

DeepRoad: GAN-based Metamorphic Autonomous Driving System Testing

Mengshi Zhang¹, Yuqun Zhang^{2*}, Lingming Zhang³, Cong Liu³, Sarfraz Khurshid¹

¹ University of Texas at Austin

² Southern University of Science and Technology

³ University of Texas at Dallas

mengshi.zhang@utexas.edu, zhangyq@sustc.edu.cn, lingming.zhang@utdallas.edu,
cong@utdallas.edu, khurshid@ece.utexas.edu

Abstract

While Deep Neural Networks (DNNs) have established the fundamentals of DNN-based autonomous driving systems, they may exhibit erroneous behaviors and cause fatal accidents. To resolve the safety issues of autonomous driving systems, a recent set of testing techniques have been designed to automatically generate test cases, e.g., new input images transformed from the original ones. Unfortunately, many such generated input images often render inferior authenticity, lacking accurate semantic information of the driving scenes and hence compromising the resulting efficacy and reliability.

In this paper, we propose DeepRoad, an unsupervised framework to automatically generate large amounts of accurate driving scenes to test the consistency of DNN-based autonomous driving systems across different scenes. In particular, DeepRoad delivers driving scenes with various weather conditions (including those with rather extreme conditions) by applying the Generative Adversarial Networks (GANs) along with the corresponding real-world weather scenes. Moreover, we have implemented DeepRoad to test three well-recognized DNN-based autonomous driving systems. Experimental results demonstrate that DeepRoad can detect thousands of behavioral inconsistencies for these systems.

1 Introduction

*“The train came out of the long tunnel into the snow country.
The earth lay white under the night sky. The train pulled up
at a signal stop.”*

The above quotation is from the first paragraph of fiction “Snow Country”, which describes the scene when the protagonist Shimamura enters the snow country. Back to that time, train was the major vehicle for long-distance travels, while people have more choices today. Now, suppose Shimamura takes a Tesla in Autopilot mode [5], after coming out of the

tunnel, there raises a question: can the autopilot system operate safely on the snow-covered road, or the story just ends with a tragedy?

Autonomous driving is expected to transform the auto industry. Typically, autonomous driving refers to utilizing sensors (cameras, LiDAR, RADAR, GPS, etc) to automatically control vehicles without human intervention. The recent advances in Deep Neural Networks (DNNs) enables autonomous driving systems to adapt their driving behaviors according to the dynamic environments. In particular, an end-to-end supervised learning framework is made possible to train a DNN for predicting driving behaviors (e.g., steering angles) by inputting driving scenes (e.g., images), using the ⟨driving scene, driving behavior⟩ pairs as the training data. For instance, DAVE-2 [13], released by NVIDIA in 2016, can predict steering angles based on only driving scenes captured by a single front-centered camera of autonomous cars.

Recent testing techniques [23; 28] demonstrate that adding error-inducing inputs to the training datasets can help improve the reliability and accuracy of existing autonomous driving models. For example, the most recent DeepTest work [28] designs systematic ways to automatically generate test cases, seeking to mimic real-world driving scenes. Its main methodology is to transform training driving scenes by applying simple affine transformations and various effect filters such as blurring/fog/rain to the original image data, and then check if autonomous driving systems perform consistently among the original and transformed scenes. With large amounts of original and transformed driving scenes, DeepTest can detect various erroneous inconsistent driving behaviors for some well-performed open-source autonomous driving models, in a cheap and quick manner.

However, it is observed that the methodologies applied in DeepTest to generate test cases cannot accurately reflect the real-world driving scenes. Specifically, real-world driving scenes can rarely be affine-transformed and captured by the cameras of autonomous driving systems; the blurring/fog/rain effects made by simply simulating the corresponding effects also appear to be unrealistic which compromises the efficacy and reliability of DeepTest. For instance, Figure 1a shows the fog effect transformation applied in DeepTest. It can be observed that Figure 1a is distorted. In particular, it appears to be synthesized by simply dimming the original image and mixing it with the scrambled “smoke” effect. In

*Corresponding author.

addition, Figure 1b shows the rain effect transformation applied in DeepTest. Similarly, DeepTest simply simulates rain by adding a group of lines over the original image. This rain effect transformation is even more distorted because usually when it rains, the camera tends to be wet and the image is highly possible to be blurred. The fact that few test cases in DeepTest appear authentic to reflect the real-world driving scenes makes it difficult to determine whether the erroneous driving behaviors are caused by the flaws of the DNN-based models or the inadequacy of the testing technique itself. Furthermore, there are many potential driving scenes that cannot be easily simulated with simple image processing. For instance, the snowy road condition requires different sophisticated transformations for the road and the roadside objects (such as trees).

