Detecting adversarial examples for deep neural networks via layer directed discriminative noise injection

Wang, Si; Liu, Wenye; Chang, Chip-Hong

2019

Wang, S., Liu, W., & Chang, C.-H. (2019). Detecting adversarial examples for deep neural networks via layer directed discriminative noise injection. 2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). doi:10.1109/AsianHOST47458.2019.9006702

https://hdl.handle.net/10356/137128

https://doi.org/10.1109/AsianHOST47458.2019.9006702

Detecting Adversarial Examples for Deep Neural Networks via Layer Directed Discriminative Noise Injection

Si Wang, Wenye Liu and Chip-Hong Chang
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email: si003@e.ntu.edu.sg, wliu015@e.ntu.edu.sg, echchang@ntu.edu.sg

***Abstract*—Deep learning is a popular powerful machine learning solution to the computer vision tasks. The most criticized vulnerability of deep learning is its poor tolerance towards adversarial images obtained by deliberately adding imperceptibly small perturbations to the clean inputs. Such negatives can delude a classifier into wrong decision making. Previous defensive techniques mostly focused on refining the models or input transformation. They are either implemented only with small datasets or shown to have limited success. Furthermore, they are rarely scrutinized from the hardware perspective despite Artificial Intelligence (AI) on a chip is a roadmap for embedded intelligence everywhere. In this paper we propose a new discriminative noise injection strategy to adaptively select a few dominant layers and progressively discriminate adversarial from benign inputs. This is made possible by evaluating the differences in label change rate from both adversarial and natural images by injecting different amount of noise into the weights of individual layers in the model. The approach is evaluated on the ImageNet Dataset with 8-bit truncated models for the state-of-the-art DNN architectures. The results show a high detection rate of up to 88.00% with only approximately 5% of false positive rate for MobileNet. Both detection rate and false positive rate have been improved well above existing advanced defenses against the most practical non-invasive universal perturbation attack on deep learning based AI chip.**

I. Introduction

Since early 2010s, Deep Learning, also referred as Deep Neural Network (DNN), has undergone a blossoming in many applications [1]. The superiority in highly end-to-end problem-solving behavior and embedded feature extraction functionality has made it a preferable option for handling majority of the challenging tasks in image classification [2], speech recognition and language processing [3]. With the progressive advancements of models and easy access to required hardware resources for training, deep learning is rapidly growing to maturity and stepping into safety and security fields, such as self-driving cars, malware detection, and surveillance [4].

While overwhelmed by the impressive performance of DNNs, an inherent vulnerability was shown to be exploitable for corrupting deep learning based image classification systems. Szegedy et al. [5] first found that the classifier is susceptible to adversarial samples generated by purposefully adding imperceptibly small perturbations to the natural inputs. Even with high generalization ability, the model could rarely refrain from misclassification in such an adversarial environment. The intriguing findings have triggered a number of studies on efficient algorithms for adversarial image generation [6]–[9]. Apart from image-specific distortions, Moosavi-Dezfooli et al. [10] have proposed universal perturbations that enable class altering on any image.

The increasing interest in deploying DNNs in safety and security critical fields has motivated researchers to seek solutions to tackle the above security breaches. Existing defend strategies can be dichotomized into robustness enhancement and adversaries detection. The former approach aims to reduce the model sensitivity to malicious changes in inputs. Related works could be broadly grouped into three categories: adversarial training, gradient masking, and input transformation. Adversarial training augments the original dataset with discovered adversarial examples for training. It has later shown to be non-adaptive to new attacks [6]. Gradient masking seeks to conceal gradient information by producing near-zero gradients, but found to be feeble in mitigating black-box attack because of the transferability of adversarial samples [11], [12]. Input transformation attempts to eliminate the effect of adversarial perturbations by introducing variations to the inputs. Previous detection techniques [13]–[15] could also be broadly categorized into three classes: sample statistics, detector training and prediction inconsistency. Sample statistics require large-scale datasets for analysis and have shown to be less effective in filtering out the adversarial samples located near the legitimate distribution. Training a detector adds additional subnetworks that are dedicated for distinguishing adversarial inputs from the clean ones. Prediction inconsistency measures the degree of consensus among multiple models in predictions.

