From Neuron Coverage to Steering Angle: Testing Autonomous Vehicles Effectively

Jack Toohey
Loyola University Maryland

M S Raunak
National Institute of Standards and Technology

Dave Binkley
Loyola University Maryland

Abstract

A Deep Neural Network (DNN) based system, such as the one used for autonomous vehicle operations, is a “black box” of complex interactions resulting in a classification or prediction. An important question for any such system is how to increase the reliability of, and consequently the trust in, the underlying model. To this end, researchers have largely resorted to adapting existing testing techniques. For example, similar to statement or branch coverage in traditional software testing, neuron coverage has been hypothesized as an effective metric for assessing a test suite’s strength toward uncovering failures and anomalies in the DNN. We investigate the use of realistic transformations to create new images for testing a trained autonomous vehicle DNN, and its impact on neuron coverage as well as the model output.

Introduction

On October 8, 2020, Waymo officially opened its driver-less riding service in three Arizona cities. Many automobile and technology companies, including Tesla, Uber, Volkswagen, Baidu, and others, are not far behind. At their core each employs a deep neural network (DNN) that learns to recognize objects in the vehicle’s environment and make split-second decisions based on current conditions.

There continues to be trust issues involving the safety and reliability of these systems [1]. Incidents such as the pedestrian fatality caused by an Uber SUV in Tempe, Arizona [2] exacerbated the situation. The primary approach for ensuring the reliability and correctness of these autonomous systems involves different software verification activities, especially testing. Effective testing of any complex software system is challenging. Furthermore, autonomous vehicles are primarily data-driven, statistical, and non-deterministic in nature, which make them even more difficult to properly test and their reliability harder to ensure.

Designing an effective test suite for verifying any system involves addressing two broad questions: a) how to select the test cases, and b) how many test cases to select. When source code is accessible, *code coverage* oriented criteria that use the structure of the source code are commonly employed to address these questions. One can aim to ensure that all statements or all branches in the source code are *covered*, i.e., executed by at least one of the test cases in the test suite. These two criteria are known as *statement coverage* and *branch coverage*, respectively. The motivation here is that a test suite is unlikely to reveal a bug located in a statement or a branch that has never been executed. Such coverage metrics are also helpful in providing concrete goals for selecting a sufficient number of test cases. One can set goals such as achieving 90% branch coverage to consider a test suite to be sufficient.

It is generally accepted that a test suite that provides higher *statement or branch coverage* better tests a piece of software [3]. This type of coverage metric, however, is not suitable when testing DNNs [4]. Consequently, researchers and practitioners have looked for other metrics and strategies to test DNN-based systems [5]. Consequently, newer metrics such as *neuron coverage* (NC), used by DeepXplore [6] and DeepTest [7], have emerged. Although the inadequacy of traditional test coverage criteria when applied to DNN-based systems is well established, the usefulness of new criterion such as NC is yet to be fully studied. While DeepTest [7] has provided results in favor of using NC as an effective test selection criterion, some newer studies have questioned its usefulness [8].

In the domain of autonomous vehicle operations, one of the primary test inputs is the contiguous set of front-view images and one of the computed outputs is the steering angle of the vehicle. Numerous research has been performed to look for effective ways of training a DNN-based machine learning algorithm to convert images into specific actions [9], [10]. Our focus in this paper, however, is on the challenges of testing these systems. Testing the trained DNN requires the selection of test images. Based on the differing findings regarding NC's usefulness, further exploration is necessary to see how NC is impacted by different test image selection strategies, and whether they lead to more effective testing of the underlying DNN model. This is what we investigate in this paper.

Another challenge comes from the fact that one needs test images that have not been seen by the DNN before and for which the correct steering angle is known, which provides a test oracle. Synthetically creating new test images from the existing ones with known expected driving angle addresses this problem. We utilize this approach in our study by deriving reasonable, synthetic test images from existing images. Specifically, we consider seven synthetic image transformations to gain insight into their effectiveness when testing autonomous vehicle software. The only image transformations considered are the ones that enable us to predict the expected steering angle, which solves the test oracle problem. Our work builds on the framework found in DeepTest [7]. We identify an important issue in that approach, update it, consider new and refined image transformations, all to better understand the interplay between NC and effective testing of autonomous vehicles.

