Apply Image-to-Image Translation on Autonomous Driving Systems Testing

Yilin Han
Computer Science
University of Toronto
Toronto Ontario Canada
yilinhan10@cs.toronto.edu

Zi Yi Chen
Computer Science
University of Toronto
Toronto Ontario Canada
zychen@cs.toronto.edu

ABSTRACT

The rapid advancement in machine learning techniques has enabled researchers and technology companies to develop many applications that can improve the quality of human lives. One of these applications is autonomous driving systems (ADS). We believe by taking human out of the task of driving, we can reduce the traffic on the road and fatal car accidents, as over 90 percent of the accidents on the road are caused by driver error [24], hence saving us time and lives. Tech companies and car manufacturers are competing to be the first one to this market. As with regular automobiles, ADS should be tested thoroughly to ensure the safety of other road users. However, many ADS are developed using machine learning techniques such as Deep Neural Networks (DNN), and machine learning systems require different testing methodologies than traditionally software testing [23]. Although verification on the machine learning system's performance can be achieved, verification on the safety of such systems is still an unknown.

DeepTest [26] and DeepRoad [29] are two of the recent studies on verification of DNN ADS. Both frameworks use Metamorphic Testing (MT) [22] to determine whether the ADS is predicting inconsistent driving behaviours. DeepTest uses simple affine transformations and various effect filters to generate test cases from original image data, while DeepRoad uses Generative Adversarial Network (GAN)-based image generator to generate realistic driving scenes. However, the Metamorphic Relation (MR) used in those two studies does not reflect real-life driving behaviours.

In our project, we implemented a naive image generator, similar to the one used in DeepTest, and we leveraged another recent published GAN-based algorithm Pix2Pix [11] to train our realistic image generator and compare the performance of both. We propose a Metamorphic Testing method that reflects real-life driving behaviours to compare the performance of various ADS from Udacity [27].

KEYWORDS

Autonomous Driving System, Metamorphic Testing, Machine Learning, Convolutional Neural Network, Recurrent Neural Network, Generative Adversarial Network

1 INTRODUCTION

Autonomous vehicle is the future research direction of automobiles, and it has a profound impact on the automotive industry and even the transportation industry. The advent of driverless cars will be able to liberate human hands, reduce the frequency of traffic accidents, and ensure people's safety. At the same time, with the breakthrough and continuous advancement of technologies such as artificial intelligence and sensor detection, autonomous vehicle can be even more efficient and effective. However, implementation of autonomous vehicle in the worldwide still requires the unremitting efforts from researchers and scientists.

As one of the pioneers to develop unmanned technology, Google has built fully-automated driving vehicles that can automatically start and stop. The Google autopilot project was reorganized into an independent company called Waymo. Waymo announced November 7th, 2017 that it will test unmanned cars without safety drivers [9]. Unmanned vehicle testing is deployed by Waymo in 2018. In addition to the traditional automotive industry and the internet companies, Apple and Uber etc. have also joined the competition to build driverless cars.

Today, various levels of ADS have been deployed in production vehicles. Motor vehicle users can experience level 1 autonomy in most modern vehicles, while some models such as the Tesla Model S offers level 2 autonomy via its Autopilot system [13]. Even though all these autonomous driving systems are logging tons of mileage, they still contain critical safety concerns. Driver fatality involving a Tesla Model X on March 23rd, 2018 [25] and pedestrian fatality involving a Volvo refitted with an autonomous driving system on March 18th, 2018 [17] show that ADS are not ready for mass deployment.

These incidents also show the importance of testing autonomous driving systems. But traditional software testing methods, such as code coverage, have shown to be difficult to apply to deep neural networks as they lack explicit control-flow structure [23]. Recent studies have demonstrated that adding error-induced inputs to the training datasets can help improve the reliability and accuracy of existing autonomous driving models [18]. DeepTest and DeepRoad are two of the works that attempt to generate realistic test cases to mimic real-world driving scenes by transforming original observed image data using various effect filters and Generative Adversarial Networks (GAN), respectively. Despite the realistic test cases that DeepRoad can generate, both DeepTest and DeepRoad use metamorphic testing algorithms to detect inconsistent behaviours from the autonomous driving systems.

