Detecting adversarial example attacks to deep neural networks

Fabio Carrara
ISTI-CNR
Pisa, Italy
fabio.carrara@isti.cnr.it

Fabrizio Falchi
ISTI-CNR
Pisa, Italy
fabrizio.falchi@isti.cnr.it

Roberto Caldelli
CNIT, MICC-University of Florence
Florence, Italy
roberto.caldelli@unifi.it

Giuseppe Amato
ISTI-CNR
Pisa, Italy
giuseppe.amato@isti.cnr.it

Roberta Fumarola
University of Pisa
Pisa, Italy

Rudy Becarelli
MICC-University of Florence
Florence, Italy
rudy.becarelli@unifi.it

ABSTRACT

Deep learning has recently become the state of the art in many computer vision applications and in image classification in particular. However, recent works have shown that it is quite easy to create adversarial examples, i.e., images intentionally created or modified to cause the deep neural network to make a mistake. They are like optical illusions for machines containing changes unnoticeable to the human eye. This represents a serious threat for machine learning methods. In this paper, we investigate the robustness of the representations learned by the fooled neural network, analyzing the activations of its hidden layers. Specifically, we tested scoring approaches used for kNN classification, in order to distinguishing between correctly classified authentic images and adversarial examples. The results show that hidden layers activations can be used to detect incorrect classifications caused by adversarial attacks.

CCS CONCEPTS

• **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; • **Computing methodologies** → **Neural networks**;

KEYWORDS

Adversarial images detection, Deep Convolutional Neural Network, Machine Learning Security

1 INTRODUCTION

Deep Neural Networks (DNNs) have recently led to significant improvement in many areas of machine learning. They are the state of the art in many vision and content-base multimedia indexing tasks such as classification [18, 31, 36], recognition [32], image tagging [21], video captioning [4], face verification [28, 30], content-based image retrieval [1, 16], super resolution [10], cross-media searching [7, 11], and image forensics [5, 39, 40].

Unfortunately, researchers have shown that machine learning models, including deep learning methods, are highly vulnerable to adversarial examples [12, 17, 25, 37]. An *adversarial example* is a malicious input sample typically created applying a small but intentional perturbation, such that the attacked model misclassifies it with high confidence [12]. In most of the cases, the difference between the original and perturbed image is imperceptible to a human observer. Moreover, adversarial examples created for a specific neural network have been shown to be able to fool different models with different architecture and/or trained on similar but different data [25, 37]. These properties are known as cross-model and cross-dataset generalization of adversarial examples and imply that adversarial examples pose a security risk even under a threat model where the attacker does not have access to the target's model definition, model parameters, or training set [19, 25].

Most of the effort of the research community in defending from adversarial attacks had gone into increasing the model robustness to adversarial examples via enhanced training strategies, such as adversarial training [12, 26] or defensive distillation [27]. However, studies have shown [25] that those techniques only make the generation of adversarial examples more difficult without solving the problem. A different, less studied, approach is to defend from adversarial attacks by distinguishing adversarial inputs from authentic inputs.