Most discussed defensive mechanisms are either implemented with small datasets, such as MNIST and CIFAR-10, or shown to have limited success. Furthermore, they are analyzed at software level without considering the attack feasibility and defence effectiveness when they are performed in the trained DNN model implemented on hardware platforms. In this paper, we proposed a detection strategy directed by the high sensitivity of few individual layers to weight perturbation by noise injection. The discriminative noise injection adversaries detection method is implemented on three state-of-the-art 8-bit truncated Convolutional Neural Network (CNN) models targeting ImageNet classification. The approach falls into the

TABLE I: Three ImageNet models under evaluation

Models	Year	# layers	# parameters	Top-1/5 accuracies
VGG 16 [16]	2014	16	138 million	71.5% / 89.8% [17]
VGG 19 [16]	2014	19	144 million	71.1% / 89.8% [17]
MobileNet [18]	2017	28	4.3 million	70.9% / 89.9% [17]

third category of detection methods: prediction inconsistency. It is driven by the observations that sensitivity differences between adversarial and benign samples vary across layers when random noises are injected to their respective weight parameters. This suggests the existence of biasing in layers towards adversarial discriminability. Instead of an even perturbation across all layers, it is sufficient to subject a few dominant candidate layers to non-uniform weight distortions to maximize the discriminability of adversaries while retaining the model accuracy on normal samples. Multiple single-layer noise injections are experimented to progressively tune the adaptive detection thresholds of each selected dominant layer.

The rest of this paper is organized as follows. Section II provides a brief introduction of ImageNet dataset, the three CNN architectures: VGG 16, VGG 19 and MobileNet, and the universal perturbation method that is applied for crafting adversarial samples. Section III describes the proposed discriminative noise injection method. Section IV presents and discusses the evaluation results. Section V concludes the paper.

II. Preliminaries

A. ImageNet

ImageNet is a dataset of color images with high resolution, provided by the ImageNet Large Scale Visual Recognition Challenge 2012 for image classification tasks. It contains 1.2 million training images and 50,000 images for validation, all of which are human-annotated with 1000 categories. Each image is represented using 24-bit color with separate red, green and blue color channels. Each channel is 8-bit scale with an integer value ranging from 0 to 255.

B. Deep Neural Network Architectures

In this work, three deep image classification models that are implemented with the proposed detection techniques are listed below. Their basic attributes are summarized in Table I.

VGGNet: Both VGG 16 and VGG 19 belong to the VGGNet [16] family. They possess similar architectures as AlexNet [19], but with much more convolutional layers. Furthermore, VGGNets use 3×3 kernel for all the convolutional layers.

MobileNet: MobileNet [18] is a class of efficient deep learning models dedicated for mobile and embedded vision applications with constrained resources. They are built based on depth-wise separable convolutions that factorize a standard convolution into a depth-wise convolution and a point-wise convolution. The former uses a single filter to convolve with each input channel. The results are then combined by the latter using a 1×1 convolution. In this study, MobileNet-v1 is used.

C. Universal Perturbation

Moosavi-Dezfooli et al. [9] extended the image-dependent DeepFool method, which is an untargeted attack that utilizes concepts from geometry to seek the minimum perturbation required for altering the classification results. A generalized image-agnostic adversarial perturbation can be mathematically expressed as follows [10]:

$$v := \left\{ v | Err(X_v) := \frac{1}{m} \sum_{i=1}^{m} 1_{\widehat{k}(x_i+v) \neq \widehat{k}(x_i)} \geq 1 - \delta \right\},$$

$$s.t. \left\| v \right\|_p \leq \xi. \quad (1)$$

where X represents a set of natural images $\{x_1, ..., x_m\}$ collected from a population, $\widehat{k}$ denotes a classifier that predicts an approximate label $\widehat{k}(x)$ for each input x, v is a fixed perturbation vector that leads to the misclassification of most samples in X, X_v defines a set of perturbed data points $\{x_1 + v, ..., x_m + v\}$, δ quantifies the accuracy of the tampered samples, and ξ constrains the extent of distortion v.

The algorithm iteratively feeds the benign inputs with addition of previously derived universal perturbation into the classifier. It calculates the minimum distortion that sends the current tampered input to the decision boundary in each iteration. The instance of universal aberration is updated by adding the computed distortion and then used for the next round.

