Trace and Detect Adversarial Attacks on CNNs using Feature Response Maps

Mohammadreza Amirian[1,2], Friedhelm Schwenker[2], and Thilo Stadelmann[1]

[1] ZHAW Datalab & School of Engineering, Winterthur, Switzerland
[2] Institute of Neural Information Processing, Ulm University, Germany
amir@zhaw.ch, friedhelm.schwenker@uni-ulm.de, stdm@zhaw.ch

Abstract. The existence of adversarial attacks on convolutional neural networks (CNN) questions the fitness of such models for serious applications. The attacks manipulate an input image such that misclassification is evoked while still looking normal to a human observer—they are thus not easily detectable. In a different context, backpropagated activations of CNN hidden layers—"feature responses" to a given input—have been helpful to visualize for a human "debugger" what the CNN "looks at" while computing its output. In this work, we propose a novel detection method for adversarial examples to prevent attacks. We do so by tracking adversarial perturbations in feature responses, allowing for automatic detection using average local spatial entropy. The method does not alter the original network architecture and is fully human-interpretable. Experiments confirm the validity of our approach for state-of-the-art attacks on large-scale models trained on ImageNet.

Keywords: neural network · diagnostic · robustness · practical applications

1 Introduction

The success of deep neural nets for pattern recognition [37] has been a main driver behind the recent surge of interest in AI. Arguably, this success is due to the Convolutional Neural Net (CNN) [12,22,5] and its descendants, applied to image recognition tasks. Respective methods have reached the application level in business and industry [40] and lead to a wide variety of deployed models for critical applications like automated driving [2] or biometrics [48].

However, concerns regarding the reliability of deep neural networks have been raised through the discovery of so-called adversarial examples [43]. These inputs are specifically generated to "fool" [30] a classifier by visually appearing as some class (to humans), but being misclassified by the network with high confidence through the addition of barely visible perturbations (see Figure 1). The perturbations are achieved by an optimization process on the input: the network weights are fixed, and the input pixels are optimized for the dual criterion of (a) classifying the input differently than the true class, and (b) minimizing the changes to the input. A growing body of literature confirms the impact of this discovery on

practical applications of neural nets [1]. It raises questions on how—and in what respect different from humans—they achieve their performance, and threatens serious deployments with the possibility of tailor-made adversarial attacks.

For instance, Su et al. [42] report on successfully attacking neural networks by modifying a single pixel. The attack works without having access to the internal structure nor the gradients in the network under attack. Moosavi-Dezfooli et al. [29] furthermore show the existence of universal adversarial perturbations that can be added to any image to fool a specific model, whereas transferability of perturbations from one model to another is for example shown by Xu et al. [46]. The impact of similar attacks extends beyond classification [28], is transferable to other modalities than images [6], and also works on models distinct from neural networks [33]. Finally, adversarial attacks have been shown to work reliably even after perturbed images have been printed and captured again via a mobile phone camera [20]. Apparently, such research touches a weak spot.

On the other hand, there is a recent interest in the interpretability of AI agents and in particular machine learning models [44,7,32]. It goes hand in hand with societal developments like the new European legislation on data protection that is impacting any organization using algorithms on personal data [15]. While neural networks are publicly perceived as "black boxes" with respect to how they arrive at their conclusions [17], several methods have been developed recently to allow insight into the representation and decision surface of a trained model, improving interpretability. Prime candidates amongst these methods are feature visualization approaches that make the operations in hidden layers of a CNN visible [9,47,39,31]. They can thus serve a human engineer as a diagnostic tool in support of reasoning over success and failure of a model on the task at hand.

In this paper, we propose to use a specific form of CNN feature visualization, namely feature response maps, to not only *trace* the effect of adversarial inputs on algorithmic decisions throughout the CNN; we subsequently also use it as input to a novel automated *detection* approach, based on statistical analysis of the feature responses using average of image local spatial entropy. The goal is to decide if a model is currently under attack by the given input. Our approach has the advantage over existing methods of not changing the network architecture, i.e., not affecting classification accuracy; and of being interpretable both to humans and machines, an intriguing property also for future work on the method. Experiments on the validation set of ImageNet [36] with VGG19 networks [38] shows the validity of our approach for detecting various state-of-the-art attacks.