In this work, we present an approach to detect adversarial examples in deep neural networks, based on the analysis of activations of the neurons in hidden layers (often called deep features) of the neural network that is attacked. Being deep learning a subset of representation learning methods, we expect the learned representation to be more robust than the final classification to adversarial examples. Moreover, adversarial images are generated in order to look similar to humans and deep features have shown impressive results in visual similarity related tasks such as content-based image retrieval [13, 33]. The results reported in this paper show that, given an input image, searching for similar deep features among
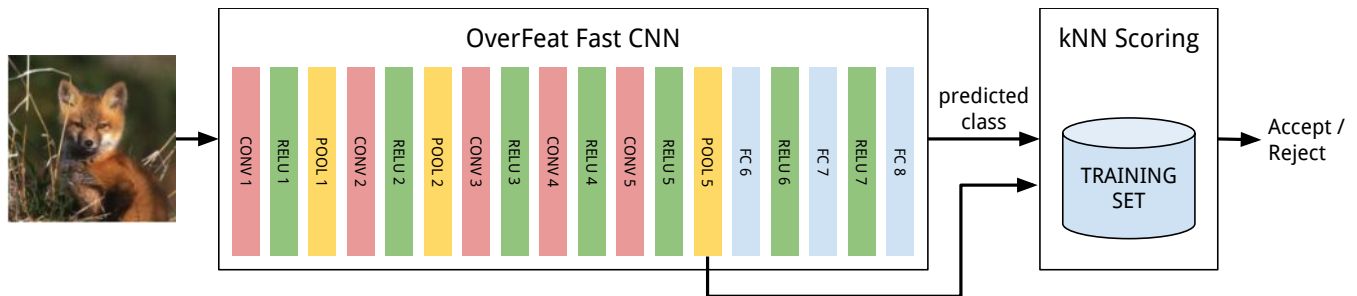
Figure 1: Overview of our detection approach. The input image is classified by the CNN, but we consider the classification valid only if the kNN score of the predicted class based on deep features (*pool5*) is above a certain threshold.

the images used for training, allows to predict the correctness of the classification produced by a DNN. In particular, we use traditional kNN classifiers scoring approaches as a measure the confidence of the classification given by the DNN (see Figure 1). The experiments show that we are able to filter out many adversarial examples, while retaining most of the correctly classified authentic images. The choice of the discriminative threshold is a trade-off between accepted false positives (FP) and true positives (TP), where positive means non-adversarial.

The rest of the paper is structured as follows. Section 2 reviews the most relevant works in the field of adversarial attacks and their analysis. Section 3 provides background knowledge to the reader about DNNs, image representations (known as deep features), and adversarial generation. In section 4 our approach is presented, while in section 5 we describe the experimental settings we used to validate it. Finally, section 6 concludes the paper and presents some future research directions.

2 RELATED WORK

2.1 Generation of Adversarial Examples

Szegedy et al. [37] firstly defined an *adversarial example* as the smallest perturbed image that induces a classifier to change prediction with respect to the original one. They successfully generated adversarial examples through the use of the box-constrained Limited-memory approximation of Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm, and they proved that adversarial examples exhibit cross-model and cross-training set generalization properties. To overcome to the high computational cost of the L-BFGS approach, Goodfellow et al. [12] proposed the Fast Gradient Sign (FGS) method, which derives adversarial perturbations from the gradient of the loss function with respect to the input image, that can be efficiently computed by backpropagation. In [24], Nguyen et al. used evolutionary algorithms and gradient ascent optimizations to produce fooling images which are unrecognizable to human eyes but are classified with high confidence by DNNs. Papernot et al. [26] used forward derivatives to compute adversarial saliency maps that show which input feature have to be increased or decreased to produce the maximum perturbation of the last classification layer towards a chosen adversarial class. In [23], Moosavi et al. presented an algorithm to find image-agnostic (universal) adversarial perturbations for a given trained model, that are able fool the classifier with high probability when added to any input.

2.2 Defense Strategies for Adversarial Attacks

Different kinds of defenses against adversarial attacks have been proposed. Fast adversarial generation methods (such as FGS) enable adversarial training, that is the inclusion in the training set of adversarial examples generated on-the-fly in the training loop. Adversarial training allows the network to better generalize and to increase its robustness to this kind of attacks. However, easily optimizable models, such as models with non-saturating linear activations, can be easily fooled due to their overly confident linear responses to points that not occur in the training data distribution [12]. In [14], the authors found that denoising autoencoders can remove substantial amounts of the adversarial noise. However, when stacking the autoencoders with the original neural network, the resulting network can again be attacked by new adversarial examples with even smaller distortion. Thus, the authors proposed Deep Contractive Network, a model with an end-to-end training procedure that includes a smoothness penalty. Similarly, in [27] a two-phase training process known as *distillation* is used to increase the robustness of a model to small adversarial perturbations by smoothing the model surface around training points and vanishing the gradient in the directions an attacker would exploit. Still, attackers can find potential adversarial images using a non-distilled substitute model. Papernot et al. [25] showed that successfully attacks are possible even if the attacker does not have direct access to the model weights or architecture. In fact, the authors successfully performed adversarial attacks to remotely hosted models, and Kurakin et al. [19] also showed that attacks in physical scenarios, such as feeding a model with a printout adversarial example through a digital camera, are possible and effective.