Our research contributions presented in this paper include:

- Showing that the use of new test images created by applying transformations to existing ones, both individually and in groups, increases NC.
- Some transformations are more effective than others at achieving higher NC.
- There is a positive correlation between higher NC and the test-suite's ability to extract output deviations, which suggests NC's potential as a measure of test-suite quality.

The remainder of this paper first presents background material before considering our research setup and approach, which includes the two research questions we consider. This is followed by the presentation of our results and a discussion thereof. The paper finishes by considering related work and then presenting our conclusions.

Background and Relevant Concepts

Neural Networks and Neuron Coverage (NC)

A neural network is a computing structure that attempts to mimic the design and behavior of a human brain. At its core is a computing unit called a neuron (or a perceptron). The neurons are placed in layers with edges connecting one layer to the next. There are weights associated with the edges that connect neurons. Based on the inputs and the weights, a non-linear activation

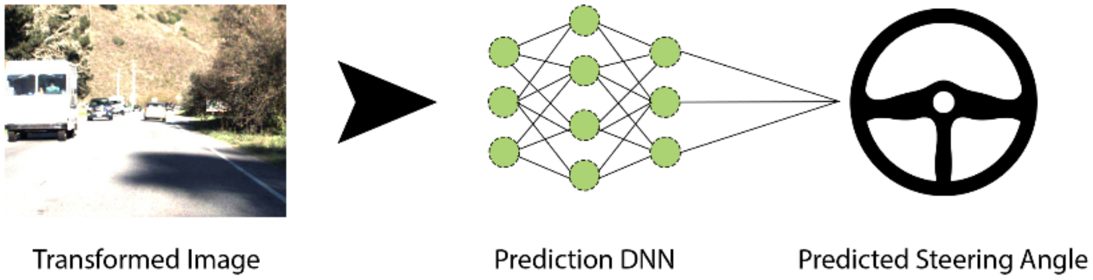

Figure 1. From image to steering angle output using a DNN

function is used to decide when a neuron is activated and thus impacts the neurons in subsequent layers. Typically, a neural network will have a layer of neurons for accepting inputs as well as a layer for producing the computed output. A deep neural network or DNN includes additional neurons in a series of hidden layers, which enable it to be used for complex computation such as image classification. Users interact with the first and last layer of a DNN, which handle input and output respectively. Neurons in intervening hidden layers, through their connection weights and activation functions, learn complex decision mechanisms that contribute to the overall output.

There is natural appeal to the notion of transferring traditional code coverage based test adequacy criteria to *neuron coverage* based test adequacy criteria for DNN model testing [6]. The idea here is to select a set of test cases (e.g., a set of images) that maximize the activated neurons. The underlying argument is that if certain neurons never get activated during testing, then it is possible that there may be undiscovered erroneous behavior associated with their activation that has never been witnessed. Despite this appeal, test effectiveness of complex structures such as a DNN may be uniquely different. Thus there is an ongoing need for the exploration of NC based test-adequacy criteria.

Autonomous Vehicles

An autonomous vehicle is a self-driving automobile that uses sensory devices such as cameras, infrared sensors, lidar, and global positioning systems (GPS) to navigate the world around it by making very fast decisions related to steering angle adjustment, acceleration, and braking. At the core of this complex process is a DNN that learns different vehicle operation responses from a massive amount of training data. The images from the front view camera are the primary data sources influencing the steering angle decision. As shown in Figure 1, a front-view image is fed into a trained DNN, which, in turn, predicts the steering angle that the self-driving car should maintain or adopt.

