Defending Against Adversarial Attacks in Deep Learning with Robust Auxiliary Classifiers Utilizing Bit Plane Slicing

Yuan Liu* †‡, Pingqiang Zhou*
* School of Information Science and Technology, ShanghaiTech University, Shanghai, China
† Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China
‡ University of Chinese Academy of Sciences, Beijing, China
liuyuan1@shanghaitech.edu.cn, zhoupq@shanghaitech.edu.cn

2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST) | 978-1-7281-8952-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/ASIANHOST51057.2020.9358268

***Abstract*—Deep Neural Networks (DNNs) have been widely used in variety of fields with great success. However, recent researches indicate that DNNs are susceptible to adversarial attacks, which can easily fool the well-trained DNNs without being detected by human eyes. In this paper, we propose to combine the target DNN model with robust bit plane classifiers to defend against adversarial attacks. It comes from our finding that successful attacks generate imperceptible perturbations, which mainly affects the low-order bits of pixel value in clean images. Hence, using bit planes instead of traditional RGB channels for convolution can effectively reduce channel modification rate. We conduct experiments on dataset CIFAR-10 and GTSRB. The results show that our defense method can effectively increase the model accuracy on average from 8.72% to 85.99% under attacks on CIFAR-10 without sacrificing accuracy of clean images.**

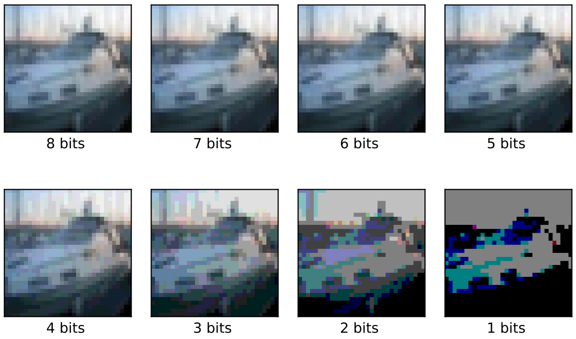
***Index Terms*—adversarial defense, security of neural networks, bit plane slicing**

I. Introduction

Deep learning has been widely applied in image recognition [1], speech processing [2] and automatic driving [3], etc. However, most designers seldom consider the security problem. Recent research [4] shows that deep learning models are vulnerable to adversarial examples which are crafted by adding small perturbations to clean images, also known as adversarial attack. Adversarial examples can induce the classifier to make wrong predictions while they can still be identified correctly by human eyes. Imagining an automatic vehicle recognizes a stop sign as a turn left sign, then it will cause serious traffic accidents which incur huge costs. Therefore, it is essential to defend against adversarial attacks.

In this paper, we build robust bit plane classifiers combined with target model to mitigate the adversarial attack. Bit plane slicing [5] decomposes each R/G/B channel into 8 bit planes. Fig. 1 shows images constructed from different number of bit planes. We can see that high-order bit planes contain a large amount of image data while the low-order bit planes contribute the details in the image. Besides, by analyzing perturbation rate of each bit plane, we find that perturbations are mainly located in the low-order bits of pixel value, which motivates us to extract the unaffected high-order bit plane for classification.

978-1-7281-8952-9/20/$31.00 ©2020 IEEE



8 bits 7 bits 6 bits 5 bits

4 bits 3 bits 2 bits 1 bits

Fig. 1: Image reconstruction from bit planes. The number means how many top bit planes are used from the highest order.

Based on the findings mentioned above, we train several bit plane classifiers combined with target model as the whole classification system shown in Fig. 2. We will give more details about it in Section II-C.

Our contributions are as follows:

- We propose to utilize bit plane slicing to transform the input images. We find the distribution of perturbations by analyzing perturbation rate of each bit plane, which motivates us to choose the unaffected high-order bit plane as input for classification.
- We propose a new defense architecture combining target model with different bit plane classifiers. Bit plane classifiers are more robust than RGB channel based classifiers and are more difficult to be attacked.
- We conduct experiments on two RGB datasets CIFAR-10 [6] and GTSRB [7] with three famous attack methods Fast Gradient Sign Method [8], Carlini and Wagner [9], DeepFool [10]. Results show that our method can improve the accuracy efficiently under the three attacks.