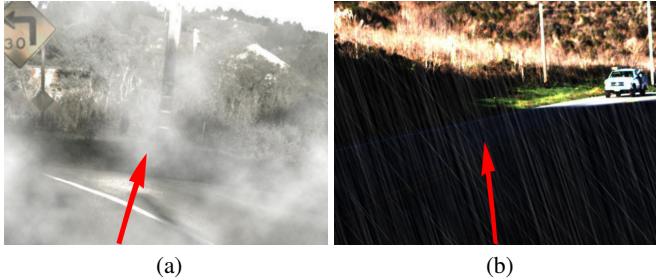


Figure 1: Foggy and rainy scenes via DeepTest

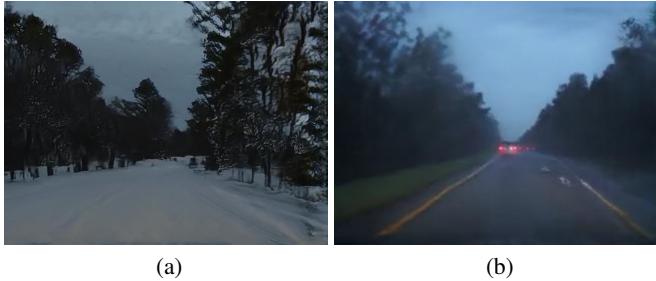


Figure 2: Snowy and rainy scenes via DeepRoad

In order to automatically synthesize large amounts of authentic driving scenes for testing DNN-based autonomous driving systems, in this paper, we propose an unsupervised framework, namely DeepRoad, that employs a Generative Adversarial Network (GAN)-based technique [15] to deliver authentic driving scenes with various weather conditions which are rather difficult to be collected manually. Specifically, DeepRoad develops a metamorphic testing module for DNN-based autonomous systems, where the metamorphic relations are defined such that no matter how the driving scenes are synthesized to cope with different weather conditions, the driving behaviors are expected to be consistent with those under the corresponding original driving scenes. At this point, DeepRoad enables us to test the accuracy and reliability of existing DNN-based autonomous driving systems under different extreme weather scenarios, including heavy snow and hard rain, and can greatly complement the existing autonomous driving system testing approaches (such as DeepTest). For instance, Figure 2 presents

the snowy and rainy scenes generated by DeepRoad (from fine scenes), which can hardly be distinguished from genuine ones and cannot be generated using simple transformations.

Although our DeepRoad approach is general, and can be used to simulate various weather conditions, in this work, we first synthesize driving scenes under heavy snow and hard rain. In particular, based on the GAN technique, we collect images with the two extreme weather conditions from Youtube videos to transform real-world driving scenes and deliver them with the corresponding weather conditions. Subsequently, these synthesized driving scenes are used to test three well-recognized Udacity DNN-based autonomous driving systems [8]. The experimental results reveal that DeepRoad can effectively detect thousands of behavioral inconsistencies of different levels for these systems, indicating a promising future for testing autonomous driving systems via GAN-based road scene transformation.

The **contributions** of this paper are as follows.

- **Idea.** We propose the first GAN-based metamorphic testing approach, namely DeepRoad, to generate authentic driving scenes with various weather conditions for detecting autonomous driving system inconsistencies.
- **Implementation.** We implement the proposed approach based on Pytorch and Python to synthesize driving scenes under heavy snow and hard rain based on training data collected from Youtube videos.
- **Evaluation.** We use DeepRoad to test well-recognized DNN-based autonomous driving models and successfully detect thousands of inconsistent driving behaviors for them.