Below, Section 2 reviews related work in contrast to our approach. Section 3 presents the background on adversarial attacks and feature response estimation before Section 4 introduces our approach in detail. Section 5 reports on experimental evaluations, and Section 6 concludes with an outlook to future work.

2 Related work

Work on adversarial examples for neural networks is a very active research field. Potential attacks and defenses are published at a high rate and have been sur-

veyed recently by Akhtar and Mian [1]. Amongst potential defenses, directly comparable to our approach are those that focus on the sole detection of a possible attack and not on additionally recovering correct classification.

On one hand, several detection approaches exist that exploit specific abnormal behavioral traces that adversarial examples leave while passing through a neural network: Liang et al. [24] consider the artificial perturbations as noise in the *input* and attempt to detect it by quantizing and smoothing image filters. A similar concept underlies the SqueezeNet approach by Xu et al. [45], that compares the network's *output* on the raw and filtered input, and raises a flag if detecting a large difference between both. Feinman et al. [10] observe the network's output confidence as estimated by dropout in the forward pass [13], and Lu et al's SafetyNet [25] looks for abnormal patterns in the ReLU activations of *higher layers*. In contrast, our method performs detection based on statistics of activation patterns in the complete *representation learning* part of the network as observed in feature response maps, whereas Li and Li [23] directly observe convolutional filter statistics there.

On the other hand, a second class of detection approaches trains sophisticated classifiers for directly sorting out malformed inputs: Meng and Chen's MagNet [26] learns the manifold of friendly images, rejects far away ones as hostile and modifies close outliers to be attracted to the manifold before feeding them back to the network under attack. Grosse et al. [16] enhance the output of an attacked classifier by an additional class and retrain the model to directly classify adversarial examples as such. Metzen et al. [27] have a similar goal but target it via an additional subnetwork. In contrast, our method uses a simple threshold-based detector and pushes all decision power to the human-interpretable feature extraction via the feature response maps.

Finally, as shown in [1], different and mutually exclusive explanations for the existence of adversarial examples and the nature of neural network decision boundaries exist in the literature. Because our method enables a human investigator to trace attacks visually, it can be helpful in this debate in the future.

3 Background

We briefly present adversarial attacks and feature response estimation in general before assembling both parts into our detection approach in the next Section.

3.1 Adversarial attacks

The main idea of adversarial attacks is to find a small perturbation for a given image that changes the decision of the Convolutional Neural Network. Pioneering work [43] demonstrated that negligible and visually insignificant perturbations could lead to considerable deviations in the networks' output. The problem of finding a perturbation $\boldsymbol{\eta}$ for a normalized clean image $\boldsymbol{I} \in \mathbb{R}^m$, where m is the image width × height, is stated as follows [43]:

$$\min_{\boldsymbol{\eta}} \| \boldsymbol{\eta} \|_2 \quad \text{s.t.} \quad \mathscr{C}(\boldsymbol{I}+\boldsymbol{\eta}) \neq \ell \;\; ; \quad \boldsymbol{I}+\boldsymbol{\eta} \in [0,1]^m \tag{1}$$

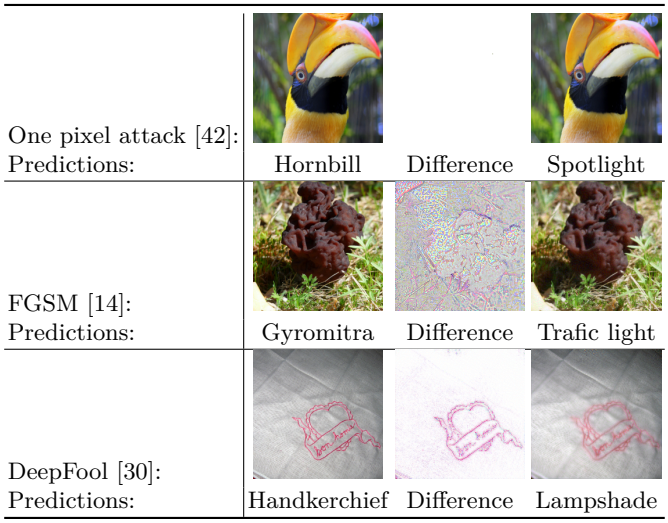


Figure 1: Examples of different state-of-the-art adversarial attacks on a VGG19 model: original image and label (left), perturbation (middle) and mislabeled adversarial example (right). In the middle column difference of zero is encoded white and maximum difference is black because of visual enhancement.