Detection of adversarial examples is still an open problem [26]. The most related work to ours is from Metzen et al. [22], that proposed to add a parallel branch to the classifier and train it to detect whether the input is an adversarial example. However, the proposed branch is still vulnerable to adversarial attacks, and a more complicate adversarial training procedure is needed to increase the robustness of the whole system.

3 BACKGROUND

3.1 Deep Learning and Features

Deep learning methods are "representation-learning methods with multiple levels of representation, obtained by composing simple

but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level" [20].

Starting from 2012, deep learning has become state-of-the-art in image classification given the excellent results in ILSVRC challenges based on ImageNet [15, 18, 31, 34, 36]. In the context of Content-Based Image Retrieval, deep learning architectures are used to generate high level features. The relevance of the internal representation learned by the neural network during training have been proved by many recent works [2, 6, 9, 20]. In particular, the activation produced by an image within the intermediate layers of a deep convolutional neural network can be used as a high-level descriptor of the image visual content [2, 3, 8, 29, 31].

In this work, we employed the image representations extracted using OverFeat [31], a well-known and successful deep convolutional network architecture that have been studied for the analysis of adversarial attacks to convolutional neural networks [38], and for which implementations of adversarial generation algorithms are publicly available (see Section 5). Specifically, we used the Fast Over-Feat network pre-trained on ImageNet (whose code and weights are publicly available at https://github.com/sermanet/OverFeat), and we selected the activations of the *pool5* layer as deep features for images.

3.2 Adversarial Generation

In this subsection we provide a brief description of the two approaches we used in our work to generate adversarial images.

Box Constrained L-BFGS [37, 38]. Given an input image x and a DNN classifier $y = f(x)$, an adversarial example is generated finding the smallest distortion η such that $x' = x + \eta$ is misclassified by the target model, that is $f(x + \eta) \neq y$. The adversarial perturbation η is modeled as the solution of the following optimization problem:

$$\begin{aligned} \underset{\eta}{\text{minimize}} \quad & ||\eta|| + C \cdot H(y, y^A) \\ \text{subject to} \quad & L <= x + \eta <= U, \\ & y = f(x + \eta) \end{aligned} \quad (1)$$

where L and U are respectively lower and upper bound of pixel values, f is the attacked classifier, $H(y, y^A)$ is the cross-entropy loss computed between the output class probability distribution y and the target adversarial distribution y^A (which assigns probability 1 to the adversarial label and 0 to the remaining ones). The parameter C controls the trade-off between the magnitude of η and its fooling power. An adversarial perturbation is found by solving (1) using the box-contrained L-BFGS optimization algorithm. A first feasible value of C is found with a coarse grid search and then tuned with a binary search.

Fast Gradient Sign [12]. In the Fast Gradient Sign method, the adversarial perturbation is proportional to the sign of the gradient back-propagated from the output to the input layer. Mathematically speaking, let θ be the parameters of a model, x the input to the model, y the targets (the desired output) associated with x, and $J(\theta, x, y)$ the cost function used to train the neural network. The cost function can be linearized around the current value of θ, obtaining

an optimal max-norm constrained perturbation:

$$\eta = \epsilon \cdot sign(\nabla_x J(\theta, x, y))$$

Note that the gradient can be computed easier using backpropagation. The adversarial input is given by $x' = x + \eta$.