DeepRoad assumes an autonomous driving system's steering angle prediction does not change after modifying the weather condition of driving images. It identifies inconsistent driving behaviours by comparing steering angles from original observed images with clear road conditions to the generated synthetic images with snow or rain conditions. However, it is obvious that driving under different road conditions will require different steering, gas, and brake inputs. Therefore, different steering angle prediction in snow condition when compared to the clear condition should not be considered as inconsistent driving behaviours. Although DeepRoad relaxes its assumption and accepts if its prediction is within an error bound of the expected steering angle, it does not discuss how to determine a safe error bound. We propose a testing method that inconsistent driving behaviours should be compared under the same condition. Since DeepRoad considers the test cases generated by DeepTest are unrealistic simply because the images are observed to be artificial, we want to investigate how the steering angle predictions differ between a naïve image generator and an image generator developed using machine learning.

This project has three objectives based on the shortcomings of DeepTest and DeepRoad:

1. We implement a naïve image generator with simple affine transformations and various effect filters, and we trained a GAN-based image generator to compare the difference in predictions between two image generators.
2. we propose a metamorphic testing approach with a different metamorphic relation compared to ones used in DeepTest and DeepRoad.
3. lastly, we evaluate our testing algorithm using multiple established autonomous driving systems from Udacity with statistical measurements.

2 BACKGROUND

In this section, we present some background knowledge on deep neural networks used in autonomous driving systems and adversarial attacks.

2.1 Deep Neural Networks (DNN) in autonomous driving systems

With the advancement in Computer Vision (CV) and Deep Learning (DL) technologies, many tech companies and automobile manufacturers are competing to develop their own autonomous driving systems. Most of the autonomous driving systems take input from various sensors such as cameras mounted on different positions of the vehicle, light detection and ranging sensors (LiDAR), infrared (IR) sensors, and vehicle information such as driving speed, and output the driving behaviours including steering angles, gas, and braking decisions to the controller of the vehicle [26]. The DNN in the autonomous driving system may contain a number of processing layers, like fully connected layers and convolutional layers, to abstract the input to identify objects such as stop signs, lane markings, and other road users.

There are two main types of DNNs used in autonomous driving systems. Feed-forward Convolutional Neural Networks (CNN) [14] are commonly used in machine learning applications such as image processing, video analysis, and natural language processing. In CNNs, the neurons in a convolution hidden layer may share the same weights and be connected to only some of the neurons in the next layer to decrease the training time. CNNs have been shown to be effective at object detection in autonomous driving systems as it resembles the human visual system which learns the abstract representation of the visual input.

Recurrent Neural Networks (RNN) [21] is another type of DNNs used in autonomous driving systems. Unlike Feed-forward CNNs, RNNs allow loops in the network to allow previous inputs to be considered in the prediction output. Long-Short Term Memory (LSTM) [21] network is a variant of RNN and it fixes the problem of vanishing gradients in vanilla RNN algorithm, which has been more favourable in autonomous driving systems.

2.2 Adversarial Attack

In deep learning, adversarial examples where inputs intended to add a small perturbation that may mislead a deep neural networks or machine learnings system to incorrect results [16]. Until now, there is no machine learning model is believed to be perfect. It is very common for them to make mistakes, even though it may have much smaller chances than humans. Machine learning models consist of a series of special transformation, and those transformations are extremely sensitive to small changes to any input change. Using this feature to modify or train a deep learning model is one of the most important research topics in the area of safety of machine learning models.

Recently, a group of researchers from Keen Security Labs in China were able to to create a "fake lane" that tricked the Autopilot system in a Tesla Model S by placing bright-coloured stickers on the road [10]. This caused the vehicle veering from the appropriate driving lane into the opposite lane on a test course.