where $\mathscr{C}(.)$ presents the classifier and ℓ is the ground truth label. Szegedy et al. [43] proposed to solve the optimization problem in Equation 1 for an arbitrary label ℓ' that differs from the ground truth to find the perturbation. However, box-constrained L-BFGS [11] is alternatively used to find perturbations satisfying Equation 1 to improve computational efficiency. Optimization based on the L-BFGS algorithm for finding adversarial attacks are computational inefficient compared with gradient-based methods. Therefore, in this paper, we use the following gradient-based attacks to compute adversarial examples in addition to a one-pixel attack and boundary attack (see Figure 1).

Fast Gradient Sign Method (FGSM) Goodfellow et al. [14] suggested to compute adversarial perturbations based on the gradient $\nabla_{\boldsymbol{I}} J(\boldsymbol{\theta}, \boldsymbol{I}, \ell)$ of the cost function with respect to the original image pixel values:

$$\boldsymbol{\eta} = \epsilon \ \text{sign}(\nabla_{\boldsymbol{I}} J(\boldsymbol{\theta}, \boldsymbol{I}, \ell)) \tag{2}$$

where $\boldsymbol{\theta}$ represents the network parameters and ϵ is a constant factor that constrains the max-norm l_∞ of the additive perturbation η. The ground truth label is presented by ℓ in Equation 2. Optimizing the perturbation in Equation 2 in a single step is called Fast Gradient Sign Method (FGSM) in the literature. This method is a white box attack, i.e. the algorithm for finding the adversarial example requires the information of weights and gradients of the network.

Gradient attack is a simple and straightforward realization of finding adversarial perturbations in the FoolBox toolbox [35]. It optimizes pixel values of an original image to minimize the ground truth label confidence in a single step.

One pixel attack [42] is a semi-black box approach to compute adversarial examples using differential evolution [41]. The algorithm is not white box since it does not need the gradient information of the classifier; however, it is not fully black box as it needs the class probabilities from the network. The iterative algorithm starts with a fixed number of randomly initialized parent perturbations. The generated offspring competes with their parent at every iteration, and the winner advances to the next step. The algorithm is finished as soon as the ground truth label probability is lower than 5%.

DeepFool [30] is a white box iterative approach in which the closest direction to the decision boundary is computed in every step. It is equivalent to finding the corresponding path to the orthogonal projection of the data point onto the affine hyperplane which separates the binary classes. The initial method for binary classifiers can be extended to a multi-class task by considering it as multiple one-versus-all binary classifications. After finding the optimal updates toward the decision boundary, the perturbation is added to the given image. The iterations continue with estimating the optimal perturbation and apply it to the perturbed image from the last step until the network decision changes.

Boundary attack is a reliable black-box attack proposed by Brendel et al. in [3]. The iterative algorithm already starts with an adversarial image and iteratively optimize the distance between the this image and the original image. It is aimed at finding an adversarial example with minimum distance from the original image.

3.2 Feature response estimation

The idea of visualizing CNNs through feature responses is to find out which region of the image leads to the final decision of the network. Computing feature responses enhances the interpretability of the classifier. In this paper, we use this visualization tool to track the effect of the adversarial attacks on a CNN's decision as well as to detect perturbed examples automatically.

Erhan et al. [9] used backpropagation for visualizing feature responses of CNNs. This is implemented by evaluating an arbitrary image in the forward pass, thereby retaining the values of activated neurons at the final convolutional layer, and backpropagating these activations to the original image. The feature response has higher intensities in the regions that cause larger values of activation in the network (see Figure 2). The information of max-pooling layers in the forward pass can further improve the quality of visualizations. Zeiler et al. [47] proposed to compute "switches", the position of maxima in all pooling regions, and then construct the feature response using transposed convolutional [8] layers.