4 DETECTING ADVERSARIAL EXAMPLES

In this work, we propose to detect adversarial examples analyzing the representation learned in the hidden layers (deep features) of the fooled convolutional neural networks. Being deep learning a subset of representation learning methods, we expect the learned representation to be more robust than the final classification to adversarial examples. The recent renaissance of neural networks is due to the ability of learning powerful representations that can be used for classification but also for many other tasks such as recognition [32], face verification [30], content-based image retrieval [16], super resolution [10], cross-media searching [7, 11], etc. There are two reasons why deep features should be more robust: first, the adversarial generation algorithms are not meant to fool the representation itself but only the final classification; second, adversarial images are generated in order to look similar to authentic ones for humans, and deep features have shown impressive results in visual similarity related tasks such as content-based image retrieval [13, 33]. We decided to test kNN classifiers score assignments because they rely on those similarity among the representations.

In particular, we perform a kNN similarity search among the deep features obtained from the images used for training using as query a given image classified by the DNN. We then use the score assigned by a kNN classifier to the class predicted by the DNN as a measure of confidence of the classification. Please note, we do not rely on the classification produced by the kNN classifier, but we only use the score assigned to the class predicted by the DNN as a measure of confidence.

Given a set of labeled images $X = \{(x_i, c_i)\}$ where x_i is an image and c_i is its class label, a kNN classifier assigns labels to an unknown image q considering the ordered results of its k nearest neighbors $NN(q, k) = \{(x_1, c_1) \ldots (x_k, c_k)\}$, obtained performing a kNN search over X for a predefined distance function $d(x, y)$ between any two images. We define the distance function as $d(x, y) = ||\phi(x) - \phi(y)||_2$, where $\phi(x)$ is the deep feature extracted from the image x using the DNN, and $|| \cdot ||_2$ is the L2 norm. A score $s(q, c)$ is assigned to every class c found in the retrieved nearest neighbors of q as follows:

$$s(q, c) = \frac{\sum_{i=1}^{k} w_i \mathbb{1}\{c_i = c\}}{\sum_{i=1}^{k} w_i} \quad (2)$$

where q is a query image, c is the class for which we are computing the score, c_i is the groundtruth class of the i-th result in $NN(q, k)$, w_i the weight assigned to the same i-th result, and $\mathbb{1}\{c_i = c\}$ has value 1 if $c_i = c$, 0 otherwise. In Table 1, we report w_i assignments for famous variants of kNN classifiers.

Please note that our strategy does not make use of the class predicted by the kNN classifier. Rather, to detect whether an input image is an adversarial example, we first use the DNN to predict the class, then we use the kNN classifier to obtain a score for the class predicted by the DNN. The intuition is that while it is unlikely that a class correctly predicted by the DNN has the highest kNN

kNN	Weighted kNN	d-Weighted kNN
$w_i = 1$	$w_i = \dfrac{1}{i}$	$w_i = \dfrac{1}{d(q, x_i)^2}$

Table 1: Weighting functions for the various kNN classifiers

score among the scores of all the classes, it is implausible that a correct classification has a very low score.

More formally, let x be the input image and c_x the class predicted by the DNN with a forward computation: $c_x = f(x)$. The kNN classifier is used to compute the score $s(x, c_x)$ of the class c_x predicted by the DNN. We decide that the classification is reliable, i.e. it is not an adversarial, if the score $s(x, c_x)$ is above a predefined threshold. The score $s(x, c_x)$ is, basically, a measure of the confidence of the classification given by the DNN. As anticipated, the choice of the score threshold is a trade-off between false positives (FP) and true positives (TP), where FP are the adversarial examples (negatives) not detected using the specific threshold (false) and TP is the rate of correctly classified authentic images (positives) successfully identified (true). Please note that we do not rely on additional models or data other than the fooled DNN and its original training set for the extraction of deep features or for the detection task.

As better detailed in next section, we used the OverFeat fast network [31] pre-trained on ImageNet ILSVRC2012 as DNN considering *pool5* as deep features.