The rest of the paper is organized as follows. Section 2 introduces the background of autonomous driving systems and their existing testing techniques. Section 3 illustrates the overall approach of DeepRoad. Section 4 presents our experimental results on DeepRoad. Section 5 discusses some related work. Finally, Section 6 concludes this paper.

2 Background

Nowadays, DNN-based autonomous driving systems have been rapidly evolving [24; 13]. For example, many major car manufacturers (including Tesla, GM, Volvo, Ford, BMW, Honda, and Daimler) and IT companies (including Waymo/Google, Uber, and Baidu) are working on building and testing various DNN-based autonomous driving systems. In DNN-based autonomous driving systems, the neural network models take the driving scenes captured by the sensors (LiDar, Radar, cameras, etc.) as input and output the driving behaviors (e.g., steering and braking control decisions). In this work, we mainly focus on DNN-based autonomous driving systems with camera inputs and steering angle outputs. To date, feed-forward Convolutional Neural Network (CNN) [17] and Recurrent Neural Network (RNN) [25] are the most widely used DNNs for autonomous driving systems. Figure 3 shows an example CNN-based autonomous driving system. Shown in the figure, the system consists of an input (the camera image inputs) and an output layer (the steering

angle), as well as multiple hidden layers. The use of convolution hidden layers allows weight sharing across multiple connections and can greatly save the training efforts; furthermore, its local-to-global recognition process actually coincides with the manual object recognition process.

DNN-based autonomous driving systems are essentially software systems, which are error-prone and can lead to tragedies. For example, a Tesla Model S plowed into a fire truck at 65 mph while using Autopilot system [6]. To ensure the quality of software systems, many software testing techniques have been proposed in the literature [12; 21], where typically, a set of specific test cases are generated to test if the software programs perform as expected. The process of determining whether the software performs as expected upon the given test inputs is known as the *test oracle* problem [12]. Despite the abundance of traditional software testing techniques, they cannot be directly applied for DNN-based systems since the logics of DNN-based softwares are learned from data with minimal human interference (like a blackbox) while the logics of traditional software programs are manually created.

Recently, researchers have proposed various techniques to test DNN-based autonomous driving systems, e.g., DeepXplore [23] and DeepTest [28]. DeepXplore aims to automatically generate input images that can differentiate the behaviors of different DNN-based systems. However, it cannot be directly used to test one DNN-based autonomous driving system in isolation. The more recent DeepTest work utilizes some simple affine transformations and blurring/fog/rain effect filters to synthesize test cases to detect the inconsistent driving behaviors derived from the original and synthesized images. Although DeepTest can be applied to test any DNN-based driving system, the synthesized images may be unrealistic, and it cannot simulate complex weather conditions (e.g., snowy scenes).

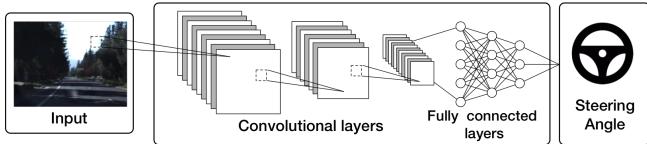


Figure 3: Autonomous driving system on CNN

3 Approach

3.1 Metamorphic DNN Testing

Metamorphic Testing [26] (MT) has been widely used to automatically generate tests to detect software bugs. The strength of MT lies in its capability to automatically solve the test oracle problem via Metamorphic Relations (MR). Formally, let p be a program mathematical representation mapping program inputs into program outputs (e.g., $p[[i]] = o$). Also, assuming f_I and f_O are two functions for transforming the input and output domain, respectively. Then, a MR can be formed as:

$$\forall i. p[[f_I(i)]] = f_O(p[[i]]) \quad (1)$$

With such MR, we can test an actual implementation \hat{p} of p by checking whether $\hat{p}[[f_I(i)]] = f_O(\hat{p}[[i]])$ for various inputs

i . The idea of testing a program implementation via cross-checking inputs and outputs with MR is called MT. For instance, given a program implementing the \sin function, we can use MT to create various new tests without worrying about the test oracle problem. For any existing input i for testing \sin , there are various facts that can directly serve as MR, e.g., $\sin(-i) = -\sin(i)$ and $\sin(i + 2\pi) = \sin(i)$. Note that $f_I(i) = f_O(i) = -i$ for the first example MR, while $f_I(i) = i + 2\pi \wedge f_O(i) = i$ for the second. With such MRs, we can transform the existing test inputs according to f_I to generate additional tests, and check the output based on f_O .