Ultimately, Springenberg et al. [39] proposed a combination of both methods called guided backpropagation. In this approach, the information of "switches" (max-pooling spatial information) is kept, and the activations are propagated backwards with the guidance of the "switch" information. This method leads to the best performance in network innards visualization, therefore we use guided backpropagation for computing feature response maps in this paper.

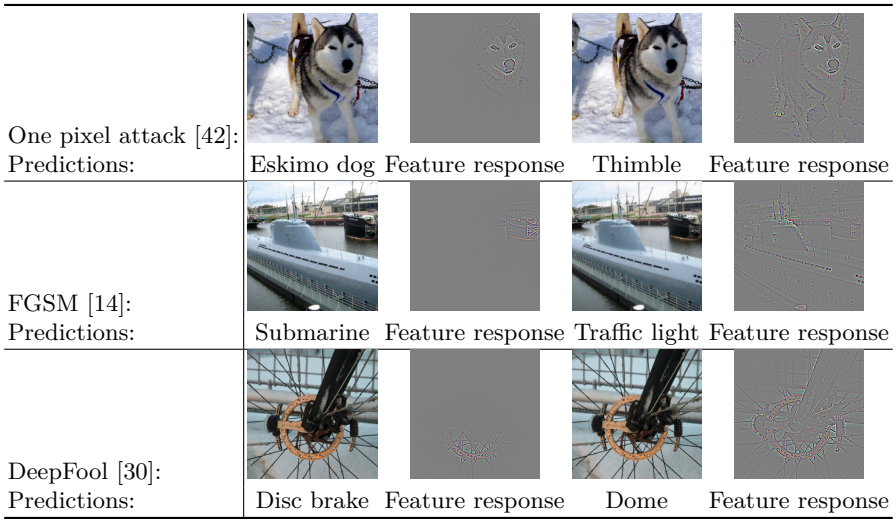


Figure 2: Effect of adversarial attacks on feature responses: original image and feature response (left), perturbed versions (right).

4 Human-interpretable detection of adversarial attacks

After reviewing the necessary background in the last Section, we will now present our work on tracing adversarial examples in feature response maps, which inspired a novel approach to automatic detection of adversarial perturbations in images. Using visual representations of the inner workings of neural network in this manner additionally provides a human expert guidance in developing deep convolutional networks with increased reliability and interpretability.

4.1 Tracing adversarial attacks in feature responses

The research question followed in this work is to obtain insight into the reasons behind misclassification of adversarial examples. Their effect in the feature response of a CNN is for example traced in Figure 2. The general phenomenon observed in all experiments is the broader feature response of adversarial examples. In contrast, Figure 2 demonstrates that the network looks at a smaller region of the image—is more focused—in case of not manipulated samples.

The adversarial images are visually very similar to the original ones. However, they are not correctly recognizable by deep CNNs. The original idea which triggered this study is that the focus of CNNs changes during an adversarial attack and lead to the incorrect decision. Conversely, the network makes the correct decision once it focuses on the right region of the image. Visualizing the feature response provides this and other interesting information regarding the decision making in neural networks: for instance, the image of the submarine in Figure 2 can be considered a good candidate for an adversarial attack since the CNN is making the decision based on an object in the background (see the feature response of the original submarine in Figure 2).

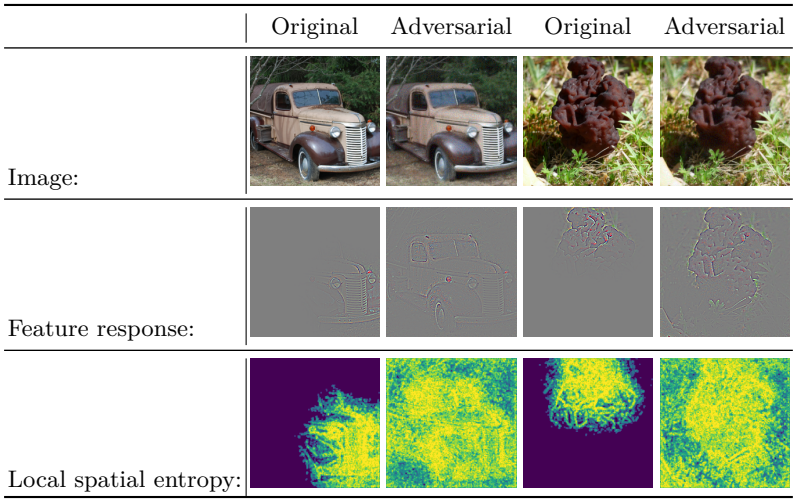


Figure 3: Input, feature response and local spatial entropy for clean and perturbed images, respectively.

