtual Image (left) rendered by simulator is first segmented by the segmentation network and then translated to Real Image (right) by the image translation network (VISRI).

To demonstrate the effectiveness of our method, we trained our reinforcement learning model by using the realistic translation network to filter virtual images to synthetic real images and feed these real images as state inputs. We further trained another two models for comparison. The first model is also a reinforcement learning model, however, with only virtual image input, and has never seen real images before. The second model is a simple supervised learning model that maps real images to their corresponding actions. Our experiments illustrate that reinforcement learning model trained with translated real images has significantly better performance than reinforcement learning model trained with only virtual input, and has similar or close performance compared with supervised learning model which requires lots of labeled data samples for training.

Our method has several contributions:

Novel Virtual to Real Image Translation Network. We propose a novel virtual to real image translation network that first segment images into their semantic labels and then translate segmented images to real images using network trained on segmentation-real image pairs.

Train Autonomous Vehicle with Real Data as Prior Information. Instead of solely relying on simulation data, we propose to use real image-segmentation pairs to train an image-to-image translation network that can translate segmented images into real images. Besides, the real data are not annotated with corresponding actions which means our method does not require real world interactions or annotations which are often expensive to acquire.

2. Related Work

Supervised Learning for Autonomous Driving. Supervised learning methods are obviously straightforward way to train autonomous vehicles. ALVINN [20] provides an early example of using neural network for autonomous driving. Their model is simple and direct, which maps image inputs to action predictions with a shallow network. Powered by deep learning especially convolutional neural network, NVIDIA [3] recently provides an attempt to leverage driving video data for simple lane following task. Another work by [4] learns a mapping between input images to a number of key perception indicators, which are closely related to the affordance of a driving state. However, the learned affordance must be associated with actions through hand-engineered rules. These supervised methods works relatively well in simple tasks such as lane-following and driving on highway. On the other hand, imitation learning can also be regarded as supervised learning approach [33], where the agent observes the demonstrations performed by some expert and learns to imitate the action of the expert. However, an intrinsic shortcoming of the imitation learning algorithm is that it can not generalize very well. There is also the covariant shift problem in imitation learning [21].

Reinforcement Learning for Autonomous Driving. Reinforcement learning has been applied to a wide variety of robotics related tasks, such as computer games [19], robot locomotion [13, 9], and autonomous driving [1, 26]. One of the challenges in practical real-world applications of reinforcement learning is the high-dimensionality of state space as well as the non-trivial large action range. Developing an optimal policy over such high-complexity space is time consuming. Recent work in deep reinforcement learning has made great progress in learning in a high dimensional space with the power of deep neural networks [14, 19, 25, 16, 18]. However, both deep Q-learning method [19] and policy gradient method [16] require the agent to interact with the environment to get reward and feedback. It is unrealistic to train autonomous vehicle with reinforcement learning in a real environment because of the potential traffic accidents risk.

Reinforcement Learning in the Wild. Performing reinforcement learning with a car driving simulator and transfer learned models to real environment could enable faster, lower-cost training, and it is much safer than training with a real car. However, real-world driving challenge usually spans a diverse range, and it is often significantly different from the training environment in a car driving simulator in terms of their visual appearance. Models trained purely on virtual data do not generalize well to real images [7, 28]. Recent progress of transfer and domain adaptation learning in robotics has provide examples of simulation-to-real reinforcement training [22, 11, 27]. These models either first train a model in virtual environment and then fine-tune in the real environment [22], or learn an alignment between virtual image and real image by finding representations that are shared between the two domains [28], or use randomized rendered virtual environments to train and then test in real environment [23, 27]. The work of [22] proposes to use progressive network to transfer network weights from model trained on virtual data to the real environment and then fine-tune the model in a real setting. The training time in real environment has been greatly reduced by first training in virtual environment. However, it is still necessary to train the agent in the real environment, thus it does not solve the critical problem of avoiding risky trial-and-error in real world. Methods that try to learn an alignment between virtual image and real image could fail to generalize to more complex scenarios, especially when it is hard to find a good alignment between virtual image and real image. As a more recent work, [23] proposed a new framework for training a reinforcement learning agent with only virtual environment. Their work proved the possibility of performing collision-free flight in real world with training in 3D CAD model simulator. However, as mentioned in the conclusion of their paper [23], the manual engineering work to design suitable training environments is nontrivial, and it is more reasonable to attain better results by combining simulated training with some real data, though it is unclear from their paper how to combine real data with simulated training.