In this work, we further apply MT to test DNN-based autonomous driving systems. Formally, let DNN be a DNN-based autonomous driving system that continuously maps each image into predicted steering angle signal (e.g., turn left for 15°). Then, given the original image stream \mathbb{I} , we can define various image transformations \mathbb{T} that simply change the road scene (detailed shown in Section 3.2) and do not impact the prediction results for each image $i \in \mathbb{I}$ (e.g., the predicted direction should be the approximately the same for the same road condition during fine and rainy days). In this way, we have the following MR to test DNN with additional transformed inputs:

$$\forall i \in \mathbb{I} \wedge \forall \tau \in \mathbb{T}. DNN[\tau(i)] = DNN[i] \quad (2)$$

3.2 DNN-based Road Scene Transformation

The recent work DeepTest [28] also applied MT to test DNN-based autopilot systems. However, it only performs simple synthetic image transformation, such as adding simple blurring/fog/rain effect filters, and thus has the following limitations: (1) DeepTest may generate unrealistic images (e.g., the rainy scene shown in Figure 1b), (2) DeepTest cannot simulate complex road scene transformations (e.g., snowy scenes).

To complement DeepTest and generate various real-world road scenes fully automatically, in this work, we leverage UNIT [19], a recent published DNN-based method to perform unsupervised image-to-image transformation. One insight of UNIT is a paired images in different domains can be projected into a shared-latent space and have the same latent representation. In this way, given a new image from one domain (e.g., the original driving scene), UNIT can automatically generate its corresponding version in the other domain (e.g., rainy driving scene). Overall, UNIT is composed by generative adversarial networks (GANs) [15] and variational autoencoders (VAEs) [16].

Figure 4 presents the basic structure of UNIT, S_1 and S_2 denote two different domains (e.g., images include fine and rainy scenes, respectively), E_1 and E_2 are two autoencoders which can project the images from S_1 and S_2 to the shared-latent space Z . Suppose x_1 and x_2 are corresponding images which share the same contents, ideally, E_1 and E_2 would encode them to the same latent vector z . G_1 and G_2 are two domain specific generators which can translate a latent vector back to S_1 and S_2 , respectively. D_1 and D_2 are two discriminators which can detect whether the image belongs S_1 and S_2 , respectively. Ideally, the discriminators cannot differentiate whether the input image is from the target domain or a well-trained generator. Based on the autoencoders and generators, UNIT can be used to transform images between two

domains. For instance, image x_1 can be transformed to S_2 by $G_2(E_1(x_1))$.

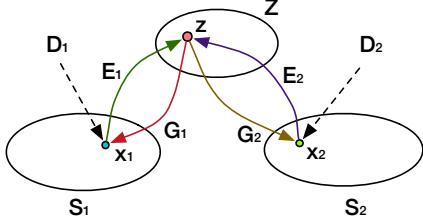


Figure 4: Structure of UNIT

The learning objective of UNIT can be decomposed to optimize three costs:

- **VAE loss** minimizes the loss of the image reconstruction for each $\langle E_i, G_i \rangle$ pair.
- **GAN loss** achieves the equilibrium point in the minimax game for each $\langle G_i, D_i \rangle$, where D_i aims to discriminate between images from the domain distribution and candidates produced by G_i aiming to fool D_i .
- **Cycle-consistency loss** minimizes the loss of cycle-reconstruction for each $\langle E_i, G_j, E_j, G_i \rangle$, ideally, $x_1 = G_1(E_2(G_2(E_1(x_1))))$ and $x_2 = G_2(E_1(G_1(E_2(x_2))))$

The total loss can be summarized as follows:

$$\begin{aligned} \min_{E_1, E_2, G_1, G_2} \max_{D_1, D_2} & \mathcal{L}_{CC_1}(E_1, G_2, E_2, G_1) \\ & + \mathcal{L}_{CC_2}(E_2, G_1, E_1, G_2) \\ & + \mathcal{L}_{VAE_1}(E_1, G_1) + \mathcal{L}_{VAE_2}(E_2, G_2) \\ & + \mathcal{L}_{GAN_1}(D_1, G_1) + \mathcal{L}_{GAN_2}(D_2, G_2) \end{aligned}$$

