Robust Pipeline for detecting adversarial images

Ankit Vohra* Abhibha Gupta* Gaurav Agarwal* Pooja Jain
IIIT Nagpur IIIT Nagpur IIIT Nagpur IIIT Nagpur

***Abstract*— Deep neural networks (DNN's) have achieved state of the art for many image related tasks like classification, object detection etc. [1] have shown that DNN's can be easily fooled by adversarial examples that can be generated by adding small distortions or perturbations to the images. Several attacks like the FGSM [1], PGD [2], etc have been proposed to create distortions in images which cause the DNN's to misclassify instances. For example, image detection systems in self driving cars can be deceived by creating 'toxic signs' that are adversarially attacked images of signboards. The system may detect a stop sign as a 'speed limit 100' sign causing accidents.Hence its is essential to come up with strategies in order to identify such spoofed images.**

Keeping this in mind, we propose an end to end pipeline capable of detecting adversarially attacked images i.e images with perturbations. Fora given image, we generate images with manipulation masks using ManTra-Net [3] and pass them through a trained binary classifier that distinguishes between perturbed and original images. Experiments on the ILSVRC 2012 [4] dataset have shown that our pipeline achieves an accuracy of 0.98 on the task of detecting adversarially crafted images.

Keywords- Adversarial attacks, ManTra-Net, Deep Neural Networks,

I. INTRODUCTION

Deep learning has fueled great strides in a variety of computer vision problems, such as object detection (e.g., [[5], [6]]), motion tracking (e.g., [10]), action recognition (e.g., [12]), human pose estimation (e.g., [14]), and semantic segmentation. Despite there success, research has shown that DNN's are broadly vulnerable to adversarial examples, carefully chosen inputs that cause the network to change output without a visible change to a human [7]. Yet, for humans these perturbations are often visually imperceptible and do not stir any doubt about the correct classification. In fact, so called adversarial examples are crucially characterized by requiring minimal perturbations that are quasi-imperceptible to a human observer. For computer vision tasks, multiple techniques to create such adversarial examples have been developed like some common methods modify each pixel by only a small amount and can be found using a number of optimization strategies such as L-BFGS [7], Fast Gradient Sign Method (FGSM) [1], DeepFool [8], Projected Gradient Descent (PGD) [9], as well as the recently proposed Logit-space Projected Gradient Ascent (LS-PGA) [10] for discretized inputs. Other attack methods seek to modify a few pixels in the image (Jacobian-based saliency map [11]), or a small patch at a fixed location of the image [12]. This has serious consequences as shown by [13]. Image detection systems in self driving cars can be deceived by creating 'toxic signs' that are adversarially attacked images of signboards. The system may detect a stop sign as a 'speed limit 100 sign' causing accidents. Many detection strategies have been

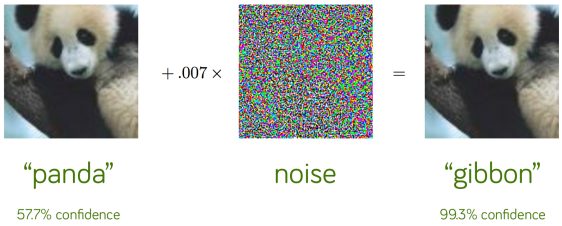

Fig. 1. Example of Adversarial Attack

proposed involving statistical methods [14], augmentation of model [15] [16], novel training techniques [17] and feature squeezing][18].

We propose a simple end to end pipeline that can detect such adversarially crafted images. We train our pipeline on the ILSVRC 2012 dataset. To improve the robustness of our proposed method we incorporate adversarial examples generated using the Fast Gradient Signed Attack (FGSM), Projected Gradient Descent (PGD) Attack and the Patch attack into the training process. Experiments have shown that our method performs well irrespective of the attack method used.
[1]

II. RELATED WORK

Attacking DNN's by adding imperceptible carefully chosen noise (adversarial perturbations) to images was first investigated by [15]. The researchers found out that the trained model incorrectly classified instances that too with high confidence. Adversarial examples have also been shown to be propagated in the real world i.e these images can remain adversarial even if clicked with an eternal camera. [16] have outlined that adversarial examples can be constructed for an unknown network by training an auxiliary network on similar data and exploiting the transferability of adversarial inputs. iTh this vulnerability becomes a serious issue when human safety is involved. Most of the work on addressing adversarial attacks focuses on increasing the robustness of the model itself for the detection of adversarial examples.