The remainder of this paper is organized as follows: In Section II, we introduce threat model, bit plane slicing and the overall defense architecture. In Section III, we talk about experiments and give a conclusion in Section IV.

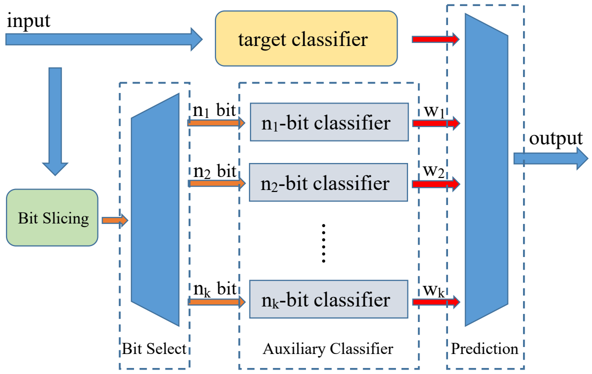


Fig. 2: The overall architecture for defense.

II. Proposed Method

In this section, we introduce threat model, definition of bit plane slicing and the whole defense architecture.

A. Threat Model

We suppose attackers know everything about the target model including the architecture, training data, optimization method and all hyper-parameters. The attackers can use all such information to generate adversarial examples to fool the network. However, attackers have no information about the defensive policy. On the side of defenders, they know the training data, and they can give inputs to model and get outputs. However, they can not change the inner parameters or any state of the model. Besides, they have no information about how the adversarial examples are generated.

B. Bit Plane Slicing

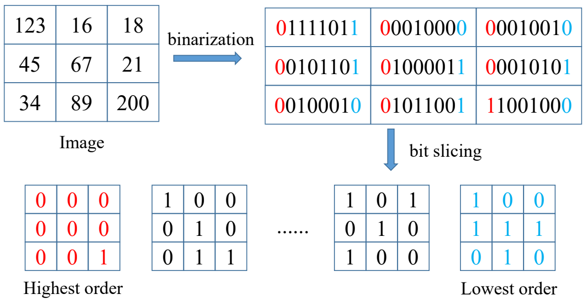


Fig. 3: Process of bit plane slicing.

Images consist of pixels and each pixel contains R/G/B channels. The value range of each channel is from 0 to 255. Bit plane slicing [5] transforms each pixel value from integers to binary representation. Therefore, a gray image can be considered as 8 one-bit planes and an RGB image can be considered as 24 one-bit planes. Reconstructions of each pixel are done by multiplying 2^{n-1} with bit values of the n-th plane and then adding all resulting planes. Obviously, the high-order bits contribute more in pixel value than low-order bit planes. To make it clear, Fig. 3 shows the transformation process from a gray image to 8 one-bit planes. Given a 3x3 gray image, each pixel value can be transformed to its binary form with 8 bit and the same order bit can be taken out to form a new bit plane.

Based on the finding that most adversarial perturbations are located in the low-order bits of pixel value shown in Fig. 5. We only take the unaffected high-order bit plane as input feature for classification, then the bit plane classifiers taking high-order bit plane as input can greatly filter the small perturbations in adversarial examples without changing the semantic content of images.

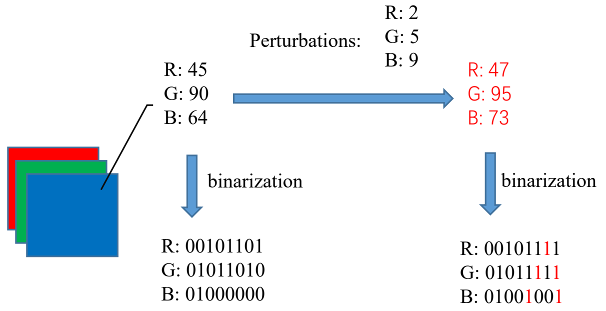


Fig. 4: Channel modification rate of RGB and bit plane.