3.3 Overall Framework

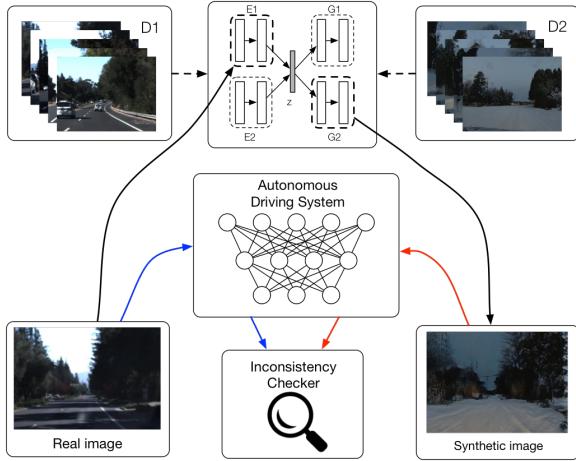


Figure 5: Framework of DeepRoad

Figure 5 shows the overall design of our metamorphic testing framework for DNN-based autonomous driving systems, DeepRoad. Shown in the figure, DeepRoad firstly takes unpaired training images from two target domains (e.g., one fine driving scene dataset and one rainy driving scene dataset), and utilizes the unsupervised UNIT to map the two

Table 1: Details of image sets

Dataset	Frame	Duration	Weather Cond.
Udacity Ep1	15212	N.A.	Sunshine
Udacity Ep2	5614	N.A.	Sunshine
Youtube Ep1	1000	28:55	Heavy snow
Youtube Ep2	1000	1:09:03	Hard rain

scene domains to the same latent space using the loss functions presented in Section 3.2. In this work, we sample images from the real-world Udacity Challenge 2 dataset [7] (fine scenes) and Youtube video (snowy or rainy scenes [11; 10]) and feed them into UNIT for training. After it is well-trained, DeepRoad uses the UNIT model to transform the whole real-world Udacity driving dataset to another scene (e.g., snowy or rainy scenes). That is, given any original fine driving scene i , DeepRoad can apply the trained UNIT model to derive its corresponding version in another weather condition (e.g., rainy scene), $\tau(i)$. Then, DeepRoad will feed each pair of real and synthesized images to the autonomous driving systems under test (i.e., DNN), and compare their prediction results (i.e., $DNN[\tau(i)]? = DNN[\tau]$) to detect any inconsistent behaviors. Since the road scenes should not largely impact the steering angles, any inconsistency may indicate correctness or robustness issues of the systems under test. Note that although in this work we only explore the rainy and snowy scene transformations, our DeepRoad approach is general, can support any scene transformation supported by the underlying UNIT model.

4 Experiments

4.1 Data

We use a real-world dataset released by Udacity [9] as a baseline to check the inconsistency of autonomous driving systems. From the dataset, we select two episodes of highway driving video where obvious changes of lighting and road conditions can be observed among frames. To train our UNIT model, we also collect images of extreme scenarios from Youtube. In the experiments, we select snow and hard rain, two extreme weather conditions to transform real-world driving images. To make the variance of collected images relatively large, we only search for videos which is longer than 20mins. In the scenario of hard rain, the video would record wipers swiping windows, and in the data preprocessing phase, we manually check and filter out those images. Note that all images used in the experiments are cropped and resized to 240×320 , and we have performed down-sampling for Youtube videos to skip consecutive frames with close contents. The detailed information is present in Table 1.

