Testing DNN Image Classifiers for Confusion & Bias Errors

Yuchi Tian*
Columbia University
yuchi.tian@columbia.edu

Ziyuan Zhong*
Columbia University
ziyuan.zhong@columbia.edu

Vicente Ordonez
University of Virginia
vicente@virginia.edu

Gail Kaiser
Columbia University
kaiser@cs.columbia.edu

Baishakhi Ray
Columbia University
rayb@cs.columbia.edu

ABSTRACT

We found that many of the reported erroneous cases in popular DNN image classifiers occur because the trained models confuse one class with another or show biases towards some classes over others. Most existing DNN testing techniques focus on per-image violations, so fail to detect class-level confusions or biases. We developed a testing technique to automatically detect class-based confusion and bias errors in DNN-driven image classification software. We evaluated our implementation, DeepInspect, on several popular image classifiers with precision up to 100% (avg. 72.6%) for confusion errors, and up to 84.3% (avg. 66.8%) for bias errors.

KEYWORDS

whitebox testing, deep learning, DNNs, image classifiers, bias

1 INTRODUCTION

Image classification has a plethora of applications in software for safety-critical domains such as self-driving cars, medical diagnosis, *etc.* Even day-to-day consumer software includes image classifiers, such as Google Photo search and Facebook image tagging. Image classification is a well-studied problem in computer vision, where a model is trained to classify an image into single or multiple predefined categories [4]. Deep Neural Networks (DNNs) have enabled major breakthroughs in image classification tasks over the past few years, sometimes even matching human-level accuracy under some conditions [3], which has led to their ubiquity in modern software.

However, in spite of such spectacular success, DNN-based image classification models, like traditional software, are known to have serious bugs. For example, Google faced backlash in 2015 due to a notorious error in its photo-tagging app, which tagged pictures of dark-skinned people as "gorillas" [2]. Analogous to traditional software bugs, the Software Engineering (SE) literature denotes these classification errors as *model bugs* [7], which can arise due to either imperfect model structure or inadequate training data.

At a high-level, these bugs can affect either an *individual image*, where a particular image is mis-classified (*e.g.,* a particular skier is mistaken as a part of a mountain), or an *image class*, where a class of images is more likely to be mis-classified (*e.g.,* dark-skinned people are more likely to be misclassified), as shown in Table 1. The latter bugs are specific to a whole *class* of images rather than individual images, implying systematic bugs rather than the DNN equivalent of off-by-one errors. While much effort from the SE literature on Neural Network testing has focused on identifying individual-level violations—using white-box [5, 6, 9, 14], grey-box [7, 13], or concolic testing [12], detection of class-level violations remains relatively less explored. This paper focuses on automatically detecting such class-level bugs, so they can be fixed.

2 ERROR TYPES AND METHODOLOGY

After manual investigation of some public reports describing the class-level violations listed in Table 1, we determined two root causes: (i) **Confusion**: The model cannot differentiate one class from another. For example, Google Photos confuses skier and mountain [8]. (ii) **Bias**: The model shows disparate outcomes between two related groups. For example, Zhao *et al.* in their paper "Men also like shopping" [16], find classification bias in favor of women on activities like shopping, cooking, washing, *etc.* We further notice that some class-level properties are violated in both kinds of cases. For example, in the case of *confusion errors*, the classification error-rate between the objects of two classes, say, skier and mountain, is significantly higher than the overall classification error rate of the model. Similarly, in the bias scenario reported by Zhao *et al.*, a DNN model should not have different error rates while classifying the gender of a person in the shopping category. Unlike individual image properties, this is a class property affecting all the shopping images with men or women. Any violation of such a property by definition affects the whole class although not necessarily every image in that class, *e.g.,* a man is more prone to be predicted as a woman when he is shopping, even though some individual images of a man shopping may still be predicted correctly. Thus, we need a class-level approach to testing image classifier software for confusion and bias errors.

The bugs in a DNN model occur due to sub-optimal interactions between the model structure and the training data [7]. To capture such interactions, the literature has proposed various metrics primarily based on either neuron activations [5, 6, 9] or feature vectors [7]. However, these techniques are primarily targeted at the individual image level. To detect class-level violations, we abstract away such model-data interactions at the class level and analyze the inter-class interactions using that new abstraction. To this end, we propose a metric using neuron activations and a baseline metric

*Both are first authors, and contributed equally to this research.

ICSE '20 Companion, October 5–11, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7122-3/20/05.
https://doi.org/10.1145/3377812.3390799

Table 1: Examples of real-world bugs reported in neural image classifiers

Bug Type	Name	Report Date	Outcome
Confusion	Gorilla Tag [2]	Jul 1, 2015	Black people were tagged as gorillas by Google photo app.
	Elephant is detected in a room [11]	Aug 9, 2018	Image Transplantation (replacing a sub-region of an image by another image containing a trained object) leads to mis-classification.
	Google Photo [8]	Dec 10, 2018	Google Photo confuses skier and mountain.
Bias	Nikon Camera [10]	Jan 22, 2010	Camera shows bias toward Caucasian faces when detecting people's blinks.
	Men Like Shopping [16]	July 29, 2017	Multi-label object classification models show bias towards women on activities like shopping, cooking, washing, *etc.*
	Gender Shades[1]	2018	Open-source face recognition services provided by IBM, Microsoft, and Face++ have higher error rates on darker-skin females for gender classification.

using weight vectors of the feature embedding to capture the class abstraction.