[1]* signifies equal contribution

Techniques like adversarial training [17] and knowledge distillation have been shown to be effective methods to tackle adversarially attacked images. There drawback is that they suffer from the problem of generalisation. [15] have trained a binary classifier that separates the adversarial part from the clean data. They receive good accuracy scores of 0.99 but their results are sensitive to epsilon values, dropping to 0.003 for epsilon = 0.001. While, [14] have shown that adversarial examples are not drawn from the same distribution than the original data, and can thus be detected using statistical tests. They augment their machine learning model with an additional input to classify adversarial examples. Though the model gives good accuracy but it fails to perform for large sample sizes on the MNIST dataset and for some attacks like the decision tree attack. [16] augment their DNN with a small 'detector' subnetwork trained on the binary classification task of distinguishing genuine data from data containing adversarial perturbations. Their network generalises to other adversaries as well. They also propose a new adversarial attack that fools both the classifier and the detector and a novel training procedure for the detector that counteracts this attack. Finally, [18] propose a novel strategy, feature squeezing to improve the detection capability of DNN's. They compare the DNN's prediction on the original vs the squeezed inputs and show that feature squeezing detects adversarial examples with high accuracy. They explore two feature squeezing methods, reducing the color bit depth of each pixel and spatial smoothing. We aim to address the above mentioned limitations by proposing an attack independent deep learning pipeline that is capable of detecting adversarial examples from a set of samples. During training, we experiment with different epsilon values and attack methods (FGSM, PGD, Patch attack) to evaluate the robustness of our pipeline.

III. BACKGROUND

A. Fast Gradient Signed Method (FGSM) attack

[1] hypothesized that DNNs are vulnerable to adversarial perturbations because of their linear nature . They proposed the fast gradient sign method (FGSM) for efficiently finding adversarial examples. To control the cost of attacking, FGSM assumes that the attack strength at every feature dimension is the same, essentially measuring the perturbation ∂**(x,x')** using the L∞-norm. The strength of perturbation at every dimension is limited by the same constant parameter, which is also used as the amount of perturbation. As an untargeted attack, the perturbation is calculated directly by using gradient vector of a loss function:

$$(\partial)(x, x') = (\epsilon)(\cdot)sign((\partial)xJ(g(x), y)) \qquad (3)$$

Here the loss function, J(·, ·), is the loss that have been used for training the specific DNN model, and y is the correct label for **x**. Equation (3) essentially increases the loss J(·, ·) by perturbing the input **x** based on a transformed gradient.

B. Patch attack

[12] have introduced the concept of 'adversarial patches' that are a form of targeted attack that can cause a classifier to output any target class. To obtain the trained patch pb we use a variant of the Expectation over Transformation (EOT) framework of [19]. In particular, the patch is trained to optimize the objective function

$$\hat{p} = argmax_{pEx\sim X,t\sim T,l\sim L}[logPr(\hat{y}|A(p, x, l, t)] \qquad (1)$$

where X is a training set of images, T is a distribution over transformations of the patch, and L is a distribution over locations in the image. While images may contain several items, only one target label is considered true, and thus the network must learn to detect the most "salient" item in the frame. The adversarial patch exploits this feature by producing inputs much more salient than objects in the real world. These patches can be added to any image which causes the model to ignore the other parts of the image and concentrate on the patch itself, thus predicting the chosen target class.

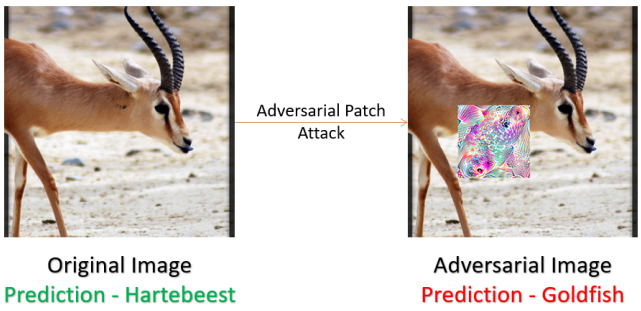

Fig. 2. Adversairal Patch due to which model misclassifies Hartebeest to Goldfish

C. Projected Gradient Descent (PGD) attack

It is essentially a white box attack i.e the model gradients are accessible to the attacker. This threat model provides more power to the attacker unlike black box attacks as they can specifically craft their attack to fool your model without having to rely on transfer attacks that often result in human-visible perturbations. PGD attempts to find the perturbation that maximises the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount referred to as epsilon. This constraint is usually expressed as the L^2 or L∞ norm of the perturbation and it is added so the content of the adversarial example is the same as the unperturbed sample

D. MANTRA-NET

It is a deep neural architecture proposed by [3] which is an end to end framework that performs both detection and localisation without extra preprocessing and postprocessing. It is a Convolutional Neural Network (CNN) that can handle arbitrarily sized images and other forgery types like splicing, copy-move, removal, enhancement, and even

unknown types. It detects forged pixels by identifying local anomalous features, and thus is not limited to a specific forgery or manipulation type. Experiments carried out by the researchers have demonstrated the generalizability, robustness and superiority of ManTra-Net, not only in single types of manipulations/forgeries, but also in their complicated combinations

E. ILSVRC 2012 Dataset

The ILSVRC 2012 dataset [4] is a subset of the ImageNet dataset (10,000,000 labeled images, 10,000+ object categories) with 10000 classes and 50,000 samples.