Another important point is that using bit planes as input features instead of original RGB channels can decrease the channel modification rate. Here we define the channel modification rate as the number of modified channels after perturbations divided by the total channels. Take Fig. 4 as an example, when perturbations are added to the RGB pixel value, the total RGB channels are modified even with small magnitude, which leads to a channel modification rate 3/3. However, when pixel value is transformed to binary form, few bit channels are modified which leads to a channel modification rate 5/24. If we only use the top 3-bit, 4-bit and 5-bit from the highest order planes as input features, the modification rate reduces to 0/9, 0/12 and 1/15 respectively. With fewer high-order bit planes used, the channel modification rate can get smaller and finally reach to zero. When channel modification rate is zero, adversarial examples and clean images are the same from the perspective of high-order bit planes. In this case, adversarial examples will behave the same as clean images in bit plane classifier and the attack attributes will disappear.

C. Overall Architecture of Defense

In this section, we introduce the overall architecture which has been shown in Fig. 2. To keep the accuracy of clean images, we still keep the target model as a part of the architecture. Here the n-bit classifiers use the top n bit planes from the highest order as input data and target model refer to the target classifier. The processing flow works as follows: First an RGB image is fed into the target classifier. Then we do bit plane slicing operation on the input image and get all 24 bit planes. A Bit Select function chooses different bit planes as inputs fed into the corresponding bit plane classifier. Finally, all results from target classifier and bit plane classifiers are fed into the Prediction module. The Prediction module gives a final result according to the averaging strategy.

The details of algorithm can be seen in Algorithm 1.

Algorithm 1: defense algorithm

```
Input: RGB image x; Label y; Number of class C;
       Target classifier f; Number of auxiliary
       classifiers k; Auxiliary classifier m =
       $\{m_1,m_2,...,m_k\}$; The top n bit planes used for
       each classifier from the highest order b =
       $\{b_1,b_2,...,b_k\}$; Weights for auxiliary classifier w
       = $\{w_1,w_2,...,w_k\}$;
Output: Classifier result r
result = f(x);
x_bit = BIT_SLICING(x);
for $i \in [1, k]$ do
    bit_planes = BIT_SELECT(x_bit, b[i]);
    classifier = m[i];
    weights = w[i];
    result += weights*classifier(bit_planes);
end
r = n where result[n] $\geq$ result[j] for $j \in [1, C]$;
return r;
```

III. Experiment Results

A. Experimental setup

All experiments are performed using pytorch on TITAN X with Ubuntu 16.04.4 LTS. Two RGB datasets CIFAR-10 [6] and GTSRB [7] are used to evaluate our defense effectiveness.

TABLE I: Model Architecture

Layer	Model(CIFAR-10)	Model(GTSRB)
Conv	32 filters with size (3 x 3)	32 filters with size (3 x 3)
Conv	32 filters with size (3 x 3)	32 filters with size (3 x 3)
BatchNorm		
Maxpool	kernel size (2 x 2)	kernel size (2 x 2)
Conv	64 filters with size (3 x 3)	64 filters with size (3 x 3)
Conv	64 filters with size (3 x 3)	64 filters with size (3 x 3)
BatchNorm		
Maxpool	kernel size (2 x 2)	kernel size (2 x 2)
Linear	256	256
Linear	128	128
Linear	10	43

Model Architectures. We refer to the model used in [11] with small modifications. We experiment with two target models on dataset CIFAR-10 and GTSRB. CIFAR-10 has 10 output classes and GTSRB has 43 output classes. Models of them are only different in the final output layer. The architecture of target models can be seen in Table I. We also have trained six bit plane classifiers using top 3 to top 8 bit planes from the highest order. In order to keep the quality of images, we constrain the number of bit planes to be at least 3. Bit plane classifiers have the same architecture as target model in Table I while they are different from inputs. Take 3-bit bit plane classifier as an example, the classifier has an input data with dimension 32x32x9 where 9 is the summation of RGB top 3 bit planes from the highest order.

Training. We train the target model in 200 epochs using Adam optimizer with learning rate 0.001. For the bit plane classifiers, we first get all bit planes of training data and then combine different bit planes to train the classifiers in 200 epochs using the same configuration as target model.

