Work-in-Progress: Testing Autonomous Cyber-Physical Systems using Fuzzing Features from Convolutional Neural Networks

Sunny Raj   Sumit Kumar Jha
Computer Science Department
University of Central Florida
Orlando, FL, USA
sraj,jha@eecs.ucf.edu

Arvind Ramanathan   Laura L. Pullum
Computational Science and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN, USA
ramanathana,pullumll@ornl.gov

ABSTRACT

Autonomous cyber-physical systems rely on modern machine learning methods such as deep neural networks to control their interactions with the physical world. Testing of such intelligent cyber-physical systems is a challenge due to the huge state space associated with high-resolution visual sensory inputs. We demonstrate how fuzzing the input using patterns obtained from the convolutional filters of an unrelated convolutional neural network can be used to test computer vision algorithms implemented in intelligent cyber-physical systems. Our method discovers interesting counterexamples to a pedestrian detection algorithm implemented in the popular OpenCV library. Our approach also unearths counterexamples to the correct behavior of an autonomous car similar to NVIDIA's end-to-end self-driving deep neural net running on the Udacity open-source simulator.

1 INTRODUCTION

Safety-critical autonomous cyber-physical systems (CPS) like self-driving cars depend on machine learning algorithms such as deep neural nets for their correct intelligent interactions with the physical world. Testing of such intelligent CPS is challenged by the high-dimensional nature of the input space, such as images and videos. Recent experiences by commercial self-driving pioneers like Uber [5] and Tesla [3] have demonstrated that intelligent CPS may need to be tested more thoroughly to prevent accidents from occurring in response to hitherto unseen visual sensory inputs. In this paper, we investigate the fuzzing-based validation of intelligent cyber-physical systems and present the following results:

- Counterexamples to the Histogram of Oriented Gradients human detection algorithm are readily obtained by fuzzing using *patterns derived from the convolutional filters* learned by a convolutional neural network (CNN). Such an approach is *faster* than fuzzing using random perturbations.
- A self-driving car similar to the NVIDIA end-to-end autonomous car suffers an accident in the Udacity open-source simulator when its input is perturbed using patterns obtained from the convolutional filters learned by an unrelated CNN.

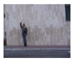
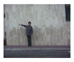
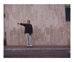
Image	Random Perturbations (seconds)	CNN Perturbations (seconds)	Speedup (Random/CNN)
	74.72	45.24	165%
	855.24	53.13	1610%
	84.00	55.44	151%
	33.47	29.64	113%
	427.68	45.24	945%

Table 1: Perturbations using CNN patterns produce counterexamples faster than random perturbations.

2 RELATED WORK

It has recently been shown that machine learning algorithms are susceptible to adversarial attacks [4]. Deep learning networks can be fooled to classify images differently even when they look the same to the human eye. On the other hand, images that look like noise to humans can be assigned semantically meaningful labels with high confidence by learning algorithms. In related work, it has been shown that perturbation of individual pixels of images can be used to generate adversarial counterexamples. The perturbation of individual bits produces counterexamples but the process is too slow; so, a natural way to attain speedup would be to perturb multiple pixels simultaneously.

We observe that increasing the number of randomly perturbed pixels does not always give us a significant speed up (see Table 1). *Intuitively,* random perturbations do not necessarily lead to the addition of new and interesting features to an input and such random perturbations are filtered away by the machine learning algorithm. However, perturbing the image using patterns obtained from convolutional filters learnt by an unrelated CNN on a different image database speeds up the generation of the counterexample.

3 APPROACH

Our approach for testing intelligent cyber-physical systems using fuzzing consists of the following steps. The user provides an original image or a video, the CPS system under test, and an acceptable Type I error bound [2] on the acceptable error rate from our approach. Our method then picks a pattern obtained from the convolutional filter layer of an unrelated CNN and adds it to a random location in the image or video frame.

EMSOFT'17 Companion, October 15–20, 2017, Seoul, Republic of Korea

If the new image is a counterexample to the vision algorithm, our approach stops. Otherwise, it continues to add more patterns derived from the convolutional filter layer of the CNN. Each image pattern from the CNN is modified by re-normalizing the value of each pixel between 0 and a small constant. We emphasize that the image database used to train the convolutional neural network was unrelated to the image database used in our experimental evaluation. Figure 1 shows 8 patterns obtained from the CNN that were used in our experiments.

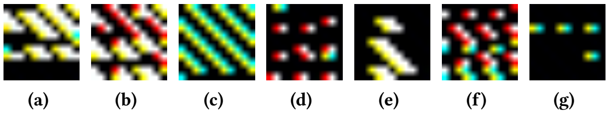


Figure 1: Patterns derived from an unrelated CNN.