The output produced by a DNN is the result of a series of neurons being activated inside. The DNN's edge weights and neuron thresholds are determined during the training of the network. For a given input, the neurons of the neural network can be labeled either *active* when the weighted input value exceeds the threshold, or *inactive*.

Metamorphic Testing

Traditional software testing relies on the presence of a test oracle, which is capable of providing the correct output or expected behavior for a given test input. Automated software testing uses the oracle to identify failures of the software system where the output behavior deviates from that of the oracle. In the case of systems that are often termed as 'non-testable' [11] due to the absence of an oracle, such as cryptographic functions or scientific computations, metamorphic testing provides a way forward by producing two sets of inputs whose outputs should be the same or differ in a deterministically predictable way. Thus the two test cases can be used as pseudo-oracles for each other [12]. With DNN based systems, such as autonomous vehicles, producing oracle output is possible, but expensive. Thus pseudo-oracles, such as those provided by metamorphic testing, should be utilized whenever possible. This approach has been shown effective in testing DNN-based systems [13]. In our case we apply synthetic image transformations to create a set of new test images. The transformed images are metamorphic in nature, i.e., while these transformations modify the image they do not change the expected behavior of the autonomous vehicle. For example, we don't expect the steering angle to change if we transform an image by increasing its brightness or reducing its contrast. The expected behavior can be determined from the original oracle behavior and the transformation applied. One of the metamorphic transformations that we introduce, Flip, requires that the oracle output be computed by turning the steering wheel in the opposite direction.

DeepXplore

DeepXplore [6] showed that even a randomly selected set of test cases could achieve 100% statement coverage of a DNN while the share of internal neurons being activated was no more than 34%. This exemplified the inadequacy of traditional code coverage for testing a DNN. With the aim of exercising more of the DNN's internals, DeepXplore proposed neuron coverage as a metric to select effective test inputs. Their testing framework used two DNNs trained for the same purpose (e.g., classifying an image) as pseudo-oracles of each other and then generated test inputs that maximized both NC as well as images for which the two DNNs produced different classifications. Their study showed this approach to be highly effective in discovering corner cases of erroneous DNN behavior.

DeepTest

The DeepTest work [7] builds on the neuron coverage idea, and created synthesized images to test DNNs trained for autonomous vehicle operations. DeepTest applies transformations to images to mimic changes in the natural environment such as changes in sunlight, rain, fog, etc. Most of the transformations they use are metamorphic in that they do not change the autonomous vehicle's expected steering angle when compared to the original image. Thus DeepTest could automatically detect if the DNN was producing an erroneous behavior using the transformed images.

The DeepTest researchers worked with trained DNNs from the Udacity self-driving Challenge data set [14]. Their tool detects previously inactive neurons being activated due to a transformed image. It uses a greedy search algorithm that repeatedly applies transformations to a *single* test image in an attempt to maximize activated neurons before moving on to the next test image.

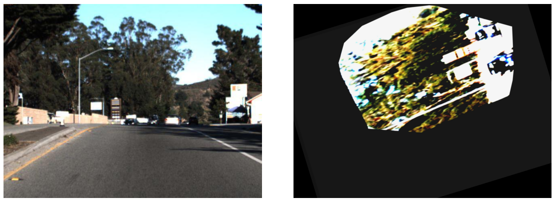

Figure 2. An unaltered test image (left) and the same image transformed by the DeepTest algorithm (right).

A possibly unintended consequence of the gradient descent search used in DeepTest is that it achieves high neuron coverage at the cost of potentially over-transforming images. Repeatedly modifying a single image causes ever increasing distortion leading to the original scene being almost unrecognizable. An example is shown in Figure 2 where the image on the right is the result of repeatedly transforming the image on the left. The fixation on increasing neuron coverage seems to void the metamorphic nature of the transformations. The use of such transformations in the experiments and the corresponding usefulness of the results become dubious. In our experiments, we build on the DeepTest framework, but remove the repeated transformations of an image to avoid this scenario.