TABLE II: Accuracy of bit plane classifiers under attacks

bit plane	CIFAR-10			GTSRB		
classifier	FGSM	C&W	DeepFool	FGSM	C&W	DeepFool
3-bit	88.39%	93.97%	95.60%	96.12%	97.55%	96.56%
4-bit	88.37%	93.20%	95.71%	93.53%	97.71%	96.12%
5-bit	87.53%	92.94%	95.11%	92.24%	97.73%	96.60%
6-bit	87.39%	92.91%	95.29%	91.62%	97.99%	96.04%
7-bit	87.65%	92.40%	94.45%	93.81%	98.61%	98.20%
8-bit	87.47%	92.52%	93.88%	91.11%	97.63%	94.81%

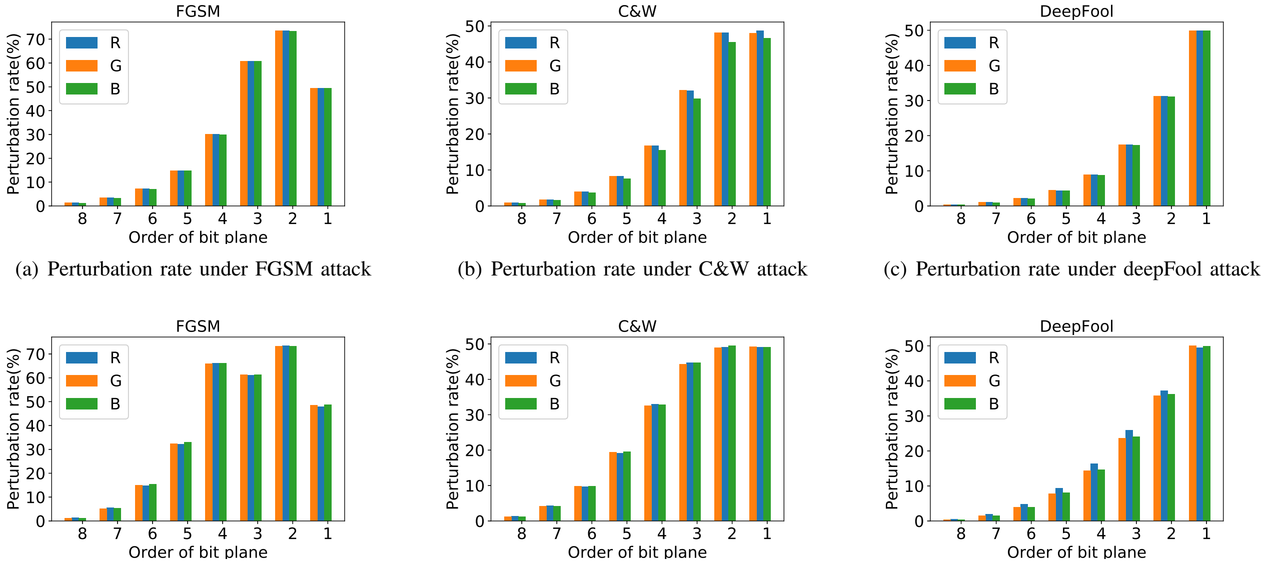
Attacks. We experiment with three famous attack methods FGSM [8], C&W [9] and DeepFool [10]. For all the attacks, L_2 norm is used as the distance metric. For FGSM attack, we set factor ϵ to 0.01 and 0.03 for CIFAR-10 and GTSRB. For C&W attack, the max-iterations is 20 and binary-search step is 4. For DeepFool attack, max-iteration is also 20.

B. Results

*1) **Bit Plane Perturbation Rate**:* We discuss how perturbations affect the bit planes of pixel value. Fig. 5 shows the perturbation rate in adversarial examples along all 24 bit planes. The X-axis represents the order of bit planes, 8 means the highest order and 1 means the lowest order of bit planes. The Y-axis perturbation rate represents how many numerical values in bit planes are different between clean images and adversarial examples. This rate is calculated from all the test data with an average. Since bit planes consist of only 0 and 1, XOR operation can be applied to calculate this rate efficiently. From the figure, we can see that when perturbations are added, most of them are located in the low-order bits and seldom affect the high-order bits, especially for attack methods with optimization-based solution like C&W and DeepFool. Besides, for the same order bit planes in R/G/B channels, they have the similar perturbation rate. As for the few perturbations in high-order bit planes, it is mainly from the carry bit propagated to the high bits when perturbations are added.