Our approach implements Wald's Sequential Probability Ratio Test (SPRT) [2] to decide the number of samples that the algorithm should observe. Our null hypothesis states that the detection algorithm is no more than 99% correct while the alternate hypothesis states that the detection algorithm is at least 99.9999% correct. The algorithm continues to sample until the null hypothesis is rejected with the user-specified bound on the Type I error or an adversarial counterexample is obtained.

4 EXPERIMENTAL RESULTS

We applied our approach to the pedestrian (human) detection algorithm implemented using the Histogram of Oriented Gradients (HOG) approach in the popular OpenCV library. OpenCV implementation is fairly robust to a variety of noise such Gaussian blurring and is able to correctly detect pedestrians even when the blurring effects are strong. This resilience to traditional perturbations makes HOG a good candidate to test our algorithm. On using our approach, OpenCV fails to detect pedestrians or humans in an image despite the human being clearly visible to the human eye.

Table 1 compares the runtime performance of our approach using CNN convolutional filters to an approach that adds random patterns of the same size. It is clear that our approach has a more consistent performance and outperforms the addition of random noise.

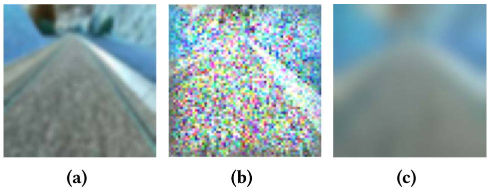


Figure 2: (a) Non-perturbed image. Autonomous driver is robust against (b) salt & pepper noise & (c) Gaussian blurring.

For the second set of experiments, we applied our approach to an autonomous vehicle in the Udacity simulator driven by a deep neural network similar to the NVIDIA end-to-end autonomous driver [1]. The autonomous driver is robust to Gaussian as well as pixelated noise. Figure 2(a) is the non-perturbed original image. Figure 2(b) shows an image perturbed with salt and pepper noise and Figure 2(c) shows strongly perturbed images created using Gaussian blurring. The autonomous driver was successfully able to drive through these perturbations that would arguably prove challenging to an ordinary human driver. However, Figure 3 shows that the autonomous car made steering predictions differing by as much as 10% on visually similar inputs. Eventually, the vehicle crashed because of these perturbations.

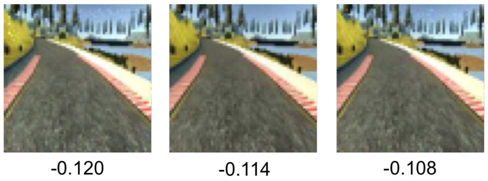


Figure 3: Three nearly-identical images cause a DNN to predict steering angles that vary by as much as 10%.

In our third experiment, we applied our approach to perturb an area of the image that would not impact the performance of a human driver significantly (see Figure 4). To our surprise, the autonomous driver crashed even under this input.

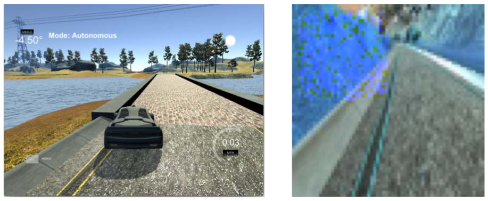


Figure 4: An autonomous driver crashes even if only the upper left corner of the frame is perturbed using our approach.

5 CONCLUSIONS & FUTURE WORK

We have introduced an approach for generating counterexamples to computer vision algorithms using fuzzing patterns derived from the convolutional filter layer of an unrelated CNN. We have successfully applied the approach to a pedestrian detection algorithm implemented in OpenCV and an autonomous car in the Udacity simulator similar to the NVIDIA end-to-end autonomous driver.

Several promising directions for future research remain. Can we use this approach to stress test pedestrian detection systems? Can we exploit the learned patterns to make existing intelligent CPS more robust? Our preliminary results indicate that features learned by convolutional neural networks may be helpful in fuzzing against other unrelated neural nets and vision algorithms.

REFERENCES

[1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).

[2] Bhaskar Kumar Ghosh and Pranab Kumar Sen. 1991. *Handbook of sequential analysis*. CRC Press.

[3] Jordan Golson. 2016. Driver in fatal Tesla Autopilot crash had seven seconds to take action. https://www.theverge.com/2017/1/19/14326604/tesla-autopilot-crash-driver-seven-seconds-inattentive-nhtsa. (2016). Accessed: 2017-05-20.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[5] Ryan Randazzo. 2017. Here's what happened in Uber's self-driving car crash. https://www.usatoday.com/story/news/nation-now/2017/03/30/self-driving-uber-crash-police-report/99814322/. (2017). Accessed: 2017-05-20.