Using an ensemble color space model to tackle adversarial examples

Shreyank N Gowda\textsuperscript{a,}\textsuperscript{*}, Chun Yuan\textsuperscript{b}

\textsuperscript{a}The University of Edinburgh, UK
\textsuperscript{b}Tsinghua University, China

Abstract

Minute pixel changes in an image drastically change the prediction that the deep learning model makes. One of the most significant problems that could arise due to this, for instance, is autonomous driving. Many methods have been proposed to combat this with varying amounts of success. We propose a 3 step method for defending such attacks. First, we denoise the image using statistical methods. Second, we show that adopting multiple color spaces in the same model can help us to fight these adversarial attacks further as each color space detects certain features explicit to itself. Finally, the feature maps generated are enlarged and sent back as an input to obtain even smaller features. We show that the proposed model does not need to be trained to defend an particular type of attack and is inherently more robust to black-box, white-box, and grey-box adversarial attack techniques. In particular, the model is 56.12 percent more robust than compared models in case of white box attacks when the models are not subject to adversarial example training.

Keywords: Adversarial, color spaces, robustness

\textsuperscript{*}Supported by NSFC project Grant No. U1833101, Shenzhen Science and Technologies project under Grant No. JCYJ20160428182137473 and the Joint Research Center of Tencent and Tsinghua.

\textsuperscript{*}Corresponding author

Email addresses: s1960707@ed.ac.uk (Shreyank N Gowda), yuanc@sz.tsinghua.edu.cn (Chun Yuan)
1. Introduction

Deep learning models have achieved extraordinary results in computer vision and natural language processing tasks due to the extensive amount of computing power available. They have shown near human capabilities in speech recognition [1], natural language processing [2], scene understanding tasks [3] and video recognition [4]. Although these algorithms show great promise, neural networks are black-box systems because there is restricted theoretical comprehension for their behavior. This lack of understanding has the potential to be dangerous.

Recent research has shown them to be low on security. Small perturbations to existing data have shown to yield negative classification results with high confidence. Recent research [5][6][7] has shown how these adversarial examples can be generated off existing data. These deftly crafted adversarial examples can also force the neural network to classify the given image into any chosen target class. These vulnerabilities of neural networks escalate critical questions regarding the robustness, dependability, and security of deep learning applications. Figure 1 shows one such example.

![Figure 1: An example of how noise added can change classification result of a model [5] (in this case from a panda to a gibbon)](image)

In [8] and [9] it was seen that speech or image input can be modified to resemble relevant input to the neural network but look/sound like complete
gibberish to humans. In [10], real-time input captured through a cell phone
camera can be subject to adversarial perturbations that result in complete mis-
classification. Authors in [11] proposed the use of manifold distance, while,
[12] utilized principal components to recognize adversarial attacks. These can
easily be skipped through by prior knowledge about the nature of the system.
Optimization-based methods that are utilized to develop provably-secure mod-
els are presently restricted to relatively smaller networks and the guarantees
provided are insufficient for real-life applications [13] [14].
Our paper provides a novel idea to make a model more robust without having
the need to train the model separately for attacks. Following are the steps,

• First, we perform statistical denoising of image. Second, we use an en-
semble model consisting of small but wide Densenet [15] models.

• Each model will be having an input mutated into a different color space.

• Third, the feature maps generated are enlarged in size and sent again to
the start of each densenet, to retrain the model to obtain even smaller and
precise features.

The details of the model and the reasons behind our choices will be explained
in further sections.

2. Related Work

We provide a compendium of adversarial attacks and techniques to defend
against them in the following subsections. The input image can be considered
as a 3-D tensor $I$ where the three dimensions are height, width, and colors.
Height and width represent the dimensions of the image and color represents
the number of colors the image displays.

Adversarial examples can be broadly classified to two types: targeted and
untargeted. Targeted examples implies that the generated example compels the
network to output any given image into a particular class and an untargeted ex-
ample simply makes the network mis-classify an image. Every example involves
adding a distortion $\delta$ to the image. $L_p$ norms are often used as measures for these distortions.