III. Proposed Discriminative Noise Injection Method

A. Threat Model

The proposed method aims to defend against a powerful adversary who has full knowledge of a custom integrated DNN classifier deployed in the field, but has no accessibility to the inner system. The adversary may attempt to generate malicious images to fool the system using white-box attack techniques. The adversary cannot physically alter the internal nodes and nets of a monolithic integrated circuit. Neither can he determine the perturbed layers and extent of perturbations introduced to the classifier by reverse engineering the circuit without leaving apparent evidence of physical tampering. Attacks, such as Fast Gradient Sign Method (FGSM) [6], C&W attack [7] and DeepFool [9], inflict infinitesimal changes for manipulation, while effective on DSP algorithm and software, are infeasible in this scenario. As the magnitude of the derived adversarial perturbations are mostly less than 1, such distortions will be eliminated upon converting into the 8-bit color representation, resulting in extremely large failure rate of these attacks. Furthermore, the search space of an ImageNet image is enormous for Jacobian-based saliency map approach (JSMA) [8], which can easily render the system to run out of memory. For this reason, the most feasible and effective non-invasive attack of universal perturbation is considered in this work.

B. Hypothesis

The proposed discriminative noise injection method is inspired by the recent work [20] that discovered distinguishable sensitivity between normal and adversarial samples when random mutations are applied on the DNN model. It is founded on the observation of the disparate degree of distinguishable sensitivity of each layer to noise perturbation. Our hypothesis is that there exists multiple dominant layers that affect the tampered samples most under varying extent of layer mutations. To evaluate the hypothesis, noises are generated to randomly selected weight parameters for every layer of the three DNN architectures. Each layer is mutated 10 times with the proportion of noisy parameters varies from 0.1% to 1%. Label change rate (LCR) is used to quantify the sensitivity of samples on mutated models. It is defined as [20]:

$$\zeta(x) = \frac{|\{f_i | f_i \in F \cap f_i(x) \neq f(x)\}|}{|F|}. \quad (2)$$

where x denotes an input image, f is the original model, F is a set of mutated models $\{f_1, ..., f_m\}$ and $|F|$ is the number of elements of F.

Table II summarizes the experimental results of the measured $\zeta(x)$ from ImageNet dataset. 300 randomly selected natural images and 300 adversarial images crafted using universal perturbation method are tested for each model. The noise injection rate here refers to the proportion of abused weights in respective layers under test. Since the number of layers is different across DNN architectures, the number of generated distorted models will also be different in calculating the average LCR. It can be clearly seen from Table II that ζ_{adv} for adversarial inputs is always much larger than ζ_{nor} for normal inputs at any noise injection rate for any CNN model. This further proves that the hypothesis proposed by [20] also works for ImageNet dataset, apart from MNIST and CIFAR-10.

The LCR difference $\Delta\zeta$, computed by $(\zeta_{adv} - \zeta_{nor})$, on various layers are illustrated in Fig. 1 to Fig. 3. From Figs. 1 and 2, it is observed that $\Delta\zeta$ of convolutional layers $\Delta\zeta_{conv}$ is generally larger than $\Delta\zeta$ of fully connected layers $\Delta\zeta_{fc}$ for VGG 16 and VGG 19. This implies the stronger differentiability possessed by convolutional layers with a relatively small noise injection rate for models with no convolution factorization. The observation is also explainable since the feature extraction process is mostly done in the convolutional layers. A few changes in convolutional weights are likely to dramatically decrease the influence of adversarial perturbations. Besides, the two figures indicate a likelihood for the existence of an optimal noise injection rate for each layer. For example, $\Delta\zeta_{conv3\_1}$ increases when noise injection rate is less than 0.004, but decreases thereafter in VGG 16. Thus, the optimal noise injection rate to enlarge the difference between normal and adversarial samples is estimated to be 0.004 for $conv3\_1$ layer. The reduced distinguishability after the optimal points indicates increased mutation effect on normal samples. $\Delta\zeta$ of fully connected layers for these two CNN architectures seem to be always monotonically increasing. However, the