Image Synthesis and Image Translation. Image translation aims to predict image in some specific modality, given an image from another modality. This can be treated as a generic method as it predicts pixel from pixel. Recently, the community has made significant progress in generative approaches, mostly based on generative adversarial networks [10]. To name a few, the work of [30] explored the use of VAE-GAN [15] in generating 3D voxel models, and the work of [29] proposed a cascade GAN to generate natural image by structure and style. More recently, the work of [12] developed a general and simple framework for image-to-image translation which can handle various pixel level generative tasks like semantic segmentation, colorization, rendering edge maps, etc.

Scene Parsing. One part of our network is the semantic image segmentation network. There are already many great work in the field of semantic image segmentation. Many of them are based on deep convolutional neural network or fully convolutional neural network [17]. In order to get better quality segmented images, the work of [2] cut down the downsampling layers to avoid resolution reduction, and the work of [32, 5] uses dilated convolution to improve performance. Bi-linear interpolation and deconvolutional methods are also very popular these days such as [2, 17, 5]. In this paper, we use the SegNet for image segmentation, the structure of the network is reveal in [2], which is composed of two main parts. The first part is an encoder, which consists of Convolutional, Batch Normalization, ReLU and max pooling layers. The second part is a decoder, which replace the pooling layers with upsampling layers.

In our work, we do not use a single image translation network nor a single image segmentation network directly as it is hard to get a large amount of virtual-real image pairs, especially given the infinite possibility of real world images. Instead, we train two networks both end-to-end and then connect them together as a virtual-real translator. The image segmentation network will be used to transform virtual world images into their segmentations, and the image translation network will be used to translate segmented images into real world images.

3. Reinforcement Learning on Realistic Frames

We aim to successfully apply a trained agent in virtual environment into real-world driving. One of major gaps is that what RL observed is frames rendered by simulator. These frames is different with real frame captured by camera in terms of appearance. Therefore, we proposed a *realistic translation network* to convert virtual frames to realistic ones. Inspired by the work of pix2pix network [12], our network includes two modules, namely virtual-to-parsing and parsing-to-realistic network. The first one maps virtual frame to scene parsing image. The second one translates scene parsing to realistic frame with similar scene structure as input virtual frame. These two modules achieve realistic frames and maintain the scene structure of input virtual frames. The architecture of realistic translation network is illustrated on Figure 1.

Finally, we train a self-driving agent using reinforcement learning method on realistic frames obtained by realistic translation network. The approach we adopted is developed by [18], where they use the asynchronous actor-critic reinforcement learning algorithm to train a self-driving vehicle in the car racing simulator TORCS [31]. In this section, we will first present proposed realistic translation network and then discuss how to train driving agent under a reinforcement learning framework.

3.1. Realistic Translation Network

As there is no direct connection between virtual world images such as the images we see in the TORCS environment, and the real world images such as the data in [3], a