*2) **The Robustness of Bit Plane Classifiers**:* We train different bit plane classifiers and test the robustness of each. Take 3-bit classifier as an example, we first get all the test data which are classified correctly by it. Then we get adversarial examples from these data with different attack methods. When the adversarial examples are directly fed into the RGB-based target model, the accuracies are 26.17%, 0.01% and 0 for attack methods FGSM, C&W and DeepFool respectively on CIFAR-10. However, when they are fed into bit plane classifiers, the accuracy gets higher. Table II shows the accuracy of bit plane classifiers under attacks. Because C&W and DeepFool methods have fewer perturbations in high-order bits compared with FGSM method shown in Fig. 5, accuracy under C&W and DeepFool attack can be higher than FGSM. The result is also consistent with our findings in Fig. 4.

*3) **The influence of k (the number of bit plane classifiers)**:* The bit plane classifiers have been proved to be robust above. To make attacks more difficult, we combine these bit plane



(a) Perturbation rate under FGSM attack

(b) Perturbation rate under C&W attack

(c) Perturbation rate under deepFool attack

(d) Perturbation rate under FGSM attack

(e) Perturbation rate under C&W attack

(f) Perturbation rate under deepFool attack

Fig. 5: Perturbation rate in bit plane of adversarial example under different attacks. CIFAR-10 is used in (a)(b)(c). GTSRB is used in (d)(e)(f).

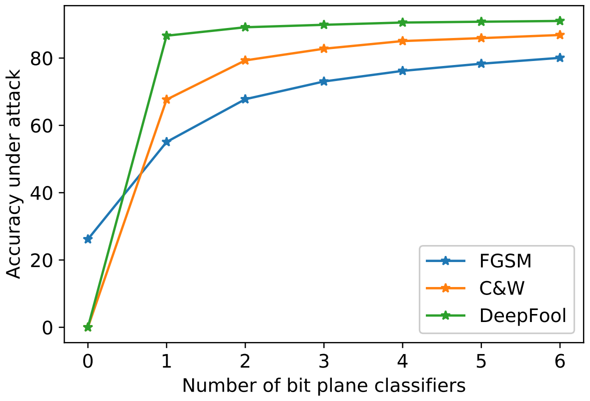


Fig. 6: CIFAR10: accuracy with various k.

classifiers with target model to form a more robust classification system. We test with different numbers of these classifiers from 0 to 6. 0 means that we do not add any bit plane classifiers which also means no defensive countermeasures are taken. Here we treat all the classifiers equally important, that is all weights are equal. Fig. 6 shows the result. X-axis represents the number of bit classifier, 1 means we use 3-bit classifier and 2 means we use 3-bit and 4-bit classifier together. Y-axis is the classification accuracy under attack. From the figure, we see that when no bit plane classifiers are used, the accuracy is very low, only 26.17% accuracy are kept under FGSM attack on CIFAR-10. The more powerful attack methods C&W and deepFool make accuracy drop to zero. However, when the robust bit plane classifiers are added for defense, the accuracy increases a lot. With more bit plane classifiers added, the accuracy becomes higher. We also test the accuracy of clean images when they are fed into the defense architecture. The accuracy is 81.12%, higher than 77.52% when the target model is used alone in CIFAR-10.

IV. Conclusion

In this paper, we take the bit plane of images as input features for classification. Bit planes as input features can filter the adversarial perturbations and decrease the channel modification rate. Therefore, we propose a new architecture which combines the target model with different robust bit plane classifiers to better defend against the adversarial attacks. The experiments show that our method can greatly improve the classification accuracy under attacks without decreasing the accuracy of clean images.

References

[1] T. He *et al.*, "Bag of tricks for image classification with convolutional neural networks," in *CVPR*, 2019, pp. 558–567.

[2] A. Graves *et al.*, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013, pp. 6645–6649.

[3] J. Choi *et al.*, "Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *ICCV*, 2019, pp. 502–511.

[4] C. Szegedy *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[5] R. Gonzalez and R. Woods, "Digital image processing, 4th edn." 2017.

[6] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[7] J. Stallkamp *et al.*, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012.

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

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *S&P*, 2017, pp. 39–57.

[10] Moosavi-Dezfooli *et al.*, "Deepfool: a simple and accurate method to fool deep neural networks," in *CVPR*, 2016, pp. 2574–2582.

[11] P. Qiu *et al.*, "Mitigating adversarial attacks for deep neural networks by input deformation and augmentation," in *ASP-DAC*, 2020, pp. 157–162.