4.2 Models

We evaluate our framework on three DNN-based autonomous driving models which are released by Udacity [9]: Autumn [2], Chauffeur [3], and Wrightman [4]. We choose these three models as their pre-trained model are public and can be evaluated directly on the synthesized datasets. To be specific, the model details of Wrightman are not publicly released, however, just like black-box testing, our approach aims to detect the inconsistencies of the model instead

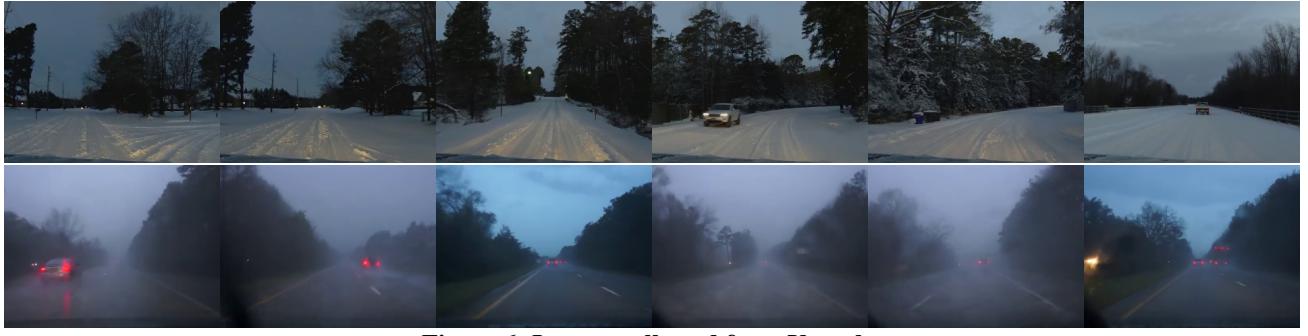


Figure 6: Images collected from Youtube



Figure 7: Real and GAN-generated images.

of localizing software faults, hence, we still use `Rwightman` for the evaluation.

Autumn. Autumn is composed by a data preprocessing module and a CNN. Specifically, Autumn first computes the optical flow of input images and input them to a CNN to predict the steering angles. The architecture of Autumn is: three 5x5 conv layers with stride 2 pluses two 3x3 conv layers and followed by five fully-connected layers with dropout. The model is implemented by OpenCV, Tensorflow and Keras.

Chauffeur. Chauffeur consists of one CNN and one RNN module with LSTM. The work flow is that CNN firstly extracts the features of input images and then utilizes RNN to predict the steering angle from previous 100 consecutive images. This model is also implemented by Tensorflow and Keras.

4.3 Metric

Based on our assumption, an autonomous driving system is consistent if its steering angle prediction does not change after modifying the weather condition of driving images. However, this assumption is too strong to be practical since minor steering angle change incurred by the scene change may still fall into the safe zone. Hence, similar with prior work [28], we relax the assumption and accept the prediction if the difference between the predicted steering angles of original and transformed images can be within an error bound. We define the number of inconsistent behaviors of autonomous driving systems as follows:

$$IB(DNN, \mathbb{I}) = \sum_{i \in \mathbb{I}} f(|DNN[i] - DNN[\tau(i)]| > \epsilon)$$

, where DNN denotes the autonomous driving model and \mathbb{I} is the real-world driving dataset. i denotes the i th image in \mathbb{I} .

τ denotes the image generator/transformer which can change the weather condition of the input image. Function f outputs 1 or 0 if and only if the input is *True* or *False* and ϵ is the error bound.

4.4 Results

Quality of generated images We first present several Youtube frames as ground truth in Figure 6 to help readers check the quality of generated images. In Figure 7, we list real and GAN-generated images pairs, where the two rows present the transformation of Udacity dataset to snowy and rainy scenes, respectively, and the odd and even columns present original and GAN-generated images, respectively. Qualitatively, the GAN-generated images are visually similar to the images collected from Youtube and they also can keep the major semantic information (such as the shape of tree and road) of the original images. Interestingly, in the first snowy image in Figure 7, the sky is relatively dark and GAN can successfully render the snow texture and the light in front of the car. In the second column, the sharpness of rainy images are relatively low and this is consistent to the real scene showed in Figure 6. Our results are consistent with the original UNIT work [19], and further demonstrate the effectiveness of UNIT for image transformation.

Inconsistency of autonomous driving models We further present examples for the detected inconsistent autonomous driving behaviors in Figure 8. In the figure, each row shows the scenes of snow and rain, respectively. In each sub-figure, the blue caption indicates the model name, while the red and green captions indicate the predicted steering angles on the real and synthesized images, respectively. The curves visualize the predictions which help check the differences. From the figure we can observe that model Autumn (the first two columns) has the highest inconsistency number on



Figure 8: Inconsistency of steering angle prediction on real and synthesized images.