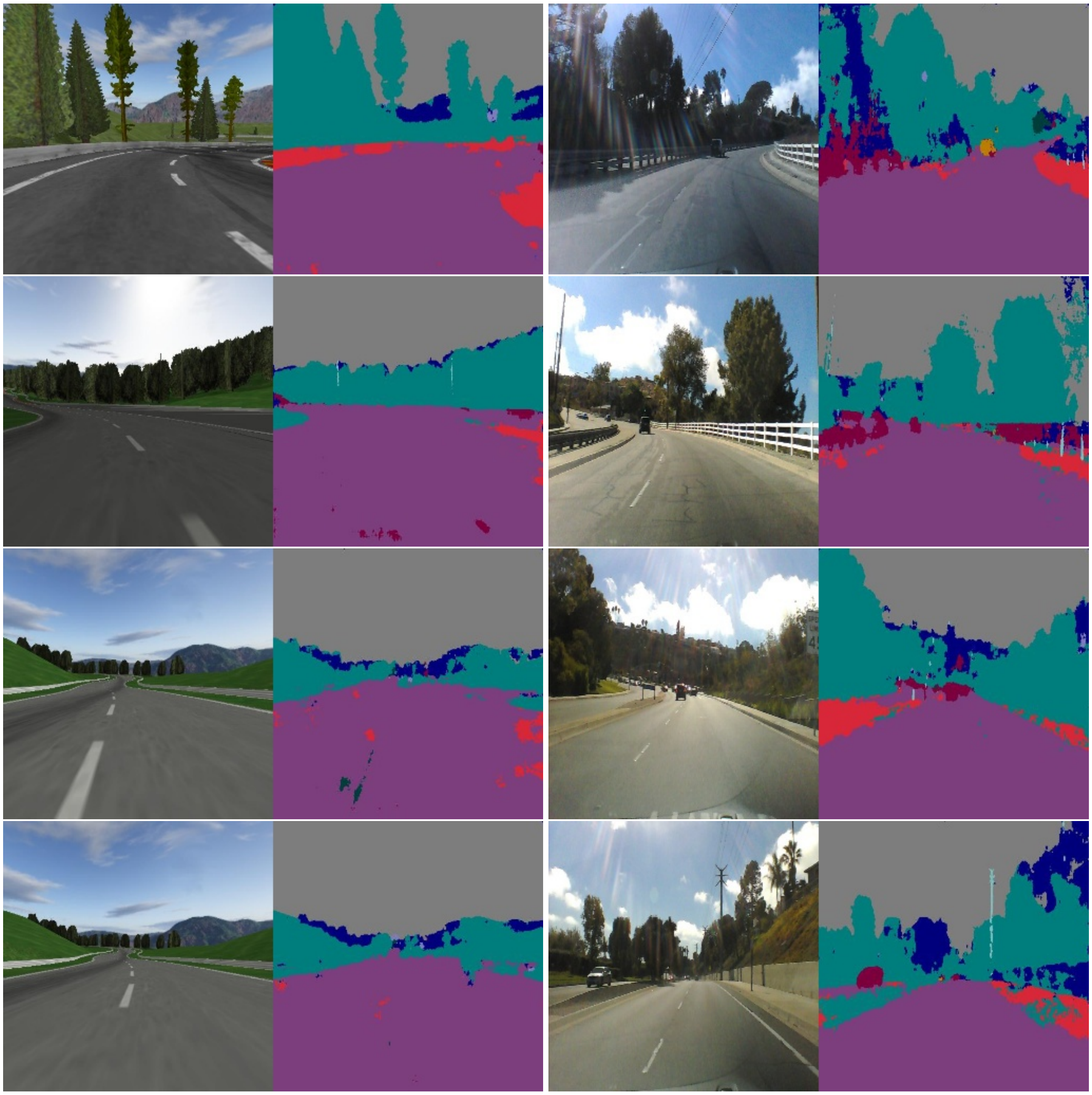

Figure 2. Example image segmentation for both virtual world images (Left 1 and Left 2) and real world images (Right 1 and Right 2).

direct mapping from virtual world image to real world image would be awkward. However, these two type of images both express driving scene. We can translate them by scene parsing representation. Therefore, our proposed realistic translation network attempts to build the connection by scene parsing on virtual image first and then translate scene parsing representation into realistic image.