5 EXPERIMENTAL SETTINGS

In this section, we describe the method and the performance measures we used to evaluate the proposed adversarial detection approach. Generated adversarials and other resources have been made public available on the paper web page [1]. Our method has been evaluated as a binary classification of the correctness of the prediction given by a DNN, in which a positive outcome means that the prediction given by the DNN is trustful, while a negative outcome indicates that the prediction given by the DNN is spurious and have to be discarded. As reported in the previous section, we used OverFeat fast network [31] for our experiments given that it is the pre-trained network on ILSVRC2012 used in the papers in which L-BFGS and Fast Gradient Sign (see Section 3.2) were presented. As reported in Section 4, our approach performs a kNN similarity search over the images that were used for training the attacked DNN.

The images used for our experiments are taken from the ILSVRC2012 validation set. In particular, we selected two subsets of images based on the classification results. The first set is composed by randomly selecting a correctly classified image for each of the $1,000$ ILSVRC classes, while the second set is composed by randomly selecting a wrongly classified image (for which the network has given a wrong prediction) for each of the same $1,000$ classes. We could not select a wrongly classified image in the class coded "n12057211" (yellow lady's slipper, yellow lady-slipper, Cypripedium calceolus, Cypripedium parviflorum) because all the instances of this class in the validation set are correctly classified by OverFeat. Thus, the two sets respectively count $1,000$ and 999 images. We

named those subsets respectively *Authentic* and *Authentic Errors*, where 'authentic' stands for non-adversarial images.

For each image in the *Authentic* subset, we generated two adversarial images using both box constrained L-BFGS[2] and FGS[3] algorithms. For both methods, we used the default parameters in every generation and we randomly selected the target class, that is the class we fool the network to predict. We observed that L-BFGS algorithm failed to generate 8 adversarial images, in the sense that the class prediction of the generated adversarial image was the same of the original image. Those failures in the generation process could be avoided tuning the parameters of the algorithm for each input, but for sake of simplicity we discarded the failed adversarial examples, ending up with two sets of adversarial images respectively composed by 1000 images generated by FGS, and 992 images obtained with L-BFGS. The generated adversarial images are made publicly available[4] to make easier to reproduce the experiments.

We extracted the activations of the *pool5* intermediate layer of the pre-trained OverFeat fast network [31] from the following sets of images: *Authentic*, *Authentic Errors*, *L-BFGS Adversarial*, *FGS Adversarial* and *ILSVRC2012 train set*. Activations of *pool5* are composed by 1024 6x6 feature maps. Following [35], we applied global average pooling (GAP) to the *pool5* feature maps, which acts as a structural regularizer, obtaining an image representation of 1024 floats. For the kNN classifier, we used the features extracted from ILSVRC2012 train set as labeled set X, and we defined the distance function $d(q, x)$ as the euclidean distance between the extracted features. We chose $k = 1,000$ to have a number of nearest neighbors of the same order of magnitude of the number of images per class in the labeled set. We tested also feature L2 normalization and dimensionality reduction using PCA+Whitening with 256 dimensionality.

Given an input image x, we compute the kNN score $s(x, c)$ for the class $c = f(x)$ predicted by OverFeat, and we discard this classification if the score is below a certain threshold. We computed the kNN score for each image in the *FGS Adversarial*, *L-BFGS Adversarial* and *Authentic Errors* image sets, and for each set we measure the ability to detect an adversarial input as the performance of a binary classification problem ('trustful' / 'spurious' classification).

5.1 Results

In Table 2, we report the detection accuracy of our proposed approach for different settings. Accuracy of the binary 'trustful' / 'spurious' classification is evaluated in the equal error rate (EER) setting, that is when we choose a threshold yielding equal false positive and false negative rates. The best results were obtained processing the deep features using PCA and Whitening. The three scoring approaches considered revealed similar performance with DW-kNN and W-kNN more effective in detecting L-BFGS and FGS, respectively.