IV. METHODOLOGY

Ours is a binary classification problem i.e to determine whether an image is adversarialy attacked or not. We determine this by proposing two pipelines, the training and the testing pipeline.

A. Training pipeline

This pipeline consists of 3 steps. First is generating adversarially attacked images using different attacking methods (FGSM, PGD and Patch based attack) from the ILSVRC 2012 dataset . Samples are generated for different values of epsilon for the FGSM attack. and for the PGD attack we experiment with different placements of the adversarial patch on the image. A subset of the generated adversarial samples(100K samples) is considered as an input to the next step. Next, we pass the generated samples through Mantranet that identifies perturbations and adds manipulation masks to the affected areas of the image. A new dataset consisting of the outputs from the previous step that are the images with manipulation masks and the corresponding original images is created. Finally a binary classifier is trained on the new dataset to differentiate between real and fake images.

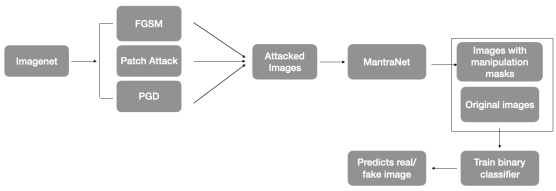

Fig. 3. Training Pipeline

B. Testing Pipeline

It consists of 2 stages. For a particular image, it is passed through MantraNet to generate images with manipulation masks. Next the trained binary classifier is used to predict the class of the sample generated from the previous step. If manipulations are detected by the binary classifier then it is said that the original image was an attacked image, else its concluded it was not an adversarialy attacked image.

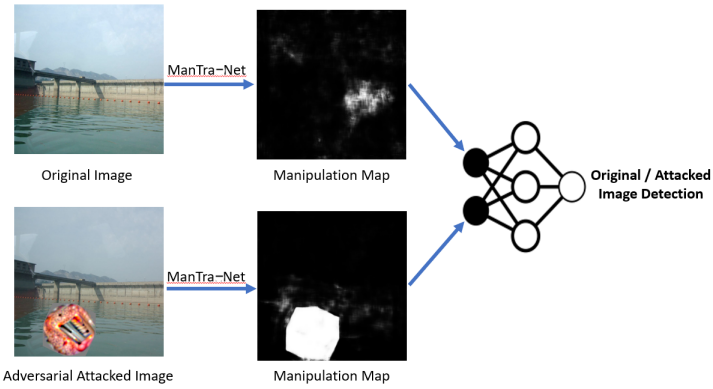

Fig. 4. Testing Pipeline

C. Metrics - Accuracy

Accuracy is used for evaluating classification models. It is the ratio of number of correct predictions to the total number of input samples. Mathematically,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

V. EXPERIMENTS

A. Creation of Adversarial image dataset

The first task was to build the training pipeline which includes collecting adversarial Samples. Since there was no benchmark dataset available incorporating multiple types of attacks like FGSM, PGD, Patch Attack, Augmentation using goggles/mask etc, we collected the ILSVRC2012. validation data and performed multiple types of attacks on the dataset. 50,000 samples consisting of adversarial perturbations were generated using FGSM (25000) and PGD (25000) attacks. We also incorporated around 10,000 samples of patch based attack into the dataset. The total 100k samples thus obtained comprise of the input dataset to Mantra-Net.

B. Extraction of manipulation masks

Once we have the input dataset, the next step is to identify the manipulations in the image. We pass the dataset through Mantra-Net which detects the manipulation masks on the image. Using the manipulation mask a binary classifier will be trained to identify whether the given image has artifical perturbations or not.

We encountered problems with images of dimensions less than 150*150 px. For such images with small sizes the pixels are very hazy and the change in pixel values is high, hence Mantra-Net wasn't able to detect the manipulation masks accurately in this case.

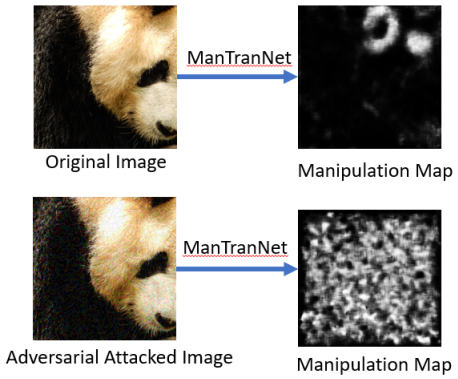

Fig. 5. Generation of Manipulation mask

C. Training a binary classifier.

Once we have collected a Trained A Binary Classifier On ManTraNet manipulation masks. Trained on all the manipulation masks of Original and Adversarial Image Achieved 99.6

D. Complications

Coming soon...