4.2 Detecting adversarial attacks using spatial entropy

Experiments for tracing the effect of adversarial attacks on feature responses thus suggested that a CNN classifier focuses on a broader region of the input if it has been maliciously perturbed. Figure 2 demonstrates this connection for decision making in case of clean inputs compared with manipulated ones. The effect of adversarial manipulation is visible in the local spatial entropy of the gray-scale feature responses as well (see Figure 3). The feature responses are initially converted to gray scale images, and local spatial entropies are computed based on transformed feature responses as follows [4]:

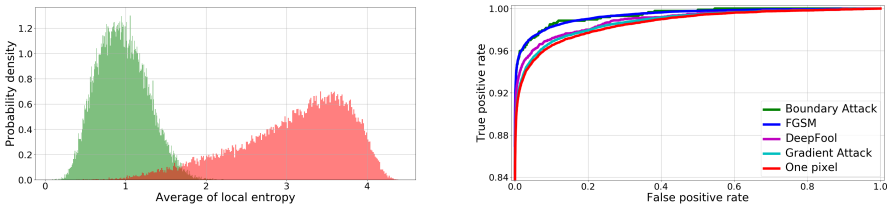
$$S_k = -\sum_i \sum_j \boldsymbol{h}_k(i,j) \log_2(h_k(i,j)) \tag{3}$$

where S_k is the local spatial entropy of a small part (patch) of the input image and $\boldsymbol{h}_k$ represents the normalized 2D histogram value of the k^{th} patch. The indices i and j scan through the height and width of the image patches. The patch size is 3×3 and the same as the filter size of the first layer of the used CNN (VGG19 [38]). The local spatial entropies of corresponding feature responses are presented in Figure 3, and their difference for clean and adversarial examples suggests a likely chance to detect perturbed images based on this feature.

Accordingly, we propose to use the average local spatial entropy of an image as the final single measure to decide whether an attack has occurred or not. The average local spatial entropy $\bar{S}$ is defined as:

$$\bar{S} = \frac{1}{K} \sum_k S_k \tag{4}$$

where K is the number of patches on the complete feature response and S_k shows the local spatial entropy as defined in Equation 3 and depicted in the last row



(a) Histogram (b) ROC curves

Figure 4: a) Distribution of average local spatial entropy in clean images (green) versus adversarial examples (red) as computed on the ImageNet validation set [36]. b) Receiver operating characteristic (ROC) curve of the performance of our detection algorithm on different attacks.

of Figure 3. Our detector makes the final decision by comparing the average local spatial entropy from Equation 4 with a selected threshold, i.e., we use this feature to measure the spatial complexity of an input image (feature response).

5 Experimental Results

To confirm the value of our final metric in Equation 4, we first perform experiments to visually compare the approximated distribution of the averaged local spatial entropy of feature responses in clean and perturbed images. We use the validation set of ImageNet [36] with more than $50,000$ images from $1,000$ classes and again the VGG19 CNN [38]. Perturbations for this experiment are computed only via the Fast Gradient Sign Attack (FGSM) method for computational reasons. Figure 4a) shows that the clean images are separable to a reasonable extent from perturbed examples although there is some overlap between the distributions.

Computing adversarial perturbations using evolutionary and iterative algorithms is demanding regarding time and computational resources. However, we would like to apply the proposed detector to a wide range of adversarial attacks. Therefore, we have drawn a number of images from the validation set of ImageNet for each attack and present the detection performance of our method in Figure 4. The selection of images is done sequentially by class and file name up to a total number of images per method that could be processed in a reasonable amount of time (see Table 1). We base our experiments on the FoolBox benchmarking implementation[3], running on a Pascal-based TitanX GPU.