### 2.1. Adversarial Attacks

Adversarial samples were originally described in [7]. This subsection will briefly look into some of these attacks and what these attacks do. Fast gradient sign method (FGSM) was described in [5]. It is an $L_\infty$ type of attack and uses the gradient obtained from the loss function to determine the direction of modification for the pixels. It is represented in (1). $\epsilon$ represents the magnitude of distortion added and $y$ is the true label.

\[
I' = I - \epsilon \cdot \text{sign}(\nabla_I J(I, y_{true}))
\]  

(1)

The randomized fast gradient sign method (RAND-FGSM) was proposed in [16]. The mathematical explanation is shown in (2) and (3).

\[
I' = I + \text{rand}(\alpha)
\]  

(2)

\[
I'' = I' + (\epsilon - \alpha) \cdot \text{sign}(\nabla_I J(I', y_{true}))
\]  

(3)

The Carlini-Wagner attack proposed in [17] is an extremely powerful optimized based adversarial attack. It is a series of $L_0$, $L_2$ and $L_\infty$ attacks with comparatively lower distortions with the above mentioned attacks. The value of constant $c > 0$ and steps to obtain it can be found in [17].

Another attack method using FGSM as base was shown in [6]. It is called basic iterative method. It applies FGSM multiple times with a smaller step size. Equation (4),(5) describes it.

\[
I_0' = I
\]  

(4)

\[
I_{n+1}' = I_n' + \alpha \cdot \text{sign}(\nabla_I J(I_{n+1}', y_{true}))
\]  

(5)

Another form of attack is called the Jacobian saliency attack[18]. It is a computationally expensive algorithm. It is an $L_0$ form of attack. It basically chooses the most influential pixels by utilizing a greedy algorithm that calculates
the Jacobian saliency map. Fault injection attack was proposed in [19]. The idea was to make slight changes to the DNN parameters to cause misclassification.

An attack called Deepfool (DF) was proposed in [20]. It determines an approximation of the decision boundary to perceive possible locations for adversarial perturbations. Attacks performed knowing the classifier details such as model parameters, weights etc are called white box attacks. Some attacks treat the classifier as a black box and hence these attacks are called black-box attacks.

2.2. Defense strategies

One of the earliest proposed techniques was to train the neural network to adversarially generated samples [16]. Training with adversarial data made the neural network more robust to adversarial attacks. Label-smoothing [21] was shown to perform well against certain types of adversarial attacks. In the context of purifying the samples, generative models have been utilized [22] to varying degrees of success.

Use of auxiliary networks to clean samples also showed marked improvement in results against adversarial attacks [22][23]. In [24] defensive distillation was proposed, it trains the classifier using a variant of distillation [25]. While it gave improved results for white-box attacks, it failed to protect the network from black-box attacks [17].

MagNet [23] trains a collection of autoencoder networks (called as reformer networks) to make adversarial examples seem more natural. At test time, one autoencoder is randomly chosen and this magnifies the strength of their strategy. More recently, Defense-GAN was proposed [26], it uses the generator part of the GAN to sanitize the input before passing it to the classifier. APE-GAN [27] amended the generator part to clean the adversarial samples and the discriminator differentiates between real and adversarial inputs.
3. Proposed Method

3.1. Denoising and Color spaces

Denoising can be considered as a minimizing problem where we minimize a loss and a penalty. We propose the use of the L1-norm of pixel updates as the penalty. The advantage of using L1-norms is that we deal directly with only the noisy pixels. We are given a noisy wavelet coefficient of a noisy image \( y_i \), we want to recover the noise-free wavelet coefficient \( x_i \) via a MAP estimator \( \hat{x}_i \). We assume that the noise in the image can be represented in the form of Gaussian noise with zero mean and standard deviation \( \sigma_n \). Then we can express \( p(y_i|x_i) \) using Baye’s rule as seen in (6). The advantage of using this non-parametric approach is that the model automatically adapts to the observed image data.

\[
p(y_i|x_i) = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-\frac{(y_i - x_i)^2}{2\sigma_n^2}}
\]

A color space is basically an organization of colors. We can also look at color spaces as an abstract mathematical model that aids us to portray colors as numbers. For more details about the theory between different color spaces, one can refer to [28]. ColorNet [29] shows us that using an ensemble model of different color spaces help us to obtain state of the art classification results. Also, based on the confusion matrix in [29] we can say that the activation map for each image is slightly different after preprocessing the image into a different color space.