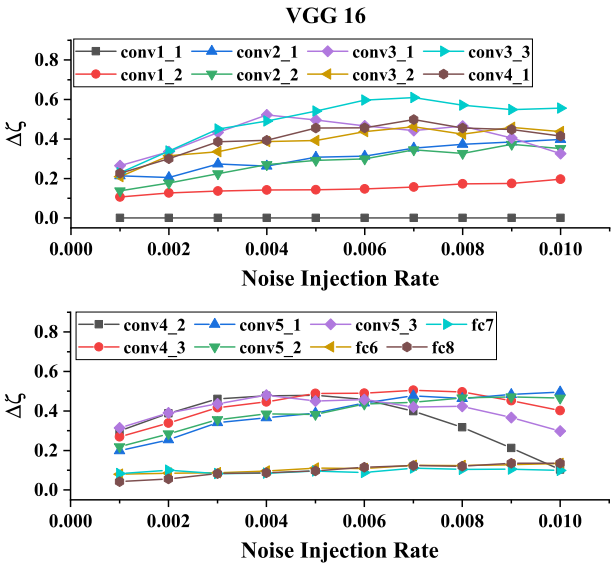

Fig. 1: Comparison of $\Delta\zeta$ in different layers of VGG 16.

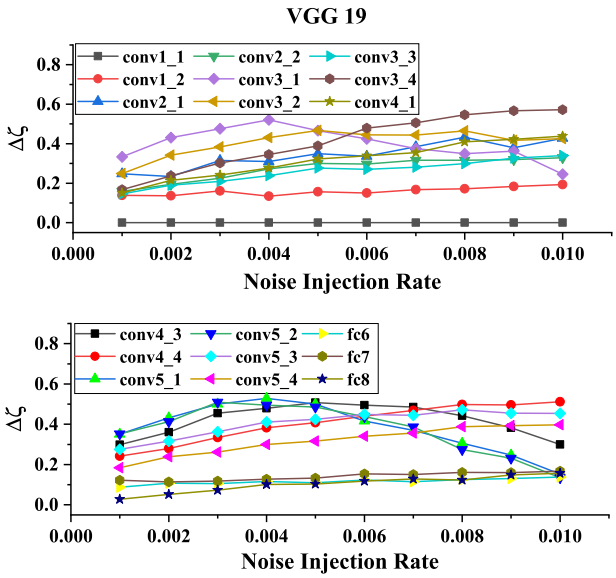

Fig. 2: Comparison of $\Delta\zeta$ in different layers of VGG 19.

lack of convexity might be due to the early halt of noise injection rate.

For MobileNet, $\Delta\zeta$ of some layers are always 0 with noise injection rate ranging from 0.001 to 0.01. They are not shown in Fig. 3 to avoid cluttering the graph. The results in Fig. 3 show that $\Delta\zeta$ for $conv5^*$ and $conv6^*$ layers decrease nearly monotonically in [0.001, 0.01], whereas most $\Delta\zeta$ for $conv2^*$, $conv3^*$ and $conv4^*$ layers rise rapidly from 0 in a short interval of noise injection rate and then fluctuate around certain value. The trend for fully connected layer is similar to those of VGG 16 and VGG 19. One possible interpretation of the unusual behavior of the $conv5^*$ and $conv6^*$ layers is that their optimal noise injection rates are smaller than 0.001. This

TABLE II: Average ζ for normal samples and adversarial samples (unit:%).

Models	Type of samples	Noise injection rate									
		0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.01
VGG 16	Normal	1.92	3.08	4.60	6.23	8.85	11.32	14.38	16.84	20.53	24.69
	Adversarial	19.98	26.17	32.75	36.79	40.87	44.51	48.57	49.99	52.71	54.78
VGG 19	Normal	1.36	2.35	4.09	6.02	9.06	12.24	15.76	19.56	22.89	27.49
	Adversarial	21.36	26.82	32.94	37.41	41.97	45.07	48.15	51.44	53.35	56.06
MobileNet	Normal	6.23	7.91	12.64	17.98	21.41	23.36	25.92	26.82	29.42	30.15
	Adversarial	18.08	23.29	26.02	32.48	34.04	35.29	36.15	38.47	39.36	39.92

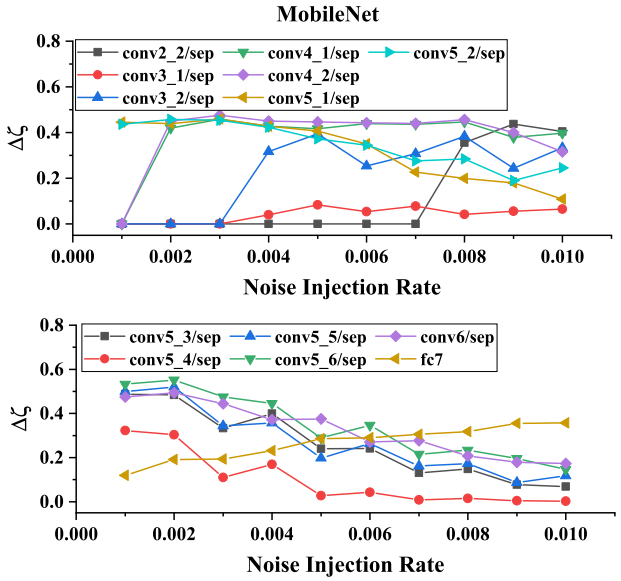

Fig. 3: Comparison of $\Delta\zeta$ of different layers in MobileNet.