Figure 4b presents the Receiver Operating Characteristics (ROC) of the proposed detector, and numerical evaluations are provided in Table 1. Our detection method performs better for gradient-based perturbations compared to the single pixel attack. Furthermore, Table 1 suggests that the best adversarial attack detection performance is achieved for FGSM and boundary attack perturbations, where the network confidences are changed the most. This observation suggests

[3] https://github.com/bethgelab/foolbox

Adversarial attack	#Images (run time [days])	Success rate	Ground truth confidence	Target class confidence	False positive rate 1%	5%	10%
FGSM [14]	50,014 (3)	0.925	0.022	0.588	0.954	0.973	0.982
Gradient attack	26,058 (8)	0.498	0.052	0.373	0.921	0.953	0.968
One pixel attack [42]	21,675 (14)	0.618	0.038	0.467	0.918	0.950	0.965
DeepFool [30]	5,012 (8)	0.602	0.042	0.455	0.935	0.961	0.971
Boundary attack [3]	1,195 (5)	0.939	0.023	0.586	0.953	0.972	0.985

Table 1: Numerical evaluation of detection performance on the three different adversarial attacks. Column two gives the amount of tested attacks and elapsed approx. run time. Success of an adversarial attack is given if a perturbation changes the prediction. Columns four and five show average confidence values of the true (ground truth) and wrong (target) class after successful attack, respectively. The last columns show detection rates for different false positive rates.

Method	Dataset	Network	Attack	Performance Recall	Precision	AUC
Uncertainty density estimation [10]	SVHN [19]	LeNet [21]	FGSM	-	-	0.890
Adaptive Noise Reduction [24]	ImageNet (4 classes)	CaffeNet	DeepFool	0.956	0.911	-
Feature Squeezing [45]	ImageNet-1000	VGG19	several attacks	0.859	0.917	0.942
Statistical Analysis [16]	MNIST	Self-designed	FGSM ($\epsilon = 0.3$)	0.999	0.940	-
Feature response (our approach)	ImageNet validation	VGG19	several attacks	0.970	0.920	0.990

Table 2: Performance of similar adversarial attack detection methods. The Area Under Curve (AUC) is the average value of all attacks in the third and last row.

that the proposed detector is more sensitive to attacks which are stronger in fooling the network (i.e., change the ground truth label and target class confidence more drastically). By using feature responses, we detect more than 91% of the perturbed samples with a low false positive rate (1%).

In general, it is difficult to directly compare different studies on attack detectors since they use a vast variety of neural network models, datasets, attacks and experimental setups. We present a short overview of the performances of current detection approaches in Table 2. Our approach is most similar to the methods of Liang et al. ([24]) and Xu et al. ([45]). The proposed detector in this paper outperforms both based on the presented results in their work; however, we cannot guarantee identical implementations and parameterizations of the used attacks (e.g., subset of used images, learning rates for optimization of perturbations). Similarly, adaptive noise reduction in the original publication [24] is applied to only four classes of the ImageNet dataset and defended a model based on CaffeNet, which differs from our experimental setup.

These results demonstrate the reality of adversarial attacks: improving the robustness of CNNs is necessary. However, we conducted further preliminary experiments on binary (cat versus dog [34]) and ternary (among three classes of cars [18]) classification tasks as proxies for the kind of few-class classifications settings frequently arising in practice. They suggest that it is considerably more challenging to find adversarial examples in such a setting without plenty of "other classes" to pick from for misclassification. Figure 6 illustrates these results.

Figure 6: Successful adversarial examples created by DeepFool [30] for binary and ternary classification tasks are only possible with notable visible perturbations.

6 Conclusion

In this paper, we have presented an approach to detect adversarial attacks based on human-interpretable feature response maps. We traced the effect of adversarial perturbations on the visual focus of the network in original images, which inspired a simple yet robust approach for automatic detection. This proposed method is based on thresholding the averaged local spatial entropy of the feature response maps and detects at least **91%** of various state-of-the-art adversarial attacks with a low false positive rate on the validation set of ImageNet. However, the results are not directly comparable with methods in the literature because of the diversity in the experimental setups as well as implementations of attacks.

Our results verify that feature responses are informative to detect specific cases of failure in deep CNNs. Furthermore, our detector can be used at the same time to increase the interpretability of neural network decisions, which is an increasingly important topic towards robust and reliable AI. Future work therefore will concentrate on developing reliable and interpretable image classification methods for practical use cases based on our preliminary results for binary and ternary classification.