In Figure 2, we report the Receiver operating characteristic (ROC) curves of the DW-kNN and W-kNN scoring approaches on adversarial images and errors. The curves illustrate the performance of the proposed binary classifier when varying the threshold on the score $s(x, f(x))$. The results show that W-kNN is generally preferable


Processing	Score	L-BFGS	FGS	Aggr.	Errors
None	kNN	70.7	69.9	70.3	58.1
	W-kNN	71.2	70.8	71.0	59.6
	DW-kNN	71.0	69.9	70.4	58.6
L2Norm	kNN	79.2	74.3	76.7	60.7
	W-kNN	81.4	76.4	78.9	62.9
	DW-kNN	81.7	76.6	79.1	61.6
PCA + Whiten	kNN	86.4	83.4	84.9	62.9
	W-kNN	85.9	**83.8**	84.8	**65.0**
	DW-kNN	**86.5**	83.5	**85.0**	63.6

Table 2: Detection accuracy in the equal error rate (EER) score threshold setting for various activation layers, score functions, and features processing. We report results for both type of adversarial (L-BFGS and FGS) and the aggregated accuracy (Aggr.). In the last column, we report the detection rate of erroneous classifications not due to adversarial examples.
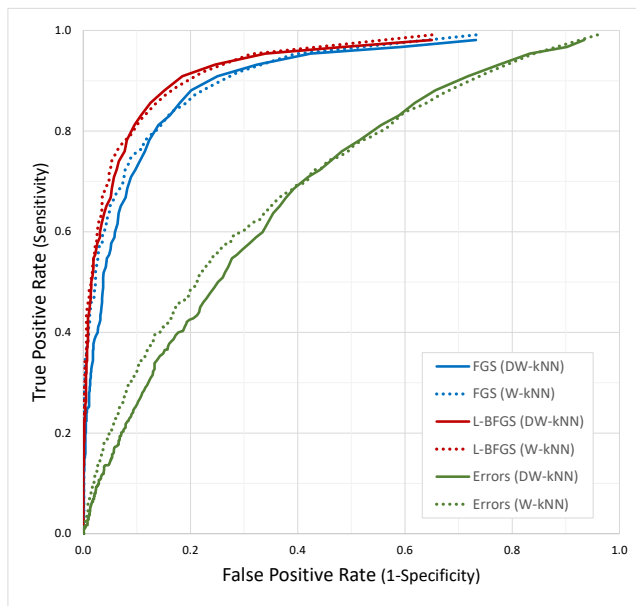
Figure 2: Receiver operating characteristic (ROC) curves of the binary classification ('prediction is right' or 'prediction is wrong') for the various types of images. The curves are obtained varying the discrimination threshold on the score assigned by the DW-kNN classifier to the class predicted by the CNN.

when low FP rate are requested while DW-kNN is more effective when FP and TP are comparable.

In the following, we focus on the DW-kNN score. In Figure 3, we report the true positive (for correctly classified authentic images) and both false positive (for adversarial images and authentic errors) rates distributions as a function of the discriminant threshold
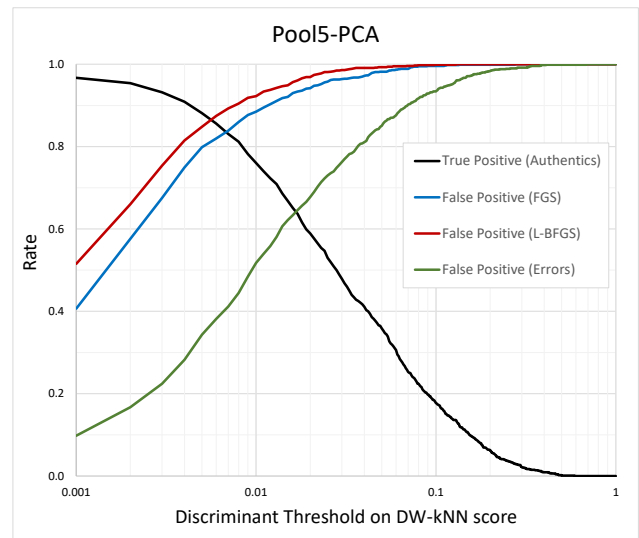
Figure 3: True positive and false positive rates using as discrimination threshold between correctly and incorrectly classified images the score assigned by the DW-kNN classifier to the class predicted by the CNN. The *pool5* layer has been used as feature with PCA and Whitening processing.

applied on the score $s(x, f(x))$. The results show that using very low discrimination score values (about 0.002), it is possible to correctly filter out more than 50% of the adversarial examples created by L-BFGS and more than 40% of the ones created by FGS, while retaining more than 98% of authentic images correctly classified. As a positive side-effect, we also discard around 10% of images the DNN would misclassify. The low threshold values reveal that while the DW-kNN score would not be effective in classifying the images, values below 0.003 are unlikely for authentic images.