2.3 Autoencoder

Autoencoder is a special neural network that has the same input and output size [12]. Autoencoder consists of two processes (Figure 1): encode and decode. The input images get compressed through the encode process to get the code, then the code will be processed by the decoder and reconstruct the input images [12]. This is like forcing lower dimension parameters to learn higher dimension input. The goal of an autoencoder is to reconstruct the output image to be as close as the input image. The performance of an autoencoder is measured by cost function below (Formula 1), such that we compare the L2 loss on input images and output images at the pixel level, the lower the value of the cost function, the more accurate it is. During the training, the cost function value is computed, and the model is using backpropagation to optimize the parameters.

$$\text{Cost Function:} \quad J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Formula 1: Formula of cost function

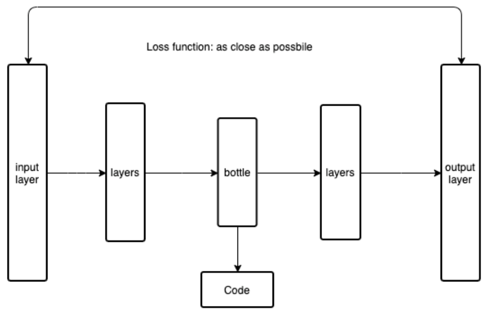

Figure 1: Sample structure of an autoencoder

2.4 Generative Adversarial Networks

Generative Adversarial Network (GAN) [8] is one of the deep neural network models. It is one of the most promising unsupervised learning algorithms in the recent couple of years. The initial goal of GAN is to generate data that does not exist in the real world. It is aiming to make artificial intelligence more creative or imaginative. It is the general solution for image-to-image translation (figure 2). The application scenarios are as follows:

1. AI writers, AI painters and other AI system that requires creativity
2. Sharpening the blurry pictures, such as remove rain, fog, mosaic, etc. It requires AI to have a so-called "imagination" that can replenish the images.
3. To perform data augmentation. Generating new data based on existing data to feed a machine learning model. Therefore, it is used to enhance the performance of a model and prevent overfitting in model training.

Typically, a GAN system is consisting of two components. Using the image-to-image translation as an example, assume we have two neural networks, G(Generator) and D(Discriminator). As their names imply, G is a generative model that generates an image with a random noise z. We call this output image G(z). D is a discriminative model that identifies an image is real or fake. Its input is an image and output are values between 0 and 1 where the higher the value is the more confident that the discriminator believes the image is real. In the training process, the goal of generative network G is to generate a real picture as much as possible to deceive the discriminative network D. The goal of D is to separate the G-generated image from the real image. Thus, G and D constitute a dynamic game process.

What is the result of the final game? In the most ideal state, G generates a picture G(z) that is spoofed. For D, it is difficult to determine whether the picture generated by G is true or not, so D(G(z))=0.5. Therefore, when the state reached, we trained a generative model G that can be used to generate images.

In real practice, GAN is a popular method to generate adversarial examples. For instance, defending adversarial examples. It is very prevalent in implementing an adversarial training routine in their DNN model training [14]. Adversarial training helps the model against overfitting. Another improvement of adversarial training is it shortens the model training time. The model is more likely to converge since the adversarial examples make the model more generalized. For the safety-critical systems, using GAN for this purpose is debatable. Nobody can guarantee that a well-trained GAN will never generate false positive results.

Another popular usable of GAN is data augmentation. Effective training of neural network model can require billion of examples of data [2]. Collecting data is a very time and money consuming event. Researchers and Scientists have tried to generate data with GAN for a couple of years. One of the promising examples is CycleGAN [30], which provides image-to-image translation in unsupervised learning manner. However, using such tools to perform data augmentation in a safety critical system is not well accepted. The effect of using these training examples is unclear, even though they may look very natural to the human. In other words, the model trained using GAN trained images may be the model generated by adversarial examples. It could mislead the system only able to recognize the adversarial examples and misclassified the real examples.