References

1. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: A survey. arXiv preprint arXiv:1801.00553 (2018)
2. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
3. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248 (2017)
4. Chanwimaluang, T., Fan, G.: An efficient blood vessel detection algorithm for retinal images using local entropy thresholding. Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on **5** (2003)
5. Cireşan, D., Meier, U., Masci, J., Schmidhuber, J.: A committee of neural networks for traffic sign classification. In: IJCNN. pp. 1918–1921. IEEE (2011)
6. Cisse, M., Adi, Y., Neverova, N., Keshet, J.: Houdini: Fooling deep structured prediction models. arXiv preprint arXiv:1707.05373 (2017)
7. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)

8. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 (2016)
9. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. University of Montreal **1341**(3), 1 (2009)
10. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410 (2017)
11. Fletcher, R.: Practical methods of optimization. John Wiley & Sons (2013)
12. Fukushima, K., Miyake, S.: Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and cooperation in neural nets, pp. 267–285. Springer (1982)
13. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016)
14. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. ICLR (2015)
15. Goodman, B., Flaxman, S.: Eu regulations on algorithmic decision-making and a "right to explanation". In: ICML workshop on human interpretability in machine learning (WHI) (2016)
16. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.: On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280 (2017)
17. Gunning, D.: Explainable artificial intelligence (xai). Defense Advanced Research Projects Agency (DARPA) (2017)
18. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: ICCV Workshops (2013)
19. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
20. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. ICRL Workshop track (2016)
21. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4), 541–551 (1989)
22. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
23. Li, X., Li, F.: Adversarial examples detection in deep networks with convolutional filter statistics. arXiv preprint arXiv:1612.07767 (2016)
24. Liang, B., Li, H., Su, M., Li, X., Shi, W., Wang, X.: Detecting adversarial examples in deep networks with adaptive noise reduction. arXiv preprint arXiv:1705.08378 (2017)
25. Lu, J., Issaranon, T., Forsyth, D.: Safetynet: Detecting and rejecting adversarial examples robustly. arXiv preprint arXiv:1704.00103 (2017)
26. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: ACM SIGSAC Conference on Computer and Communications Security (2017)
27. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. ICLR (2017)
28. Metzen, J.H., Kumar, M.C., Brox, T., Fischer, V.: Universal adversarial perturbations against semantic image segmentation. arXiv preprint arXiv:1704.05712 (2017)
29. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. arXiv preprint arXiv:1610.08401 (2017)
30. Moosavi Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR (2016)

31. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill (2017). https://doi.org/10.23915/distill.00007
32. Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., Mordvintsev, A.: The building blocks of interpretability. Distill (2018). https://doi.org/10.23915/distill.00010
33. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277 (2016)
34. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: CVPR (2012)
35. Rauber, J., Brendel, W., Bethge, M.: Foolbox v0. 8.0: A python toolbox to benchmark the robustness of machine learning models. arXiv preprint arXiv:1707.04131 (2017)
36. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
37. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015)
39. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
40. Stadelmann, T., Tolkachev, V., Sick, B., Stampfli, J., Dürr, O.: Beyond imagenet - deep learning in industrial practice. In: Braschler, M., Stadelmann, T., Stockinger, K. (eds.) Applied Data Science - Lessons Learned for the Data-Driven Business. Springer (2018), to appear
41. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization **11**(4), 341–359 (1997)
42. Su, J., Vargas, D.V., Kouichi, S.: One pixel attack for fooling deep neural networks. arXiv preprint arXiv:1710.08864 (2017)
43. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. ICLR (2014)
44. Vellido, A., Martín-Guerrero, J.D., Lisboa, P.J.: Making machine learning models interpretable. In: ESANN. vol. 12, pp. 163–172 (2012)
45. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks (2018)
46. Xu, X., Chen, X., Liu, C., Rohrbach, A., Darell, T., Song, D.: Can you fool ai with adversarial examples on a visual turing test? arXiv preprint arXiv:1709.08693 (2017)
47. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
48. Zhu, J., Liao, S., Yi, D., Lei, Z., Li, S.Z.: Multi-label cnn based pedestrian attribute learning for soft biometrics. In: International Conference on Biometrics (ICB). IEEE (2015)