The same results can be seen from the score densities reported in Figure 4, in which we can observe a distinction between the score densities of adversarial images and the ones of authentic images. Some simple metrics on those densities (such as the mean) could be computed on-line in the system hosting the model to isolate a particular source of adversarial examples, hence denying the access to the service to an attacker.

Finally in Table 3, we report some of the easiest and most difficult adversarial examples to detect, together with the nearest neighbor image in the kNN's labeled set and the score $s(x, f(x))$. We observed that higher kNN scores (which correspond to difficult adversarial to detect) usually reflects inter-class visual similarities that are independent from the adversarial nature of the input image.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach to detect adversarial examples crafted for fooling deep neural network classifiers. We inspect the activations of intermediate layers for both adversarial and authentic inputs, and we defined a classification confidence score based on kNN similarity searching among the images used for training. The proposed approach allows to filter about 80% of
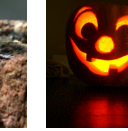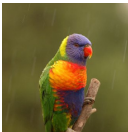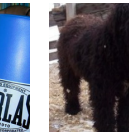
Method	L-BFGS	L-BFGS	L-BFGS	L-BFGS	...	FGS	FGS	FGS	FGS
Actual Class	basenji	Weimaraner	wine bottle	kit fox, Vulpes macrotis	...	agama	jack-o'-lantern	knee pad	agaric
Adv. Image					...				
DNN Prediction	Arctic fox, white fox, Alopex lagopus	lorikeet	pop bottle, soda bottle	chow, chow chow	...	worm fence, snake fence, snake-rail fence, ...	cellular telephone, cellular phone, cellphone, ...	punching bag, punch bag, ...	Bouvier des Flandres, Bouviers des Flandres
Predicted Nearest Neighbor					...				
kNN score s	0.13	0.10	0.08	0.08		0.00	0.00	0.00	0.00

Table 3: Adversarial images to which our best approach (*pool5*+PCA+DW-kNN) assigns the highest and lowest authenticity score s. From top to bottom, rows respectively report: the generation method, the original class of an input, its adversarial version, the class predicted by the DNN, the nearest neighbor image (in terms of L2 distance between average-pooled *pool5* activations) belonging to the predicted class, and the DW-kNN score s for the predicted class. A low score indicates that the adversarial is correctly detected while a high score means that our approach is wrongly confident about the prediction of the CNN. The results show that high scoring adversarials examples often share some common visual aspects and semantic with the predicted (adversarial) class, resulting in a more challenging detection.
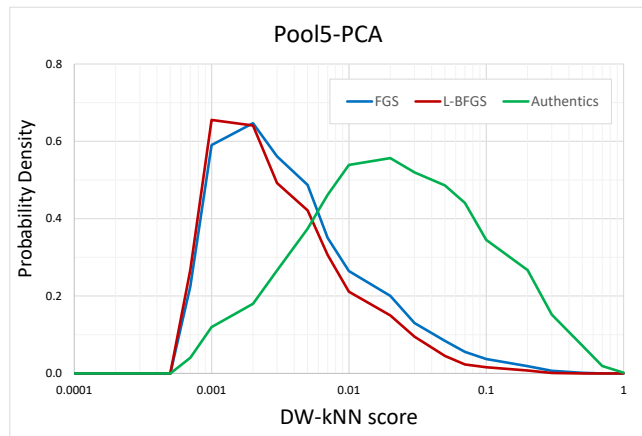

Figure 4: Density of the DW-kNN scores for both adversarial and authentic images. We report densities of scores using *pool5* as feature and PCA+Whitening as processing.

adversarial examples retaining more than 90% of the correctly classified authentic images (see Figure 3). Moreover, some examples are suggesting that hard adversarial examples are the ones for which actual and target classes are similar or have similar visual patterns.