Table 2: Number of inconsistency behavior of three models under different weather conditions

Scene	Model	Num. of Incon. Behaviors			
		10°	20°	30°	40°
Snowy	Autumn	11635	11602	11388	10239
	Chauffeur	4839	2105	1093	653
	Rwightman	334	115	45	14
Rainy	Autumn	5279	5279	5279	5279
	Chauffeur	710	175	94	71
	Rwightman	656	92	23	0

both scenes; in contrast, model *Rwightman* (the last two columns) is the most stable model under different scenes. This figure shows that DeepRoad is able to find inconsistent behaviors under different road scenes for real-world autonomous driving systems. For example, a model like *Autumn* or *Chauffeur* [1] (they are both ranked higher than *Rwightman* in the Udacity challenge) may work perfectly in a fine day but can crash into the curbside (or even worse, the oncoming cars) in a rainy or snowy day (shown in Figure 8).

Table 2 presents the detailed number of detected inconsistent behaviors under different weather conditions and error bounds for each studied autonomous driving model on the Udacity dataset. For example, when using the error bound of 10° and the rainy scenes, DeepRoad detects 5279, 710, and 656 inconsistent behaviors for *Autumn*, *Chauffeur*, and *Rwightman*, respectively. From the table we can observe that the inconsistency number of *Autumn* is the highest under both weather conditions. We think one potential reason is that *Autumn* is purely based on CNN, and does not utilize prior history information (e.g., via RNN), and thus may not always perform well in all road scenes. On the other hand, *Rwightman* performs the most consistently than the other two models under all error bounds. This result presents a very interesting phenomenon – DeepRoad can not only detect thousands of inconsistent behaviors of the studied autonomous driving systems, but can also measure different autonomous systems in terms of their robustness. For example, with the original Udacity dataset, it is hard to find the limitations of autonomous driving systems like *Autumn*.

5 Related work

Testing and verification of DNN-based autonomous driving systems. Different from traditional testing practices for DNN models [29; 20], a recent set of approaches (such

as DeepXplore [23] and DeepTest [28]) utilize differential and metamorphic testing algorithms for identifying inputs that trigger inconsistencies among different DNN models, or among the original and transformed driving scenes. Although such approaches have successfully found various autonomous driving system issues, there still lack approaches that can test DNN-based autonomous driving system with realistic synthesized driving scenes.

GAN-based virtual/real scene adaption. GAN-based domain adaption has been recently shown to be effective in virtual-to-real and real-to-virtual scene adaption [32; 18]. DU-drive [32] proposes an unsupervised real to virtual domain unification framework for end-to-end driving. Their key insight is the raw image may contain nuisance details which are not related to the prediction of steering angles, and a corresponding virtual scene can ignore these details and also address the domain shift problem. SG-GAN [18] is designed to automatically transfer the scene annotation in virtual-world to facilitate real-world visual tasks. In that work, a semantic-aware discriminator is proposed for validating the fidelity of rendered image w.r.t each semantic region.

Metamorphic testing. Metamorphic testing is a classical software testing method that identify software bugs [33; 14; 27]. Its core idea is to detect violations of domain-specific metamorphic relations defined across outputs from multiple runs of the program with different inputs. Metamorphic testing has been applied for testing machine learning classifiers [22; 30; 31]. In this paper, DeepRoad develops a specific GAN-based metamorphic testing module for DNN-based autonomous systems, where the metamorphic relations are defined such that regardless of how the driving scenes are synthesized to cope with weather conditions, the driving behaviors are expected to be consistent with those under the corresponding original driving scenes.

6 Conclusion

In this paper, we propose DeepRoad, an unsupervised GAN-based approach to synthesize authentic driving scenes with various weather conditions to test DNN-based autonomous driving systems. In principle, DeepRoad applies the metamorphic testing methodology to detect the inconsistent autonomous driving behaviors across different driving scenes. The experimental results on three real-world Udacity autonomous driving models indicate that DeepRoad can successfully detect thousands of inconsistent behaviors. Furthermore, our results also show that DeepRoad can be promis-

ing in measuring the robustness of autonomous driving systems. Currently, DeepRoad only supports two weather conditions, we plan to support more weather conditions to fully test autonomous driving systems under various conditions in the near future.