Our Approach and Setup

We investigate the impact of test images synthesized through transformations on *neuron coverage* (NC) and predicted *steering angle* (SA). We start with the basic DeepTest tool and modify it to avoid the over transformation scenario illustrated in Figure 2. Instead, we *reset* the image back to its original, untransformed, version before applying subsequent transformations.

Image Transformations

Our study employs seven metamorphic image transformations. The chosen transformations allow us to predict the expected steering angle for the new images. Some of the transformations are more realistic in term of likely-to-occur while driving. For example, changes in brightness are caused all the time by clouds moving past the sun. We experimented with all the transformations used in the DeepTest study and found that some of the transformations (e.g., shear) were not metamorphic and were prone to change the images towards unrealistic scenarios. Our initial exploration led us to choose four transformations: Scale, Contrast, Brightness, and Blur. We saw a similar problem with translating an image simultaneously both in x and y direction. We therefore separate it into two transformations: Translate-X and Translate-Y, and limit the range. This ensured the generation of reasonable images after applying the transformation. Finally, we introduce a completely new metamorphic transformation, Flip, which horizontally flips an image. Unlike the other six, which do not effect the predicted steering angle, Flip requires changing its sign, i.e, moving the steering angle in the opposite direction. Through Flip we aim to create new synthetic test images that are clear and with known expected steering angle. Moreover, we are interested in observing the impact of NC when such pseudo new images are fed to the trained DNN. All the transformations except Flip take a parameter that dictates how much alteration to make to the image (e.g., the number of pixels to translate the image by). While not reported here, we also performed a systematic study of different parameter values used for transforming the images. Those experiments did not show any significant impact on NC or predicted steering angle with particular value combinations or values outside our selected range. That work did find however that a small number of random parameter values provides sufficient diversity. Thus, we randomly select parameter values from slightly narrower ranges than used in the DeepTest experiments. The narrower range ensures that the correct steering angle is unaffected by the transformation.

The following list details the seven transformations and the parameter range used for each.

1) *Translate X*: This transformation shifts the image left or right by the given number of pixels. The range considered from left shift to right shift is $[-X : X] - [-50 : 49]$.
2) *Translate Y*: This transformation shifts the image up or down by the given number of pixels. The range considered from up shift to down shift is $[-Y : Y] - [-50 : 49]$.
3) *Scale*: This transformation shrinks or enlarges the image along both the x and y-axes by a given percentage. The range of percentages considered is [0.5 % : 1.9 %].
4) *Contrast*: This transformation increases or decreases the contrast of the image by a given alpha value. The range considered is [0.5 % : 1.9 %].
5) *Brightness*: This transformation changes an image's brightness by a given bias parameter. The range considered is [-21 : 20].
6) *Blur*: This transformation blurs the image in one of three ways (chosen randomly) based on a parameter in the range [1 : 10].
7) *Flip*: This transformation flips the image across the vertical axis. No parameter here.

One of the contributions of our work is that, in addition to applying individual transformations, we also apply multiple transformations to create a new test image. We use the term *transformation group* to refer to one or more transformations. For example, when the transformation group {Flip, Contrast, Translate-Y} is applied to an image, the image is flipped, its contrast is adjusted, and it is translated along the y-axis. While we do not present the details here due to space, our experiments show that the application order of the transformations that make up a transformation group does not effect our neuron coverage results.

The Metrics Used In Our Study

To investigate the impact of different transformation groups, we must measure the increased NC associated with the application of a particular transformation group. We accomplish this using the metric *Isolation Neuron Coverage (INC)* defined as follows:

- *Isolation Neuron Coverage (INC)* is computed relative to a set of N unaltered images that are first run through the model to establish a baseline NC. The specific transformation group being studied is then applied to each image before it is run through the model to identify the number of additional neurons activated above the baseline. Between images the coverage is reset to the baseline thus isolating the contribution of each transformed image.