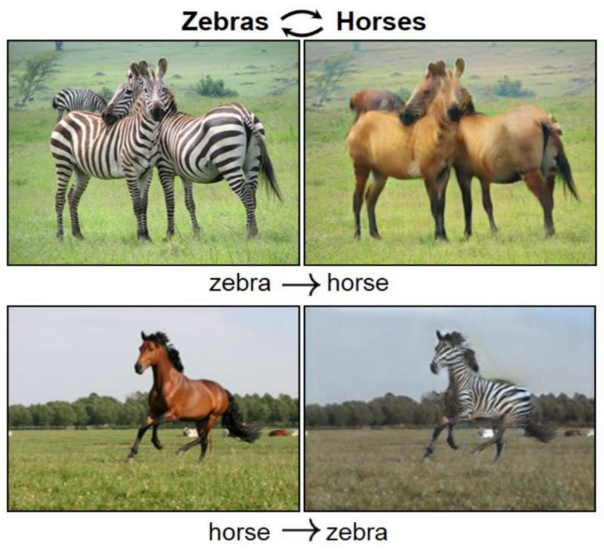

Figure 2 [30]: Image-to-image translation example generated by pix2pix

3 METHODOLOGY

3.1 Naïve Image Generator

Generative Traditional computer vision algorithms have provided strong power on images processing. Compare to Deep learning, the advantages of applying the traditional algorithm is obvious: it is explainable and easy to apply. The disadvantage is also apparent, its application is limited.

Detecting certain objects, such as tree or car is very difficult. Therefore, it is extremely difficult to seek an efficient way to perform complex image-to-image translation. In this project, we utilize OpenCV [6] to perform a series of linear or nonlinear transformation to convert the image from day to night. More specifically, we applied Gamma correction [28] and reducing of brightness on the images to make them look like they are taken at night.

Gamma correction [28] is a nonlinear operation that can be applied to an image or video to adjust its luminance or tristimulus value. Gamma is a relationship between a pixel's numerical value and its actual luminance. It modifies the pixels that make the shades closer to the human eyes' observation. The Gamma formula (Formula 2) shows V_in and V_out are value input and output. A is a constant, wherein this project is 1. When γ >1.0, it increases the grayscale value of a pixel that makes the image looks dark. When γ < 1.0, it increases the grayscale value that makes the image brighter. In the project, we set the γ =1.67 to make pictures darker so they look more like in the evening. Another advantage of applying gamma correction on this project is it will enhance the contrast of the image and look at the yellow lines and white lines on the pictures closer to the ones at night.

We further reduce the brightness of the V channel [19] of the images. The darkness of the image is changed gradually from the top to the bottom. The reason behind this is because the sky is dark at night and the area in front of the car is bright due to the headlamps.

$$V_{\text{out}} = A\, V_{\text{in}}{}^{\gamma}.$$

Formula 2: Formula to calculate Gamma. A is a constant. V stands for input and output vector.

3.2 GAN-based Image Generator

We investigated techniques that build image-to-image translation and picked Pix2Pix as our day-to-night image generator. Pix2Pix is a Conditional Generative Adversarial Network. It is conditioning the latent space on images. It feeds a combination of random noise and a conditioned image as input rather than traditionally only with random noise. Thus, the generator is learning the underlying conditional probability distribution. We tuned the Pix2Pix frameworks a couple of times and pick UNet-256 [20] and PathGAN as the generator and the discriminator.

UNet (Figure 3) is a special autoencoder structure that encoding layers skipped the bottleneck layers to connect to the decoding layers directly. The input of UNet is a 256*256 pixels image, then there are sequential convolutional layers that encode the images into a vector of 512 features. The decoding layers remain the same network structure but in reverse order. That is, it will reconstruct the image from the 512 features. Unlike the traditional multilayer perceptron that layers only connect to the one directly behind it, the encoding layers also transfer information to corresponding decoding layers. This attribute in UNet helps the generator to construct much clear translated images. In addition, dropout is enabled in UNet in order to prevent overfitting of the same data. It also saves the computational power and speeds up the converging time.

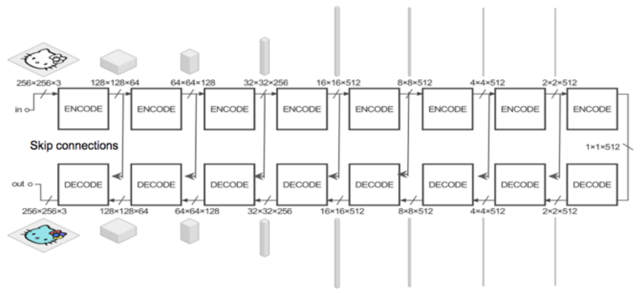

Figure 3: Unet [20] neural network framework structure.