3.2. Training Framework

We present how to train two main modules as follows.

Virtual-to-Parsing Network. This network aims to convert the virtual frames rendered by TORCS into scene parsing result. We use the network of [2] to parse the image into different semantic regions. The network structure of SegNet is a deep fully convolutional neural network architecture that follows a encoder-decoder fashion. To use the SegNet to segment images in virtual world, the network will be trained on CityScape driving scene segmentation dataset [8]. Then the trained model is used to parse virtual scenes in TORCS [31].

Parsing-to-Realistic Network. To produce realistic images, we learn how to translate the scene parsing result into realistic images with a unchanged scene structure. To this end, architecture of [12] is adopted. We train the model on the dataset of [6] with scene segmentation and natural image pairs. The scene segmentations are also obtained using [2].

3.3. Reinforcement Learning for Training a Self-Driving Vehicle

We use a conventional RL solver Asynchronous Advantage Actor-Critic (A3C)[18] to train the self driving vehicle, which has performed well on various machine learning tasks. A3C algorithm is a fundamental Actor-Critic algorithm that combines several classic reinforcement learning algorithms with the idea of asynchronous parallel threads. Multiple threads run at the same time with unrelated copies of the environment, generating their own sequences of training samples. Those actors-learners proceed as though they are exploring different parts of the unknown space. For one thread, parameters are synchronized before an iteration of learning and updated after finishing it. When the algorithm is applied to our experiment, we define a reward function proportional to the agent's velocity along the center of the track at the agent's current position [18].

4. Experiments

4.1. Experiments Setting

To demonstrate the performance of our method, we trained three models. The first one is our proposed reinforcement learning model with realistic translation network. The second one is a reinforcement learning model with virtual input as state representation. The third one is a supervised learning end-to-end model trained on real data with action labels. We further evaluate these models on a held out real driving data with action labels.

4.2. Dataset

The virtual image data are collected in the aalborg environment in TORCS [31], and a total 1673 images are collected which covers the entire driving cycle of aalborg environment. The real driving video data with action labels are from [6], which is collected in a sunny day on highway with detailed steering angle annotations per frame.

4.3. Training Details

Scene Segmentation. We adopt the image semantic segmentation network design of [2] and their trained segmentation network on the CityScape image segmentation dataset [8] to segment both virtual images rendered by TORCS and real images from [6]. The network was trained on the CityScape dataset with 11 classes and was trained with 30000 iterations. We used the Aalborg environment in TORCS [31], and collected 1673 images from this environment as well as their segmentations. We further segmented all 45569 images in [6].

Image Translation Network Training. We trained both virtual-to-parsing and parsing-to-real network using the segmented virtual-segmentation image pairs and segmentation-real image pairs. The translation network are of a encoder-decoder fashion as shown in figure 1. In the image translation network, we used U-Net architecture with skip connection to connect two separate layers from encoder and decoder respectively, which have the same output feature map shape. The input size of the generator is 256×256. Each convolutional layer has a kernel size of 4×4 and striding size of 2. LeakyReLU is applied after every convolutional layer with a slope of 0.2 and ReLU is applied after every deconvolutional layer. In addition, batch normalization layer is applied after every convolutional and deconvolutional layer. The final output of the encoder is connected with a convolutional layer which yields output of shape $3 \times 256 \times 256$ followed by Tanh. We used all 1673 virtual-segmentation image pairs to train a virtual to segmentation network. As there are redundancies in the 45k real images, we select 1762 images from the 45k images to train a parsing-to-real image translation network. To train the image translation models, we used the Adam optimizer with an initial learning rate of 0.0002, momentum of 0.5, batchsize of 16, and 200 iterations until convergence.