3.2. Architecture of model and novelties proposed

We first denoise the images as explained earlier. Next, we pass the image after denoising to our network. Our network comprises of 4 small but wide DenseNets [30]. We use a wide-approach for the DenseNet [15] as it reduces the parameters. The denoised image is sent as input to the first densenet, while further preprocessing is done in the form of color space conversion to LAB, HSV and YUV. We reduce the number of color spaces based on individual accuracy.
to aid the time of execution needed for our approach. The final activation map is taken and passed as a recurrent input after resizing the image based on the activation map size. This process is shown in Figure 2. This progressive resizing helps to detect tinier features, increasing the robustness of the network. The entire architecture is shown in Figure 3. We take the feature map at the last layer, collect the part of the image with the highest density and enlarge the image to the original size of the image (for instance 32x32 in the CIFAR datasets [32]). Hence, each image is used as an input twice to the model.

![Figure 2: Progressive resizing of activation map flowchart](image)

First, we denoise the image using a statistical approach. Second, to the best of our knowledge, color spaces have not been explored with regards to adversarial attacks. Third, progressive resizing, is a novel technique that uses the feature map of an image to augment the dataset by enlarging the map and using the same as an input to the model.

4. Experimental Analysis

We test our model on the 3 types of attacks discussed earlier namely: white-box, grey-box and black-box attacks.
4.1. Training

As discussed in the earlier section, our proposed model denoises the original input image. This modified image is sent as input to one small densenet model. The image is simultaneously processed into LAB, YUV and HSV images and sent as inputs to corresponding densenets. The activation map at the final layer is sent back as an input to the densenet by cropping out the original input with the densest parts of the activation map, and this cropped image is resized to the original image and again sent as input. This progressive resizing technique is done to obtain lower level features that help to detect an object with higher accuracy.

4.2. White-box attacks

We compare our results with Resnet-50 [33], VGG-19 [34], Pixel-Deflection (PD) [35], MagNet [24]. The algorithms used for the attacks are FGSM, IGSM, D Fool, JSMA, C and W. Table 1 contains the comparison results. The results obtained are from choosing a random subset of the imagenet dataset and taking
the average prediction over 6 runs. Since, the training involves addition of a different adversarial noise we get different results. Averaging them gives a higher level of accuracy. For reference purposes, table 2 highlights the results of how the various models perform when trained to deal with particular types of adversarial examples. As can be seen from table 1 our proposed model is significantly more robust to adversarial attacks and performs reasonably well without training the model for defense against attacks. In terms of numbers, the proposed method is 56.12 percent more robust without training. Although, after training it performs slightly worse than the other mentioned models. Even then, the model only performs 1.08 percent worse on average.

4.3. Black-box and grey-box attacks

Generating black box attacks are more difficult than white-box attacks and hence we use models that already have black-box attacks performed on them and compare the results with our model. Again, we use the vast imagenet
We compare the top-1 accuracy of our model against the models proposed in [37] (divide, denoise and dispatch or D3) and [38]. Table 3 shows the results obtained. Attacks used were deep fool and FGSM. In this case, it can be seen that the proposed model performs significantly better than the models in comparison and that the black-box attacks have insignificant impact on the model.

Grey-box attacks are attacks wherein the adversary has access to the network weights, but does not have an idea about the defense mechanism in place. We do the same comparisons as with the black-box attacks. Table 4 shows the results obtained. Again, as seen with black-box attacks, the model shows high robustness to adversarial attacks.
5. Conclusion

In this paper, we propose a novel algorithm that shows robustness to adversarial attacks. It follows a 3 step process. First, the image is preprocessed by denoising all the images. This helps the model to not overfit on specific features. Next, we use a color-space ensemble model. This was on the basis that certain classes of images were being represented better by certain color spaces. Finally, we use progressive image resizing by cropping out the part of the image that has highest density of features and then resizing into the original input size and passing it back to the network as an input. This helps us detect even smaller features and hence can detect an object correctly under adversarial conditions. Based on the results we could see that the proposed model was 56.12 percent more robust without training.