The innovation happens on the discriminator as well. In Pix2Pix, it uses an architecture called "PatchGAN" [11] (Figure 4). Traditionally, the discriminator only outputs a value to indicate whether the input image is real or fake. PatchGAN is also sequential of convolutional neural network where the output is a 30*30 matrix. Each value in the matrix is a value that indicates how real this part of the input image is. More specifically, each value in the output responding to the probability about how real a patch/segment, size 70*70, from the input images. Since the input image is 256*256, patches are overlapped. The advantage of such structure is it makes the model more sensitive to the patch of input images during the training. It improves the training quality.

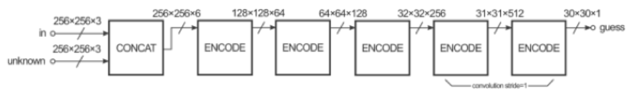

Figure 4: PatchGan neural network structure from Pix2Pix [11]

3.3 Metamorphic Testing (MT)

Traditional testing techniques such as automated tests verify the output of the software applications against the expected values. This is called an oracle [22]. However, there are situations where a test oracle is not easily determined, especially in machine learning applications. MT can solve this test oracle problem by making use of the known relations between the inputs and the outputs and check the software for those relations. The relations are also called Metamorphic Relations (MR). Thus, MT can generate test cases using defined MR to test machine learning models. For example, suppose *f* is a program that maps inputs *x* into outputs *y*, such that *f(x)* = *y*. And if we know that for a function *g*, the program outputs of *g(x)* should be the same as the program outputs of *x*, then we can define the MR as the following:

$$f(x) = f(g(x))$$

Both DeepTest and DeepRoad assumes an MR similar to the one described above. The autonomous driving systems used in their studies can be considered as the f program, the inputs to their program are the image frames from the autonomous vehicle camera, the outputs of their program are the steering angles for the vehicle, and the function g is their image generating functions. They claim that the predicted steering angles under clear road condition should be the same as the predicted steering angles under snow or rain conditions. And they consider the driving behaviour of the ADS to be inconsistent if those predicted steering angles are different. However, as we all know, driving under snow or rain conditions require different steering, gas, and braking inputs as driving under clear road condition. We are more cautious when we cannot see clearly the road ahead, or when the vehicle has poor traction with the road. Therefore, it is unrealistic to assume that the predicted steering angles should be the same regardless of the road conditions. In this project, we propose utilizing MT with an MR that reflects more closely with human driving behaviours.

On top of the same f, g, x, y variables as described above, we also introduce a function c as a classifier that can detect the road condition such that $c(z) = c(g(x))$ if the image frame z is under the same road condition as the transformed image frame from $g(x)$. We assume that the prediction steering angles should only be the same under similar road conditions. Thus, we propose the following MR for our testing methodology:

$$f(z) = f(g(x)) \quad \textit{iff} \quad c(z) = c(g(x))$$

4 Experiments

4.1 Data Collection

In order to satisfy our Metamorphic Relation, we need image frames on the same roads under different conditions. We have reviewed the dataset available from Udacity, but we could not find two sets of data that meet this requirement. Therefore, we filmed videos in various streets in downtown Toronto under different conditions to be used as our dataset. Ideally, we would like to have datasets under difficult driving conditions such as snow, rain, or foggy conditions. But it was difficult to achieve given the unpredictability of the weather this season and our schedule conflicts. In the end, we filmed videos of the streets in the day time and videos of the same streets in the night time as our MR also holds under different lighting conditions.

Once we have enough video footage of different road conditions, we review our dataset and discard the parts with large discrepancies between two videos such as vehicle changing lanes. We utilize iMovie to compare and match the day and night videos in the scope of the frame. Afterward, we use OpenCV to extract the frames from the video. Each day and night frames were resized to 640*480 with the built-in functions in OpenCV [6]. Then, the extracted image frames from those videos to be used for training and testing. In the end, we ended up with 2300 pairs of image frames with daytime and nighttime conditions.