Reinforcement Training. The network structure used in our training is similar to that of [18] where the actor network is a 4-layer convolutional network with ReLU activation functions in-between. The network takes in 4 consecutive RGB frames as state input and output 9 discrete actions which corresponds to "go straight with acceleration", "go left with acceleration", "go right with acceleration", "go straight and brake", "go left and brake", "go right and brake", "go straight", "go left", and "go right". We trained the reinforcement learning agent with 12 asynchronous threads, and with the RMSProp optimizer at an initial learning rate of 0.01, $\gamma = 0.9$, and $\epsilon = 0.1$.

Baseline Reinforcement Learning. To make a fair comparison between our method and reinforcement learning with only virtual input, we trained the vehicle in the virtual car racing simulator TORCS [31] with virtual image as input using the same reinforcement learning framework as in our method. We later show that performance improves to some extent by training with translated realistic images.

Supervised Learning Method. We further trained a neural network mapping real images to learn driving actions under supervised learning manner. The network architecture is the same as the the policy network in our proposed method. The input of the network is a sequence of four consecutive images, the output of the network is the action probability vector, and the elements in the vector represent the probability of going straight, turning left and turning right.

Evaluation. The original dataset [6] provides the steer-

Figure 3. Examples of Virtual to Real Image Translation. Odd columns are virtual images captured from TORCS. Even columns are synthetic real world images corresponding to virtual images on the left.

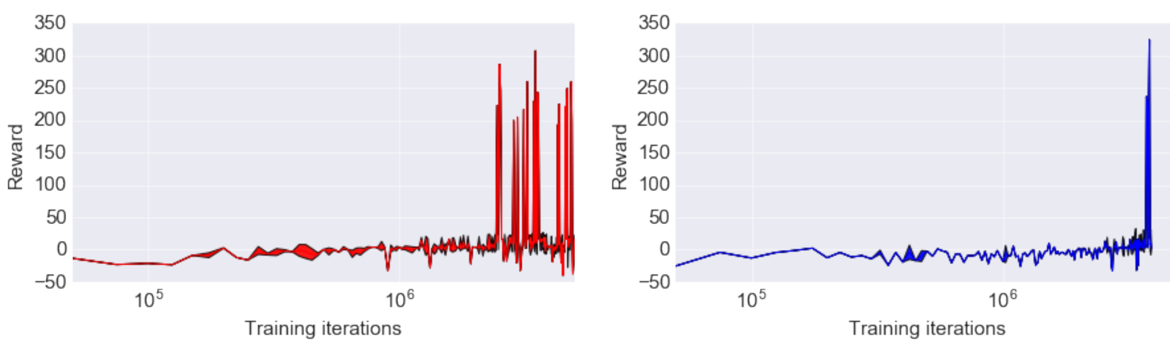

Figure 4. Training Self-Driving Vehicle in Virtual Environment (left) and Interacting with Synthesized Real World (right)

ing angle annotations per frame. However, the actions performed in the TORCS virtual environment only contain "going left", "going right", and "going straight" or their combinations with "acceleration" or "brake". Therefore, we define a label mapping strategy to translate steering angle labels to action labels in the virtual simulator. We relate steering angle in $(-10, 10)$ to action "going straight" (since small steering angle is not able to result in a distinct turning in a short time), steering angle less than -10 to action "going left" and steering angle more than 10 to action "going right".

5. Results

5.1. Image Segmentation Results

We used image segmentation model trained on the cityscape [8] dataset to segment both virtual and real images. Examples are shown in figure 2. As shown in the figure, although the original virtual image and real image look quite different, their scene parsing results are very similar. Therefore, it is reasonable to use scene parsing as the interim to connect virtual image and real image.

5.2. Qualitative Result of Realistic Translation Network

Figure 3 shows some representative results of our image translation network. The odd columns are virtual image in TORCS, and the even columns are translated images in the real world. The images in the virtual environment appears to be darker than the translated images, as the real images used to train the translation network is captured in a sunny day. Therefore, our model succeed to synthesize realistic images with similar appearance with the original ground truth real images.