References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal processing magazine 29 (6) (2012) 82–97.

[2] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.

[3] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3128–3137.

[4] S. N. Gowda, Human activity recognition using combinatorial deep belief networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1–6.

[5] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).
[6] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, arXiv preprint arXiv:1607.02533 (2016).

[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[8] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, W. Zhou, Hidden voice commands, in: 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 513–530.

[9] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 427–436.

[10] S. N. Gowda, Fiducial points detection of a face using rbf-svm and adaboost classification, in: Asian Conference on Computer Vision, Springer, 2016, pp. 590–598.

[11] R. Feinman, R. R. Curtin, S. Shintre, A. B. Gardner, Detecting adversarial samples from artifacts, arXiv preprint arXiv:1703.00410 (2017).

[12] J. H. Metzen, T. Genewein, V. Fischer, B. Bischoff, On detecting adversarial perturbations, arXiv preprint arXiv:1702.04267 (2017).

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2017).

[14] A. Raghunathan, J. Steinhardt, P. Liang, Certified defenses against adversarial examples, arXiv preprint arXiv:1801.09344 (2018).

[15] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[16] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses, arXiv preprint arXiv:1705.07204 (2017).

[17] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 ieee symposium on security and privacy (sp), IEEE, 2017, pp. 39–57.

[18] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European symposium on security and privacy (EuroS&P), IEEE, 2016, pp. 372–387.

[19] Y. Liu, L. Wei, B. Luo, Q. Xu, Fault injection attack on deep neural network, in: 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2017, pp. 131–138.

[20] S.-M. Moosavi-Dezfooli, A. Fawzi, 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, 2016, pp. 2574–2582.

[21] D. Warde-Farley, I. Goodfellow, 11 adversarial perturbations of deep neural networks, Perturbations, Optimization, and Statistics 311 (2016).

[22] A. v. d. Oord, N. Kalchbrenner, K. Kavukcuoglu, Pixel recurrent neural networks, arXiv preprint arXiv:1601.06759 (2016).

[23] D. Meng, H. Chen, Magnet: a two-pronged defense against adversarial examples, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 135–147.

[24] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 582–597.
[25] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 (2015).

[26] P. Samangouei, M. Kabkab, R. Chellappa, Defense-gan: Protecting classifiers against adversarial attacks using generative models, arXiv preprint arXiv:1805.06605 (2018).

[27] G. Jin, S. Shen, D. Zhang, F. Dai, Y. Zhang, Ape-gan: Adversarial perturbation elimination with gan, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 3842–3846.

[28] A. Rasouli, J. K. Tsotsos, The effect of color space selection on detectability and discriminability of colored objects, arXiv preprint arXiv:1702.05421 (2017).

[29] S. N. Gowda, C. Yuan, Colornet: Investigating the importance of color spaces for image classification, in: Asian Conference on Computer Vision, Springer, 2018, pp. 581–596.

[30] S. Zagoruyko, N. Komodakis, Wide residual networks, arXiv preprint arXiv:1605.07146 (2016).

[31] S. N. Gowda, C. Yuan, Stegcolnet: Steganalysis based on an ensemble colorspace approach, arXiv preprint arXiv:2002.02413 (2020).

[32] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

[33] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[34] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[35] A. Prakash, N. Moran, S. Garber, A. DiLillo, J. Storer, Deflecting adversarial attacks with pixel deflection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8571–8580.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[37] S.-M. Moosavi-Dezfooli, A. Shrivastava, O. Tuzel, Divide, denoise, and defend against adversarial attacks, arXiv preprint arXiv:1802.06806 (2018).

[38] C. Guo, M. Rana, M. Cisse, L. Van Der Maaten, Countering adversarial images using input transformations, arXiv preprint arXiv:1711.00117 (2017).