For a set of test input images, we compute the probability of activation of a neuron per predicted class. Thus, for each class, we create a vector of neuron activations where each vector element corresponds to a neuron activation probability. If the distance between the two vectors for two different classes is too close, compared to other class-vector pairs, that means the DNN under test may not effectively distinguish between those two classes. Motivated by MODE's technique [7], we further create a baseline where each class is represented by the corresponding weight vector of the last linear layer of the model under test.

3 EVALUATION AND CONTRIBUTIONS

We evaluate our methodology for both single- and multi-label classification models in eight different settings. Our experiments demonstrate that DeepInspect can efficiently detect both Bias and Confusion errors in popular neural image classifiers. We further check whether DeepInspect can detect such classification errors in state-of-the-art models designed to be robust against norm-bounded adversarial attacks [15]; DeepInspect finds hundreds of errors proving the need for orthogonal testing strategies to detect such class-level mispredictions. Unlike some other DNN testing techniques [9, 12, 13], DeepInspect does not need to generate additional transformed (synthetic) images to find these errors. The primary contributions of this paper are:

- We propose a novel neuron-coverage metric to automatically detect class-level violations (confusion and bias errors) in DNN-based visual recognition models for image classification.
- We implemented our metric and underlying techniques in DeepInspect.
- We evaluated DeepInspect and found many errors in widely-used DNN models with precision up to 100% (avg. 72.6%) for confusion errors and up to 84.3% (avg. 66.8%) for bias errors.

Our code is available at https://github.com/ARiSE-Lab/DeepInspect. The errors reported by DeepInspect are available at: https://www.ariselab.info/deepinspect.

ACKNOWLEDGMENTS

This work is supported in part by NSF CNS-1563555, CCF-1815494, CNS-1842456, CCF-1845893, and CCF-1822965. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the US Government or NSF.

REFERENCES

[1] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *FAT*.
[2] Loren Grush. 2015. Google engineer apologizes after Photos app tags two black people as gorillas. (2015). https://www.theverge.com/2015/7/1/8880363/google-apologizes-photos-app-tags-two-black-people-gorillas
[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[4] Pooja Kamavisdar, Sonam Saluja, and Sonu Agrawal. 2013. A survey on image classification approaches and techniques. *International Journal of Advanced Research in Computer and Communication Engineering* 2, 1 (2013), 1005–1009.
[5] Jinhan Kim, Robert Feldt, and Shin Yoo. 2019. Guiding deep learning system testing using surprise adequacy. In *Proceedings of the 41st International Conference on Software Engineering*. IEEE Press, 1039–1049.
[6] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. 2018. DeepGauge: Multi-granularity Testing Criteria for Deep Learning Systems. (2018), 120–131. https://doi.org/10.1145/3238147.3238202
[7] Shiqing Ma, Yingqi Liu, Wen-Chuan Lee, Xiangyu Zhang, and Ananth Grama. 2018. MODE: Automated Neural Network Model Debugging via State Differential Analysis and Input Selection. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Lake Buena Vista, FL, USA) *(ESEC/FSE 2018)*. ACM, New York, NY, USA, 175–186. https://doi.org/10.1145/3236024.3236082
[8] MalletsDarker. 2018. I took a few shots at Lake Louise today and Google offered me this panorama. (2018). https://www.reddit.com/r/funny/comments/7r9ptc/i_took_a_few_shots_at_lake_louise_today_and/dsvv1nw/
[9] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. (2017), 1–18. https://doi.org/10.1145/3132747.3132785
[10] Adam Rose. 2010. Are Face-Detection Cameras Racist? (2010). http://content.time.com/time/business/article/0,8599,1954643,00.html
[11] Amir Rosenfeld, Richard S. Zemel, and John K. Tsotsos. 2018. The Elephant in the Room. *CoRR* abs/1808.03305 (2018). arXiv:1808.03305 http://arxiv.org/abs/1808.03305
[12] Youcheng Sun, Min Wu, Wenjie Ruan, Xiaowei Huang, Marta Kwiatkowska, and Daniel Kroening. 2018. Concolic Testing for Deep Neural Networks. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) *(ASE 2018)*. ACM, New York, NY, USA, 109–119.
[13] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. DeepTest: Automated testing of deep-neural-network-driven autonomous cars. In *International Conference of Software Engineering (ICSE), 2018 IEEE conference on*. IEEE.
[14] Jingyi Wang, Guoliang Dong, Jun Sun, Xinyu Wang, and Peixin Zhang. 2019. Adversarial sample detection for deep neural network through model mutation testing. In *Proceedings of the 41st International Conference on Software Engineering*. IEEE Press, 1245–1256.
[15] Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J. Zico Kolter. 2018. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 8410–8419. http://papers.nips.cc/paper/8060-scaling-provable-adversarial-defenses.pdf
[16] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2941–2951.