Our study also considers the steering angle predicted by the model. Each image in the data set includes the expected steering angle that should be produced by the DNN. In other words, we have a test oracle for the images making it possible to consider the accuracy of the DNN's steering angle prediction. We do so by computing the *Steering Angle Deviation (SAD)*:

- *Steering Angle Deviation (SAD)* is defined as the absolute value of the difference between the predicted steering angle and the oracle steering angle for a given image.

Of particular interest here is the potential to examine the effectiveness of NC as a predictor of test suite performance. Specifically, we are interested in knowing if test suites that produce higher NC also lead to the discovery of more anomalous behavior, which would be captured as greater SAD. This situation parallels a traditional test suite that provides greater code coverage being more likely to uncover more program faults. Should such a connection exist then test suites providing higher NC could be deemed better test suites.

Research Questions

We are interested in better understanding the impact of transformation on INC and SAD. Our initial working hypothesis is that higher NC is indicative of a stronger test suite, and a stronger test suite is going to be more effective in discovering anomalies (weakness in the model). To explore this relationship, we consider two key research questions:

- **RQ1** Do certain image transformations achieve higher neuron coverage than others?
- **RQ2** What impact, if any, do transformed images have on the predicted steering angle?

RQ1 actually goes beyond simply asking if transformation increases NC and considers the relative impact of different transformation groups. Then RQ2 factors in consideration of SAD in order to evaluate the effectiveness of neuron coverage as a predictor of test suite strength.

Experimental Setup

Our experimental setup builds on the DeepTest [7] framework, written using python version 2.7. We modified the framework such that image transformations are not repeatedly applied to the same image. Like DeepTest, we work with the Rambo model [15] from the Udacity self-driving challenge [14]. Rambo uses three separate Convolutional Neural Networks (CNNs) for determining the steering angle. With each application of the model we measured the number of its total 18899 neurons that were activated during each steering angle computation.

The Udacity self-driving challenge data includes 5000 still-frame images chopped from a thirty minute video, which was taken by a car as it drove down the road. The images are from a front-view video feed and thus taken from the point of view of a driver. In some of our experiments we used a sample of 100 images selected by picking every 50^{th} image.

Results and Discussion

RQ1: All Transformations Not Created Equal

Going beyond the question "Does transformation increase neuron coverage?" we investigate if certain transformation groups distinguish themselves. Such transformation groups are valuable if higher neuron coverage proves to be a useful metric for selecting test images.

Visually, Figure 3 shows the average INC gain for each transformation relative to the baseline. A statistical test using the analysis of variance (ANOVA) [16] finds a strong difference (p-value < 0.0001) and thus in Table 1 we show the results of Tukey's post-hoc Honest Significance Difference (HSD) test [16] applied to the *INC* $_7C_1$ data (the additional neuron coverage of the individual transformations). Here $_7C_1$ denotes the combinations of seven things (our transformations) taken one at a time. Thus $_7C_1$ refers to each of the seven transformations considered individually. In the resulting groups, shown in Table 1, transformations sharing a letter are not statistically separable. While the existence of overlaps means that there is no simple order, it is clear from the data that Flip and Contrast are top performers where Flip outperforms all the other transformations except Contrast. Next, Translate-X and Translate-Y are in the middle, where it is interesting that only Translate-Y can be separated