References

- [1] Final leaderboard of udacity challenge 2. <https://github.com/udacity/self-driving-car/tree/master/challenges/challenge-2>.
- [2] Steering angle model: Autumn. <https://github.com/udacity/self-driving-car/tree/master/steering-models/evaluation>.
- [3] Steering angle model: Chauffeur. <https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/chauffeur>.
- [4] Steering angle model: Wrightman. <https://github.com/udacity/self-driving-car/tree/master/steering-models/evaluation>.
- [5] Tesla autopilot system. <https://www.tesla.com/autopilot>.
- [6] Tesla model s crash. <https://www.teslarati.com/tesla-model-s-firetruck-crash-details/>.
- [7] Udacity challenge 2 dataset. <https://github.com/udacity/self-driving-car/tree/master/datasets>.
- [8] Udacity pre-trained models. <https://github.com/udacity/self-driving-car/tree/master/steering-models/evaluation>.
- [9] Udacity self driving car. <https://github.com/udacity/self-driving-car>.
- [10] Youtube video: Driving on snow - greenville, nc 1-4-2018 at 7:00am. <https://www.youtube.com/watch?v=ps56tnQG8V0>.
- [11] Youtube video: Rain on a car roof - 1 hour - asmr. <https://www.youtube.com/watch?v=088fXBx-Qdg>.
- [12] Paul Ammann and Jeff Offutt. *Introduction to software testing*. Cambridge University Press, 2016.
- [13] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [14] Tsong Y Chen, Shing C Cheung, and Siu Ming Yiu. Metamorphic testing: a new approach for generating next test cases. Technical report, Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, 1998.
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [18] Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P Xing. Semantic-aware grad-gan for virtual-to-real urban scene adaptation. *arXiv preprint arXiv:1801.01726*, 2018.
- [19] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.
- [20] Alexis C Madrigal. Inside waymos secret world for training self-driving cars. *The Atlantic*, 2017.
- [21] William M McKeeman. Differential testing for software. *Digital Technical Journal*, 10(1):100–107, 1998.
- [22] Christian Murphy, Gail E Kaiser, Lifeng Hu, and Leon Wu. Properties of machine learning applications for use in metamorphic testing. In *SEKE*, volume 8, pages 867–872, 2008.
- [23] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 1–18. ACM, 2017.
- [24] Dean A. Pomerleau. Advances in neural information processing systems 1. chapter ALVINN: An Autonomous Land Vehicle in a Neural Network, pages 305–313. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989.
- [25] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.
- [26] S. Segura, G. Fraser, A. B. Sanchez, and A. Ruiz-Corts. A survey on metamorphic testing. *IEEE Transactions on Software Engineering*, 42(9):805–824, Sept 2016.
- [27] Sergio Segura, Gordon Fraser, Ana B Sanchez, and Antonio Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on software engineering*, 42(9):805–824, 2016.
- [28] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deepertest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering, Gothenburg, Sweden, May 27 - June 3, 2018, ICSE 2018*, 2018.
- [29] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [30] Xiaoyuan Xie, Joshua Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. Application of metamorphic testing to supervised classifiers. In *Quality Software, 2009. QSIC'09. 9th International Conference on*, pages 135–144. IEEE, 2009.
- [31] Xiaoyuan Xie, Joshua WK Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. Testing and validating machine learning classifiers by metamorphic testing. *Journal of Systems and Software*, 84(4):544–558, 2011.
- [32] Luona Yang, Xiaodan Liang, and Eric Xing. Unsupervised real-to-virtual domain unification for end-to-end highway driving. *arXiv preprint arXiv:1801.03458*, 2018.
- [33] Zhi Quan Zhou, DH Huang, TH Tse, Zongyuan Yang, Haitao Huang, and TY Chen. Metamorphic testing and its applications. In *Proceedings of the 8th International Symposium on Future Software Technology (ISFST 2004)*, pages 346–351, 2004.