4.2 Model Training

With our datasets pre-processed, we began to train our Pix2Pix model. Each pair of image frames are combined into one file, with the day time frame on the left and night time frame on the right. The combined image frames are then fed to the data loader to train our Pix2Pix image generator. We trained a number of models with different parameters and we found the results when training with 300 epochs and no flipping the images were the most realistic. With the image generator model trained, we generated test cases to be used in the autonomous driving systems.

4.3 ADS Models

We used the following three ADS available from Udacity to predict steering angles with our datasets: Rambo, Chauffeur, and Rwightman. These three ADS models were ranked second, third, and fifth places in the Udacity Challenge, respectively.

4.3.1 Chauffeur. Chauffeur consists of one CNN to extract the input images features and one RNN module with LSTM to predict the steering angle using the concatenation of 100 features extracted by the CNN module from previous 100 consecutive images. This model is implemented by Tensorflow [7] and Keras [4] frameworks.

4.3.2 Rambo. Rambo model consists of three CNNs. Two of the CNNs are inspired by NVIDIA's self-driving car architecture [3], and the third CNN is based on comma.ai's steering model [5]. Rambo takes the differences among three consecutive images as input instead of taking individual images as input. The outputs from the three CNNs are merged using a final layer. This model is implemented by Keras [4] and Theano [1] frameworks.

4.3.3 Rwightman. Rwightman is not an open-sourced model. Although the architecture of the model is unknown, it showed very favourable results in DeepRoad. Just as in deep road, we used it as black-box testing and included it in our analysis.

4.4 Results

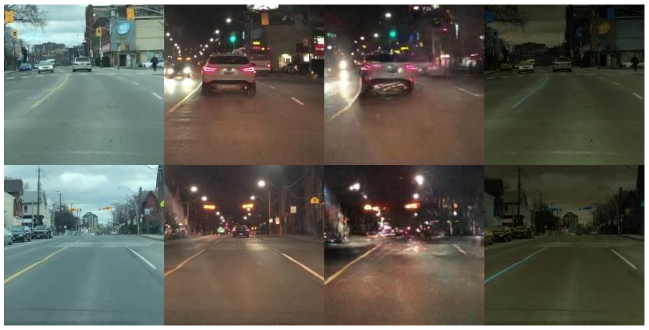

Figure 5: Two samples about the real images and machine generated images. The order (from left to right) is the daytime image, the nighttime image, the GAN generated image, and the CV generated image.

Figure 5 shows the test results from our image generator. The first two images are real images that captured by camera. The image generated with the naive image generator is simply a darker version of the original daytime image. It does not capture the street lights and taillights of the other vehicles. Headlights of the oncoming vehicles can be observed but beam reflection on the ground is not reflected. This is limited by the simple transformation that can be done using traditional computer vision algorithms, resulting in an unrealistic image when compared to the image generated from the GAN-based image generator.

With images generated with both generators, we use them as input to the three ADS mentioned in section 4.3 to predict the steering angle in each scenario. Rather than having an arbitrary error bound and count the number of inconsistent driving behaviours that each ADS predict like DeepTest and DeepRoad, we analyze the results by taking the difference between the predicted angle from synthetic night image frames and the predicted angle from original night image frames, computing the means and standard deviations of each of the three ADS to determine which model is more reliable. Table 1 shows the results of our test.

	Mean	Standard Deviation
Chauffeur GAN	1.40	1.08
Chauffeur Naïve	1.44	1.08
Rambo GAN	19.93	14.89
Rambo Naïve	27.18	18.06
Rwightman GAN	1.07	0.88
Rwightman Naïve	1.21	0.94

Table 1: Mean and standard deviation of the differences between predicted steering angles using original nighttime images and synthetic nighttime images.