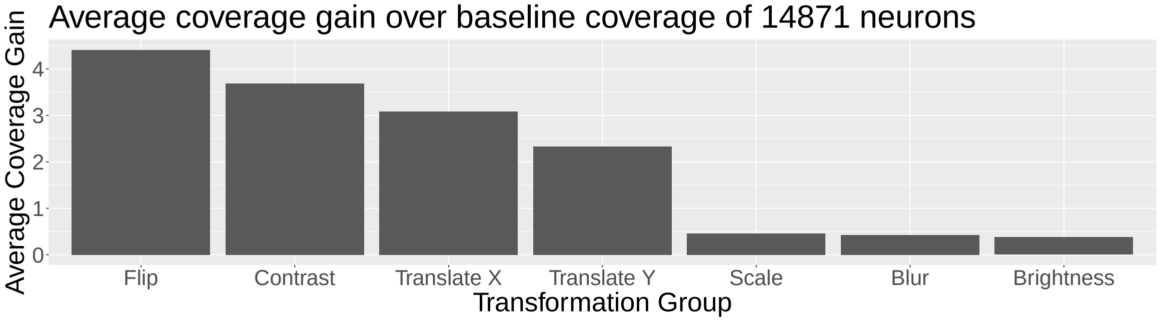

Figure 3. Transformation's impact on gain in Isolation Neuron Coverage (INC). Baseline includes 14871 neurons out of 18899 total neurons.

from Contrast. Finally, Scale, Brightness, and Blur all produce notably inferior increases.

We also considered the transformation pairs of *INC* ${}_7C_2$, i.e., combinations of two transformations from the group of 7. As shown in Table 2, the combination of Flip and Contrast with a mean of 8.45 distinguished itself from all the other pairs in terms of increased neuron coverage. While there is considerable overlap, two other main groups become evident. First, in the middle are pairs that include Translate X, and finally at the bottom are groups with *none of* Flip, Contrast, or Translate X. Looking at the combintions of three transformations applied to the images, i.e., from the *INC* ${}_7C_3$ produced data, there is greater overlap between transformation groups, but all the triples with both Flip and Contrast come before those with one of the two, which come before those without either of the two transformations. Looking at *INC* ${}_7C_i$ for $i > 3$ this basic pattern continues although as the number of transformations in the transformation groups increases, there is ever greater overlap.

In summary for RQ1, not only does transformation bring increased neuron coverage, but Flip and Contrast stand out. Thus where higher neuron coverage is the goal, these two transformations should be preferred. Intuitively, Flip changes the images drastically compared to others and thus likely requires more neurons to be activated to process those images. Similarly, it is possible that Contrast plays a relatively more important role in identifying the features that leads to the steering angle computation by the DNN.

Table 1. Transformation Cover Comparison

Transformation	Mean Neuron Count Increase	Group
Flip	4.41	a
Contrast	3.68	ab
Translate X	3.09	bc
Translate Y	2.33	c
Scale	0.46	d
Blur	0.43	d
Brightness	0.38	d

Table 2. Transformation Pair INC Comparison

Transformation	Mean INC Increase	Group
Contrast + Flip	8.45	a
Contrast + Brightness	5.93	b
...		
Translate X + Flip	4.27	$bcde$
Translate X + Brightness	3.40	cde
...		
Translate Y + Scale	2.27	ef
Brightness + Blur	0.58	f
Scale + Blur	0.55	f
Scale + Brightness	0.54	f

RQ2: Is there a Connection Between INC and SAD

Given transformation's impact on neuron coverage, a natural follow-up question is what impact, if any, does transformation have on the model's steering angle prediction? We intentionally limited our chosen transformations to preserve the known expected steering angle (Flip requires inverting the steering direction), which provides us with an oracle.

This leads us to investigate the relation between INC and SAD. Recall that INC is the neuron coverage achieved by each image in isolation and SAD is the absolute value of the difference between the model-predicted steering angle and the oracle steering angle. Here greater SAD is indicative of potentially anomalous behavior. Thus if transformation leads to greater deviation, then it has proven effective in uncovering potential bugs or weaknesses in the model.

As a preliminary investigation, we first consider transformation's impact on SAD. Similar to the pattern seen in Figure 3 when using transformation groups of size one, Flip leads to the largest mean SAD, followed by Contrast, and then Translate X. We also consider larger transformation groups where the same patterns perpetuate. For example, size-three transformation groups that include Flip and Contrast dominate the larger mean SAD values. Of the two, Flip more consistently leads to a greater deviation in steering angle. This pattern is reminiscent of the neuron coverage where Flip is a consistent top performer. Thus the data suggests a connection between neuron coverage and SAD.