In future work, we plan to extend our analysis to other model architectures and other adversarial examples generated with different algorithms.

ACKNOWLEDGMENTS

REFERENCES

[1] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Carlo Meghini, and Claudio Vairo. 2017. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications* 72 (2017), 327–334.

[2] Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, and Fausto Rabitti. 2016. YFCC100M-HNfc6: A Large-Scale Deep Features Benchmark for Similarity Search. In *International Conference on Similarity Search and Applications*. Springer, 196–209.

[3] Giuseppe Amato, Fabrizio Falchi, and Lucia Vadicamo. 2016. Visual recognition of ancient inscriptions using convolutional neural network and fisher vector. *Journal on Computing and Cultural Heritage (JOCCH)* 9, 4 (2016), 21.

[4] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2016. Hierarchical Boundary-Aware Neural Encoder for Video Captioning. *arXiv preprint arXiv:1611.09312* (2016).

[5] B. Bayar and M. C. Stamm. 2016. A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer. In *4th ACM Workshop on Information Hiding and Multimedia Security*. 5–10.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (Aug. 2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[7] Fabio Carrara, Andrea Esuli, Tiziano Fagni, Fabrizio Falchi, and Alejandro Moreo Fernández. 2016. Picture it in your mind: Generating high level visual representations from textual descriptions. *arXiv preprint arXiv:1606.07287* (2016).

[8] Vijay Chandrasekhar, Jie Lin, Olivier Morère, Hanlin Goh, and Antoine Veillard. 2015. A practical guide to cnns and fisher vectors for image instance retrieval. *arXiv preprint arXiv:1508.02496* (2015).

[9] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.. In *Icml*, Vol. 32. 647–655.

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2016. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* 38, 2 (2016), 295–307.

[11] Jianfeng Dong, Xirong Li, and Cees GM Snoek. 2016. Word2VisualVec: Cross-media retrieval by visual feature prediction. *arXiv preprint arXiv:1604.06838* (2016).

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[13] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. 2016. Deep image retrieval: Learning global representations for image search. In *European Conference on Computer Vision.* Springer, 241–257.

[14] Shixiang Gu and Luca Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068* (2014).

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 770–778.

[16] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[17] Erica Klarreich. 2016. Learning Securely. *Commun. ACM* 59, 11 (Oct. 2016), 12–14. https://doi.org/10.1145/2994577

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[19] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[21] Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees GM Snoek, and Alberto Del Bimbo. 2016. Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Computing Surveys (CSUR)* 49, 1 (2016), 14.

[22] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On Detecting Adversarial Perturbations. *arXiv preprint arXiv:1702.04267* (2017).

[23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2016. Universal adversarial perturbations. *arXiv preprint arXiv:1610.08401* (2016).

[24] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[25] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2016. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint arXiv:1602.02697* (2016).

[26] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on.* IEEE, 372–387.

[27] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on.* IEEE, 582–597.

[28] O. M. Parkhi, A. Vedaldi, and A. Zisserman. 2015. Deep Face Recognition. In *British Machine Vision Conference.*

[29] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on.* IEEE, 512–519.

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 815–823.

[31] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* (2013).

[32] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.*

[33] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.* 806–813.

[34] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[35] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1–9.

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[38] Pedro Tabacof and Eduardo Valle. 2016. Exploring the space of adversarial images. In *Neural Networks (IJCNN), 2016 International Joint Conference on.* IEEE, 426–433.

[39] A. Tuama, F. Comby, and M. Chaumont. 2016. Camera model identification with the use of deep convolutional neural networks. In *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on.*

[40] Z. Ying, J. Goha, L. Wina, and V. Thinga. 2016. Image Region Forgery Detection: A Deep Learning Approach. In *Singapore Cyber-Security Conference (SG-CRC).* 1–11.