As shown in the table above, the predicted steering angles using synthetic images generated from our naive image generator have higher discrepancy when compared to the predicted steering angles using original nighttime images. This is consistent across all three ADS models. However, some ADS models are more sensitive to artificial looking image frames than others. From the three ADS models we tested, the predictions from unrealistic looking test cases are between 3% to 27% less accurate compared to predictions from realistic looking test cases. When comparing the steering angle predictions between ADS, Table 1 shows Rwightman's results are the closest to the expected prediction with the smallest standard deviation, while Rambo's results are the worst among the three tested ADS. This is contrary to our expectation since Rambo was ranked higher than Chauffeur and Rwightman in the Udacity challenge. In the end, we think using these metrics gives us more meaningful results on how big the variances are with respect to real life images and synthetic images, as these figures reflect the safety performance of the ADS model. This can also be the key testing measurement for ADS model evaluation and comparison.

These tests are conducted with pairs of images. And each pair images have the same road condition since the synthetic image of the pair was generated using the daytime image from the daytime nighttime image pairs we extracted in previous steps. However, this model is not useful if we need a real nighttime image for every synthetic nighttime image. We attempted to implement the image classifier mentioned in section 3.3 that can classifier the road condition of the image. The classifier should be able to determine the weather and whether the road is straight or curved. We implemented the classifier using autoencoder as it can compress the images into a lower dimensional latent vector. Our goal was to compare the latent vectors to determine whether two images are under the same road condition. However, the results of our implementation did not show any consistency in classifying road conditions. This classifier requires investigation on more sophisticated machine learning techniques. Therefore, we have deferred this as part of our future work.

5 RELATED WORKS

5.1 DeepTest

The authors for DeepTest realized DNNs used in ADS demonstrate incorrect or unexpected corner-case behaviours which lead to potentially fatal accidents. And they proposed a framework to automatically generate test cases for ADS. The framework leverages the neuron coverage concept that is similar to the code coverage testing tools for traditional software and generate synthetic images that maximize neuron coverage to test the DNN. The images are generated using different types of transformations such as translation, scale, rotation, contrast, and brightness.

DeepTest also uses MT to detect inconsistent driving behaviours. The authors tested the framework in Chauffeur, Rambo, and Epoch ADS models available from Udacity. They found that the synthetic images generated by their method achieve significantly higher neuron coverage compared to their baseline cumulative coverage. When testing for inconsistent driving behaviours, they reported 4448, 741, and 821 erroneous behaviours across the three models with an error bound $\lambda = 5$, respectively.

5.1 DeepRoad

The authors in this paper have criticized the test cases generated with DeepTest look unrealistic. Even if the model detects inconsistent behaviours in ADS prediction, it is difficult to conclude whether the ADS is unsafe, or the test cases are unreliable. They proposed a GAN-based metamorphic testing approach to generate test cases that reflect authentic driving scenes.

DeepRoad leverages UNIT [15], a DNN-based unsupervised image-to-image transformation framework, to achieve the goal of generating realistic test cases. The framework consists of a GAN and VAEs. It is trained with datasets from Udacity and selected scenes from Youtube videos where snow or rain scenes are observed. DeepRoad generates synthetic images with snow or rain road conditions by inputting image frames with clear road condition. The framework uses Metamorphic Testing techniques with a Metamorphic Relation that the predicted steering angles under clear road condition should be the same as the predicted steering angles under snow or rain conditions.

The authors tested the DeepRoad framework in three different Udacity ADS models, and it concluded that Autumn produced the most inconsistent driving behaviours, while Rwightman is the most reliable ADS of the three.

6 CONCLUSION

In this project, we proposed a framework with two image generators that can synthesize driving scenes for nighttime road conditions to test DNN-based ADS. We apply Metamorphic Testing technique to determine the reliability of the ADS model by comparing its prediction on the synthesized image and an original image with the same road condition. The experimental results show that GAN-based image generator is able to outperform a naive image generator similar to the one used in DeepTest. And our results show that our testing algorithm can be used to measure the robustness of an ADS model compared to others. Currently, our GAN-based image generator can only generate images with low lighting condition such as at nighttime. For future work, we plan to support more road conditions such as snow, rain, and foggy conditions. We also plan to explore other machine learning techniques to improve our image generating results and implement our image classifier.