Finally, we compare INC and SAD directly using linear regression [16]. We applied R's lm function using SAD as the response variable and INC as the explanatory variable. Aggregated over all seven transformations, lm yields the following relation with a p-value < 0.0001:

$$SAD = 0.043 + 0.0017 \times INC.$$

Most relevant to our research is that the slope of the regression line is positive indicating that, although small, *increased* INC is associated with *greater* SAD. This result persists in the larger transformation groups. For example, none of ${}_7C_3$'s 35 transformation groups yields a line with a negative slope.

Digging deeper, we include the interaction term between transformation group and neuron coverage. The model uncovers two interesting results. First, neuron coverage continues to have a positive coefficient (p-value $= 0.007$). Second, Flip differentiates itself (p-value = 0.0164) where the slope for Flip is three times steeper than the slope using the aggregated data.

In general a *stronger* test suite for a system is one that reveals more errant behaviors or failures. Reflecting on our two research questions, in the domain of autonomous vehicle operations and especially in terms of steering angle prediction, a *stronger* test suite would reveal more and larger SADs as indicative of model weakness. If that stronger test suite also achieves higher neuron coverage, then neuron coverage could be argued to be a useful proxy for the strength of a test suite.

Based on our data, we see a small positive correlation between transformations that generate higher neuron coverage and those that reveal more and larger mean steering angle deviations. In other words, the data suggests that higher neuron coverage may be an indicator of test suite strength.

However, care must be taken with this correlation as we do not know that the increase in neuron coverage is *causing* the increase in steering angle deviations. Regardless, it seems that transformed images, likened to test cases in traditional software, have the potential to discover erroneous behaviors as manifested by their propensity to yield higher deviations in the predicted output.

Threats to Validity

Our experiments and results have some limitations. We have worked with only one DNN model, Rambo [15]. Our initial set of images for testing the DNN was taken from the set used with DeepTest [7]. Increasing the external validity of our results by repeating our experiments with other DNNs, perhaps with other failure metrics, is an important part of our future work.

Another limitation of our analysis is that it does not isolate when the SAD rises to the level of grossly erroneous or failure inducing. Instead of arbitrarily selecting a threshold where we can say angle is erroneous, we looked simply at the relationship between NC and SAD.

Related Work

A number of studies have taken the approach of testing DNNs with synthetically created road images [6], [7], [17], [18]. Among them, DeepRoad [17] argues that the images used in DeepXplore [6] and DeepTest [7] do not realistically represent real-world driving scenarios. DeepRoad alleviates the problem by using a *Generative Adversarial Network* (GAN) [19] based technique to synthesize realistic driving scenes. However, the GAN based technique does not provide any guarantee of image creation preserving desired steering angle. Contrary to DeepTest's findings, Harel-Canada et al., [8], did not find neuron coverage to be an effective metric to derive new test images. They argue that individual neurons do not represent specific features of an image and thus the value in maximizing neuron activations is questionable.

Sun [20] proposed a new set of coverage criteria instead inspired by the modified condition/decision coverage (MC/DC) criterion of traditional software systems. The search for a more effective test image selection approach led the Deepgauge study [21] to propose a set of coverage criteria based on covering different sections of the network as well as the range of values output by a neuron (boundary coverage).

Conclusion

We investigated the use of image transformation to create new test images and how these images impact the neuron coverage achieved by a DNN. We first found that while transformation increases neuron coverage, certain transformations achieve higher neuron coverage than others. In fact our newly introduced image transformation, Flip, which does not require a new oracle for testing, achieves higher neuron coverage than other existing transformations. Furthermore, combinations of transformations also prove useful and often achieve even higher neuron coverage.

Our data shows a small, but positive, correlation between neuron coverage and steering angle deviations. Parallel to the neuron coverage results, the positive correlation is strongest with Flip. More importantly, we never found a negative correlation between neuron coverage and steering angle deviation among any applied transformations. This suggests that neuron coverage might act as a proxy for test-suite strength; however, there is a need for further investigation to discover whether neuron coverage is a consistent indicator of test-suite strength.