5.3. Reinforcement Training Results

We first trained the self driving vehicle with our model and got the reward per iteration curve shown in figure 4. We further trained the self driving vehicle in the virtual environment with the baseline method and got the reward per iteration curve shown in figure 4. The two curves show that by interacting with translated realistic images, our model can achieve similar reward level as model trained solely in virtual environment.

We further provide the evaluation results of our proposed method (Ours), the baseline method (BS), and the supervised learning method (SV). The results contain action prediction accuracy shown in table 1. Results show that our proposed method has a better overall performance than the baseline method, where the reinforcement training agent is trained in a virtual environment without seeing any real data. The supervised method has the best overall performance, however, was trained with supervised labeled data.

Table 1. Action prediction accuracy for the three methods.

Accuracy	Ours	BS	SV
Dataset in [6]	43.40%	28.33%	53.60%

6. Conclusion

We proved that by using synthetic real images as training data in reinforcement learning, the agent generalize better in a real environment than pure training with virtual data. The next step would be to design a better image-to-image translation network and a better reinforcement learning framework to surpass the performance of supervised learning.

Thanks to the bridge of scene parsing, virtual images can be translated into realistic images which maintains its scene structure. The learnt RL model on realistic frames can be easily applied to real-world environment. We also notice that the translation results of a segmentation map is not unique. For example, segmentation map indicates a car, but it does not assign which color of that car should be. Therefore, one of our future work is to make parsing-to-realistic network output various possible appearances (e.g. color, texture). In this way, bias in RL training would be largely reduced.

We provide the first example of training a self-driving vehicle using reinforcement learning algorithm by interacting with a synthesized real environment with our proposed image-to-segmentation -to-image framework. We show that by training in this environment, it is possible to train a self driving vehicle that can be placed in the real world.

References

[1] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19:1, 2007.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[3] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[4] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. *CoRR*, abs/1505.00256, 2015.

[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. *arXiv:1606.00915*, 2016.

[6] S. Chen. Autopilot-tensorflow, 2016.

[7] P. Christiano, Z. Shah, I. Mordatch, J. Schneider, T. Blackwell, J. Tobin, P. Abbeel, and W. Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[9] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng. Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot. *The International Journal of Robotics Research*, 27(2):213–228, 2008.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[11] A. Gupta, C. Devin, Y. Liu, P. Abbeel, and S. Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv preprint arXiv:1703.02949*, 2017.

[12] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[13] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2619–2624. IEEE, 2004.

[14] J. Koutník, G. Cuccu, J. Schmidhuber, and F. Gomez. Evolving large-scale neural networks for vision-based reinforcement learning. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1061–1068. ACM, 2013.

[15] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *CoRR*, abs/1512.09300, 2015.

[16] 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.

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[18] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783, 2016.

[19] 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.

[20] D. A. Pomerleau. Alvinn, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, Computer Science Department, 1989.

[21] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *AISTATS*, volume 3, pages 3–5, 2010.

[22] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell. Sim-to-real robot learning from pixels with progressive nets. *CoRR*, abs/1610.04286, 2016.

[23] F. Sadeghi and S. Levine. (cad)^{2}rl: Real single-image flight without a single real image. *CoRR*, abs/1611.04201, 2016.

[24] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani. End-to-end deep reinforcement learning for lane keeping assist. *arXiv preprint arXiv:1612.04340*, 2016.

[25] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In *ICML*, pages 1889–1897, 2015.

[26] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *CoRR*, abs/1610.03295, 2016.

[27] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *arXiv preprint arXiv:1703.06907*, 2017.

[28] E. Tzeng, C. Devin, J. Hoffman, C. Finn, P. Abbeel, S. Levine, K. Saenko, and T. Darrell. Adapting deep visuomotor representations with weak pairwise constraints. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.

[29] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016.

[30] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[31] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner. Torcs, the open racing car simulator. *Software available at http://torcs. sourceforge. net*, 2000.

[32] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[33] J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.