REFERENCES

[1] Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., ... & Bengio, Y. (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688.

[2] Antoniou, A., Storkey, A., & Edwards, H. (2017). Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340.

[3] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zhang, X. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

[4] Chollet, F., et al. Keras. https://github.com/fchollet/keras. (2015).Accessed 2019-04-05.

[5] Comma.ai's steering model. https://github.com/commaai/research/blob/master/train_steering_model.py. Accessed 2019-04-12.

[6] Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012, May). A brief introduction to OpenCV. In 2012 proceedings of the 35th international convention MIPRO (pp. 1725-1730). IEEE.

[7] Girija, S. S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Software available from tensorflow. Org.

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[9] Hawkins, A., Waymo is first to put fully self-driving cars on US roads without a safety driver. https://www.theverge.com/2017/11/7/16615290/waymo-self-driving-safety-driver-chandler-autonomous. Accessed 2019-04-12.

[10] Huddleston, T. These Chinese hackers tricked Tesla's Autopilot into suddenly switching lanes. https://www.cnbc.com/2019/04/03/chinese-hackers-tricked-teslas-autopilot-into-switching-lanes.html.Accessed 2019-04-03.

[11] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

[12] Kaggle,How Autoencoder work - Understanding the math and implementation. https://www.kaggle.com/shivamb/how-autoencoders-work-intro-and-usecases.Accessed 2019-04-03.

[13] Korosec, K. Tesla Autonomy Investor Day is here; here's how to watch. https://techcrunch.com/2019/04/22/tesla-autonomy-day-is-here-heres-how-to-watch/.Accessed 2019-04-07.

[14] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[15] Liu, M. Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. In Advances in Neural Information Processing Systems (pp. 700-708).

[16] Mikhailov, E. & Trusvo, R., How Adversarial attacks work. https://blog.ycombinator.com/how-adversarial-attacks-work/.Accessed 2019-04-07.

[17] Pavia, W. Driverless Uber car 'not to blame' for woman's death. https://www.thetimes.co.uk/artichttps://www.thetimes.co.uk/article/driverless-uber-car-not-to-blame-for-woman-s-death-klkbt7vf0le/driverless-uber-car-not-to-blame-for-woman-s-death-klkbt7vf0. Accessed 2019-04-07.

[18] Pei, K., Cao, Y., Yang, J., & Jana, S. (2017, October). Deepxplore: Automated whitebox testing of deep learning systems. In proceedings of the 26th Symposium on Operating Systems Principles (pp. 1-18). ACM.

[19] Poynton, C. (2001). YUV and luminance considered harmful: A plea for precise terminology in video.

[20] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer,Cham.

[21] Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.

[22] Segura, S., Fraser, G., Sanchez, A. B., & Ruiz-Cortés, A. (2016). A survey on metamorphic testing. IEEE Transactions on software engineering, 42(9), 805-824.

[23] Sekhon, J., Fleming, C. (2019). Towards Improved Testing For Deep Learning. arXiv preprint arXiv:1902.06320.

[24] Singh, S. (2015). Critical reasons for crashes investigated in the national motor vehicle crash causation survey (No. DOT HS 812 115).

[25] The Tesla Team. An Update on Last Week's Accident. https://www.tesla.com/blog/update-last-week%E2%80%99s-accident. Accessed 2019-04-02.

[26] Tian, Y., Pei, K., Jana, S., & Ray, B. (2018, May). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In Proceedings of the 40th international conference on software engineering (pp. 303-314). ACM.

[27] Udacity pre-trained models. https://github.com/udacity/self-driving-car/tree/master/steering-models/evaluation. Accessed 2019-03-15.

[28] Understanding Gamma Correction. https://www.cambridgeincolour.com/tutorials/gamma-correction.htm. Accessed 2019-04-02.

[29] Zhang, M., Zhang, Y., Zhang, L., Liu, C., & Khurshid, S. (2018). Deeproad: Gan-based metamorphic autonomous driving system testing. arXiv preprint arXiv:1802.02295.42.

[30] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).