As DNNs become more integral to modern technology, more thorough understanding of these sophisticated black-box systems is warranted. This includes not only empirical testing work, such as presented here, but also theoretical work aimed at providing a deeper understanding. For DNNs that take images as their input, having an effective metric for selecting good test images is an important aspect of their testing. The use of transformed images that do not require a new test oracle, combined with neuron coverage, may prove a good candidate for this role.

Acknowledgements

Certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology (NIST), nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

1. Y. Ma, Z. Wang, H. Yang, and L. Yang, “Artificial intelligence applications in the development of autonomous vehicles: a survey,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020.
2. E. Besic, N. Zych, and J. Iverson, “Vehicle automation report hwy18mh010,” https://www.ntsb.gov/investigations/AccidentReports/Pages/HWY18MH010-prelim.aspx, March 2018.
3. H. Zhu, P. A. V. Hall, and J. H. R. May, “Software unit test coverage and adequacy,” *ACM Computing Surveys*, vol. 29, no. 4, p. 366–427, Dec 1997. [Online]. Available: https://doi.org/10.1145/267580.267590
4. J. Sekhon and C. Fleming, “Towards improved testing for deep learning,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, 2019, pp. 85–88.
5. J. M. Zhang, M. Harman, L. Ma, and Y. Liu, “Machine learning testing: Survey, landscapes and horizons,” *IEEE Transactions on Software Engineering*, Feb 2020.
6. K. Pei, Y. Cao, J. Yang, and S. Jana, “Deepxplore: Automated whitebox testing of deep learning systems,” *SOSP ’17 Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 1–18, 2017.
7. Y. Tian, K. Pei, S. Jana, and B. Ray, “Deeptest: Automated testing of deep-neural-network-driven autonomous cars,” *ICSE ’18*, pp. 1–11, 2018.
8. F. Harel-Canada, L. Wang, M. Gulzar, Q. Gu, and M. Kim, “Is neuron coverage a meaningful measure of testing deep neural network?” in *2020 ESEC/FSE: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposuium on the Foundations of Software Egnieering*, November 2020, pp. 851–862.
9. 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*, vol. abs/1604.07316, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

10. P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, 2020.
11. M. D. Davis and E. J. Weyuker, "Pseudo-oracles for non-testable programs," in *Proceedings of the ACM '81 Conference*, ser. ACM, 1981, p. 254–257. [Online]. Available: https://doi.org/10.1145/800175.809889
12. T. Y. Chen, "Metamorphic testing: A simple method for alleviating the test oracle problem," in *2015 IEEE/ACM 10th Int. Workshop on Automation of SW Test*, 2015.
13. Z. Q. Zhou and L. Sun, "Metamorphic testing of driverless cars," *Communications of the ACM*, vol. 62, no. 3, pp. 61–67, March 2019.
14. "Udacity self driving car challenge 2 datataset," https://github.com/udacity/self-driving-car/tree/master/datasets/CH2, 2016.
15. "The rambo dnn model for self-driving cars," https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/rambo, 2016.
16. R. L. Ott and M. Longnecker, *An Introduction to Statistical Methods and Data Analysis*. Duxbury Press, 2001.
17. M. Zhang, Y. Zhang, Z. Lingming, C. Liu, and S. Khurshid, "Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems," in *ASE 2018: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, september 2018, pp. 132–142.
18. X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, and Z. Zhao, "Deephunter: a coverage-guided fuzz testing framework for deep neural networks," in *ISSTA 2019: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and*, July 2019.
19. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, October 2020.
20. Y. Sun, X. Huang, and D. Kroening, "Testing deep neural networks," *CoRR*, vol. abs/1803.04792, 2019. [Online]. Available: http://arxiv.org/abs/1803.04792
21. L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, and Y. Wang, "Deepgauge: multi-granularity testing criteria for deep learning systems," in *Proc. of the 33rd ACM/IEEE Int. Conference on Automated Software Engineering*, 2018.