VI. RESULTS

TABLE I
COMPARISON OF RESULTS

Method	Val Accuracy	Remarks
Gong et al	0.99	sensitive to epsilon values
Resnet50	0.99	fgsm attack
Metzen et al	0.40	on ImageNet dataset
Xu et al	0.68	on ImageNet dataset
Binary classifier*	0.59	on ILSVRC 2012 dataset
Our method	0.98	attack independent

* denotes that binary classifier is trained without the Mantra-Net outputs.

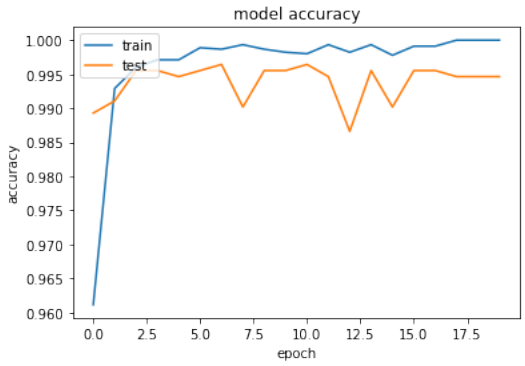

Fig. 6. Training and Validation accuracy of proposed pipeline

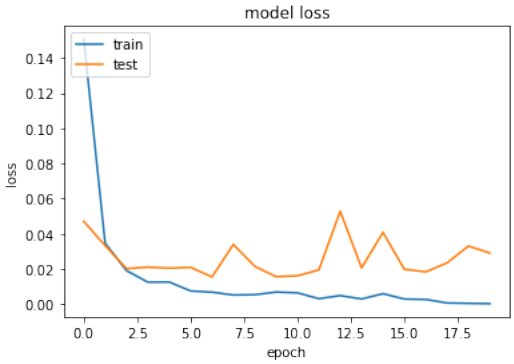

Fig. 7. Training and Validation loss of proposed pipeline

VII. CONCLUSION AND FUTURE WORK

- We introduce a simple method to detect spoofed images, irrespective of the attack method used. We obtain a good accuracy score of 0.98.
- To further improve results a dataset can be annotated to train ManTra-Net for the specific task of detecting manipulation masks
- Though the pipeline has been evaluated on augmented images (Ex: addition of glasses), experiments need to be carried out on a benchmark dataset to further evaluate the pipeline.

REFERENCES

[1] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).
[2] S. Bubeck, Convex optimization: Algorithms and complexity, arXiv preprint arXiv:1405.4980 (2014).
[3] Y. Wu, W. AbdAlmageed, P. Natarajan, ManTra-Net: Manipulation tracing network for detection and localization ofimage forgeries with anomalous features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252. doi:10.1007/s11263-015-0816-y.
[5] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, et al., Deepid-net: Object detection with deformable part based convolutional neural networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (7) (2016) 1320–1334.
[6] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, L. Van Gool, Weakly supervised cascaded convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 914–922.
[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations, 2014.
URL http://arxiv.org/abs/1312.6199
[8] 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 (CVPR), 2016.
[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2017).
[10] J. Buckman, Aurko roy, colin raffel, and ian goodfellow, Thermometer encoding: One hot way to resist adversarial examples 1 (1) (2018) 2–2.

[11] 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), 2016, pp. 372–387. doi:10.1109/EuroSP.2016.36.

[12] T. B. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer, Adversarial patch, arXiv preprint arXiv:1712.09665 (2017).

[13] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, P. Mittal, Darts: Deceiving autonomous cars with toxic signs, arXiv preprint arXiv:1802.06430 (2018).

[14] K. Grosse, P. Manoharan, N. Papernot, M. Backes, P. McDaniel, On the (statistical) detection of adversarial examples, arXiv preprint arXiv:1702.06280 (2017).

[15] Z. Gong, W. Wang, W.-S. Ku, Adversarial and clean data are not twins, arXiv preprint arXiv:1704.04960 (2017).

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

[17] A. Kurakin, I. J. Goodfellow, S. Bengio, Adversarial machine learning at scale, CoRR abs/1611.01236 (2016). arXiv:1611.01236.
URL http://arxiv.org/abs/1611.01236

[18] W. Xu, D. Evans, Y. Qi, Feature squeezing: Detecting adversarial examples in deep neural networks, arXiv preprint arXiv:1704.01155 (2017).

[19] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, Synthesizing robust adversarial examples, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 284–293.
URL http://proceedings.mlr.press/v80/athalye18b.html