implies the high sensitivity of benign images to manipulation of $conv5^*$ and $conv6^*$ layers in MobileNet.

C. Detection Framework

The proposed dominant layer directed approach takes advantage of few layers that have high distinguishability for adaptive thresholding according to the input susceptibility to label change by random noise injection to the weights of these layers successively. The detailed detection framework is demonstrated in Fig. 4. $L_{i,j}$-$\mu_{i,j}$ denotes the j-th mutated model that is imposed with μ_i perturbation to the weight parameters of layer L_i and LCR denotes the label change rate ζ. For a given input, m mutated models are generated by changing μ_i ratio of weight parameters in layer L_i to calculate LCR. If LCR is greater than a predetermined threshold $tr_{adv,i}$, the input is very likely to be deliberately distorted; If LCR is smaller than $tr_{nor,i}$, the input is considered to be legitimate; If $LCR \in (tr_{nor,i}, tr_{adv,i})$, another m mutated models will be produced with μ_{i+1} perturbation to the weights of layer L_{i+1} for further assessment. This process goes on until the window between normal threshold tr_{nor} and adversarial threshold tr_{adv} for layer L_n with μ_n rate of noise injection is closing in to a single threshold tr. n phases are shown in Fig. 4.

The use of a hierarchical instead of single-layer mutation in detecting adversaries is mainly due to the possibility that malicious perturbation may destroy manifolds of important features. Therefore, single-layer distortion may not capture all or majority of the differences. Fig. 5 depicts the histogram distribution of LCR for 300 natural samples and 300 adversarial samples with 10 mutations on layer $conv3\_3$ of VGG 16. Each mutation is completed by injecting noises to 0.006 proportion of weights randomly in $conv3\_3$. $conv3\_3$-0.006 is one of the mutating ways that have notable $\Delta\zeta$. Nevertheless, a large overlapping area still exists between the distributions of the tampered and benign inputs. This observation agrees with our conjecture.

IV. Implementation Results and Discussions

The proposed layer discriminative noise injection strategy is implemented using 8-bit truncated Caffe models for VGG 16, VGG 19 and MobileNet with MatLab interface in a PC equipped with an E5-1630 v4 3.70GHz CPU, 16GB system memory, and a GeForce GTX 1070Ti. In the following evaluations, efficiency of the approach is assessed through multiple experiments.

A. Experimental Setup

Dataset: 600 legitimate examples and 600 successfully crafted adversarial examples are collected for each network architecture. The dataset is then randomly separated into two groups, each of which contains 300 legitimate examples and 300 adversarial examples for each network architecture. One group is used for figuring out the layer compositions, respective noise injection rates as well as threshold values in each phase of the detection scheme. The other group is used for validation.

Hierarchy determination: In this stage, a dominant layer that can maximize the discriminability of adversaries within a limited range of noise injection rate is selected for each detection phase. The respective threshold values are also determined. For the early stages, our practice is to take a mutated layer that has the largest $\Delta\zeta$ with the constraints that no more than 5 legitimate samples and 10 adversarial samples are misclassified with the selected tr_{adv} and tr_{nor} for each stage. The first constraint imposed on the selection is to restrict the false positive rate (FPR) to approximately 5% for 3 to 5 phases of detection. This is because a detector with high false positive rate is undesirable in security critical applications. The second constraint is to prevent the degradation of correct

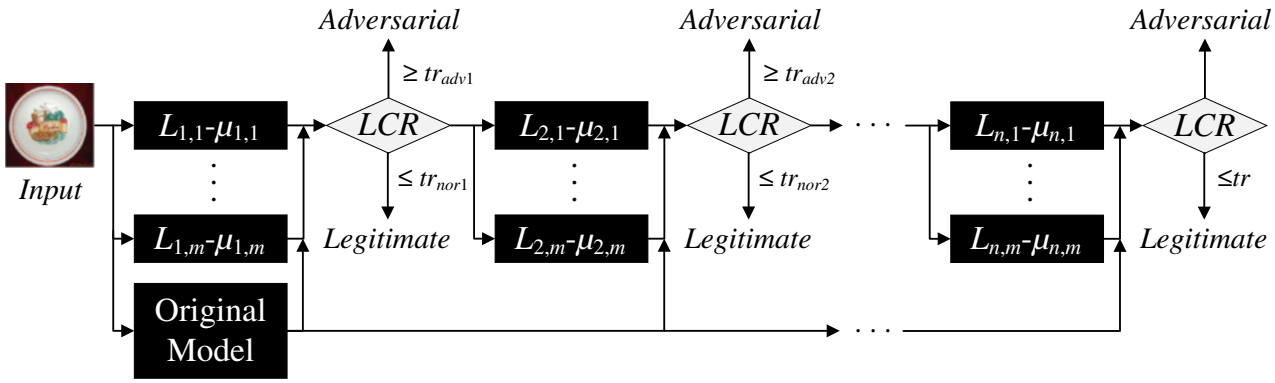

Fig. 4: Framework of proposed dominant layer directed discriminative noise injection approach.

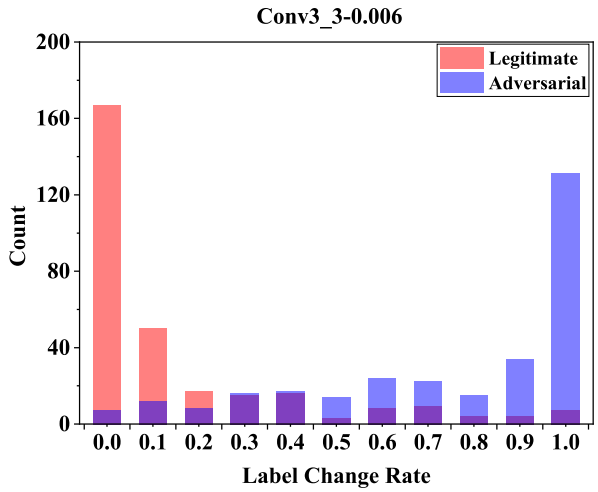

Fig. 5: Histogram distribution of LCR for legitimate and adversarial samples with 0.006 noise injection rate in layer $conv3\_3$ of VGG 16.

TABLE III: Final configurations for the three models with hierarchical detection scheme.

Models	Mutation rate	Layer	$(tr_{nor}, tr_{adv}) \vee tr$
VGG 16	0.006	$conv3\_3$	(0, 1)
	0.008	$conv5\_1$	(0, 0.8)
	0.006	$conv5\_1$	0.4
VGG 19	0.004	$conv5\_2$	(0.1, 1)
	0.004	$conv5\_1$	(0, 1)
	0.004	$conv3\_1$	(0, 1)
	0.009	$conv4\_1$	$(-0.1, 0.4)$
	0.005	$conv4\_3$	0.7
MobileNet	0.002	$conv5\_6/sep$	(0.4, 0.9)
	0.001	$conv5\_6/sep$	(0.3, 0.8)
	0.002	$conv5\_5/sep$	(0.4, 0.9)
	0.003	$conv5\_6/sep$	0.8

TABLE IV: Final configurations for the three models with single-layer detection scheme.

Models	Mutation rate	Mutated layer	Threshold
VGG 16	0.006	$conv3\_3$	0.7
VGG 19	0.008	$conv3\_4$	0.5
MobileNet	0.001	$conv5\_6/sep$	0.6

detection (or true positive) rate. The final configurations of the hierarchical detection scheme for each CNN model are shown in Table III. The configurations for detection using single-layer mutation are also listed in Table IV.

B. Results

Table V shows the validation outcomes for the proposed method with both predefined single-layer and multi-layer configurations. TPR and FPR are acronyms for true positive rate and false positive rate, respectively. They are defined as:

$$TPR = TP/(TP + FN) \tag{3}$$

$$FPR = FP/(FP + TN) \tag{4}$$

where TP, TN, FP and FN represent true positives, true negatives, false positives and false negatives, respectively.

In our case, an 'adversary' is considered to be a positive class. It can be seen that all the multi-layer configurations have much higher detection rates than single-layer configurations without much sacrifice of FPR. The outstanding performance of joint-mutation is likely due to the richer feature information included in perturbations than the limited feature change information detectable by perturbations of single-layer mutation. Moreover, the FPR for all the scenarios are around an acceptable 5% rate, which is attributed to the loss control in the setup stage. In addition, MobileNet ranks first in detection rate regardless of configurations, revealing the poor robustness of adversaries to the mutants of MobileNet. Correspondingly, a high sensitivity of decision boundary on noise injection into weights of MobileNet could also be inferred.

The results are also compared with feature squeezing method [13] in Table VI. It can be observed that the proposed method, although implemented in an 8-bit truncated version of MobileNet, has a better performance than feature squeezing method. However, the detection rate and FPR of

TABLE V: Evaluation results for different discriminative noise injection configurations.

Models	Configuration	Detection rate (TPR) (%)	FPR (%)
VGG 16	single-layer	60.00	5.00
	3-layer	82.00	5.67
VGG 19	single-layer	60.00	5.00
	5-layer	80.33	7.00
MobileNet	single-layer	75.33	4.33
	4-layer	88.00	5.67

TABLE VI: Comparison with feature squeezing method.

Metrics	This work	Feature squeezing [13]
Model	MobileNet	MobileNet
Truncated model	Yes	No
Dataset	ImageNet	ImageNet
Configuration	4-layer	3-squeezer
Detection rate (%)	88.00	85.94
FPR (%)	5.67	8.33

feature squeezing method shown in Table VI are average values over 9 attacking techniques, each of which feeds 100 adversarial and natural images, while our results are only based on universal perturbation attack with 300 adversarial and natural images. Therefore, for a fair comparison, defensive efficiency against DeepFool attack using feature squeezing method is also performed since universal perturbation attack is originated from DeepFool attack. The detection rates for DeepFool are 78.60% using joint squeezers and 71.40% using a single squeezer. The proposed layer discriminative noise injection approach has evidently higher detection rate for both configurations. Particularly for multi-layer design, it is nearly 10% better than joint feature squeezers.

V. Conclusions

A dominant layer directed discriminative noise injection method is proposed for adversarial perturbation detection in this paper. It harnesses the greater separability of adversarial and legitimate images by utilizing their difference in sensitivity under respective layer-wise parameter contamination. The optimal configuration is obtained by pre-characterization to select a small number of sensitive layers to be mutated, their respective noise injection rates and label change rate thresholds for each of the three models. The detection efficiency is maximized with as small mutation rate and number of operating rounds as possible. 8-bit truncated version of the three models are used for evaluation to account for typical fixed-point implementation of hardware DNN. The results show a strong competitiveness with improved detection efficiency compared with the state-of-the-art defenses that target software DNN classifiers. Light-weight linear-feedback shift registers (LFSRs) can be embedded to the DNN hardware accelerator for the random noise injection required by the proposed method.

Acknowledgement

This research is supported by Singapore Ministry of Education AcRF Tier 2 Grant No. MOE-2015-T2-2-013.

References

[1] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[2] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[3] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[4] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. SAN JOSE, CA: IEEE, May 2017, Conference Proceedings, pp. 39–57.

[8] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. Saarbrcken, GERMANY: IEEE, Mar. 2016, Conference Proceedings, pp. 372–387.

[9] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Nevada., Jun. 2016, Conference Proceedings, pp. 2574–2582.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, Hawaii, Jul. 2017, pp. 1765–1773.

[11] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. SAN JOSE, CA: IEEE, May 2016, Conference Proceedings, pp. 582–597.

[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. Abu Dhabi, United Arab Emirates: ACM, 2017 Apr., Conference Proceedings, pp. 506–519.

[13] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[14] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[15] J. Wang, J. Sun, P. Zhang, and X. Wang, "Detecting adversarial samples for deep neural networks through mutation testing," *arXiv preprint arXiv:1805.05010*, 2018.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sep. 2018, pp. 631–648.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, Harrahs and Harveys, Lake Tahoe, Dec. 2012, pp. 1097–1105.

[20] J. Wang, G. Dong, J. Sun, X. Wang, and P. Zhang, "Adversarial sample detection for deep neural network through model mutation testing," in *Proceedings of the 41st International Conference on Software Engineering*. QC,Canada: IEEE Press, May 2019, pp. 1245–1256.