Attack to Explain Deep Representation

Mohammad A. A. K. Jalwana Naveed Akhtar Mohammed Bennamoun Ajmal Mian
Computer Science and Software Engineering,
The University of Western Australia.
{mohammad.jalwana@research., naveed.akhtar@, mohammed.bennamoun@, ajmal.mian}@uwa.edu.au

Abstract

Deep visual models are susceptible to extremely low magnitude perturbations to input images. Though carefully crafted, the perturbation patterns generally appear noisy, yet they are able to perform controlled manipulation of model predictions. This observation is used to argue that deep representation is misaligned with human perception. This paper counter-argues and proposes the first attack on deep learning that aims at explaining the learned representation instead of fooling it. By extending the input domain of the manipulative signal and employing a model faithful channelling, we iteratively accumulate adversarial perturbations for a deep model. The accumulated signal gradually manifests itself as a collection of visually salient features of the target label (in model fooling), casting adversarial perturbations as primitive features of the target label. Our attack provides the first demonstration of systematically computing perturbations for adversarially non-robust classifiers that comprise salient visual features of objects. We leverage the model explaining character of our algorithm to perform image generation, inpainting and interactive image manipulation by attacking adversarially robust classifiers. The visually appealing results across these applications demonstrate the utility of our attack (and perturbations in general) beyond model fooling.

1. Introduction

Deep visual models have provided breakthroughs in numerous computer vision tasks, including image classification [24, 43], object detection [37, 38], semantic segmentation [27, 9] and image captioning [49]. However, despite their impressive performance, deep models are found vulnerable to adversarial perturbations to inputs [45]. These perturbations are weak additive signals that manipulate model predictions while remaining imperceptible to the human visual system. The intriguing susceptibility of deep models to adversarial perturbations is currently being actively investigated by the research community [2].

Dictated by the original 'adversarial' perspective [45],

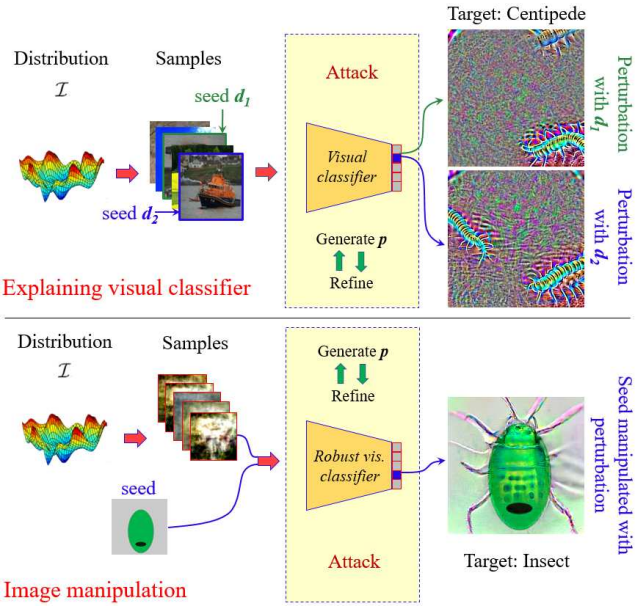

Figure 1. **Top**: Using an image distribution, our attack iteratively generates and refines a perturbation $\boldsymbol{p}$ for a standard deep visual classifier (VGG-16 here) that extracts geometric patterns deemed salient visual features of a label by the classifier. **Bottom**: Applying our attack to adversarially robust classifiers (ResNet-50 here) enables visually appealing interactive image manipulation (here), image generation (Fig. 6) and inpainting (Fig. 7).

research in this direction has taken a natural bicephalous approach. One stream of works aims at generating perturbations with modest visual perceptibility and high transferability to fool known and unknown models [16, 25, 13, 42, 11, 30]. While the other focuses on defending the models against such perturbations [50, 36, 26, 1, 34]. There are very few exceptions that deviate from the 'adversarial' brand of perturbations and cast these signals as a fooling tool for deep learning. Santurkar et al. [41] presented a notable contribution along this line by using perturbations for image synthesis with adversarially robust networks.

Investigating adversarial perturbations, Ilyas et al. [19] claimed that the existing large datasets (e.g. ImageNet [12]) admit to brittle yet highly predictive features that remain

imperceptible to humans. It is argued that deep visual models rely on these non-robust features for high accuracy, which also makes them susceptible to adversarial perturbations. Reliance of deep models on these 'apparently' incomprehensible features is also argued to indicate a misalignment between deep visual representation and human perception [14]. To remove this misalignment, Engstorm et al. [14] proposed to learn deep models under robust optimization framework. However, this entails a significant performance loss for the original model and a drastic increase in the computational complexity of model induction.

It is paradoxical that a representation misaligned with human perception still performs human-meaningful visual tasks with high accuracy. To investigate this phenomenon, we delve deep into the composition of perturbation signals with an alternate objective of model explanation instead of model fooling. We discover that under appropriate conditions, adversarial perturbations eventually emerge as salient visual features of the target label even for the non-robust models, see Fig. 1 (top). Within the context of adversarial perturbations, this observation drastically weakens the argument of misalignment between human perception and deep representation. Rather, it places adversarial perturbations as human-meaningful geometric features of the target label, albeit in a primitive and subtle form.

Our perturbation estimation algorithm stochastically maximizes the prediction probability of an image distribution's perturbed samples for a given target label. Anchored by a seed image, the maximization takes place by iteratively stepping in the Expected gradient direction of the classifier's loss surface *w.r.t.* the input samples. The optimization is guided by gradient moments and adjusting the step direction to achieve the ultimate objective more efficiently. We further channel the perturbation signal to focus more on its regions that cause high activity of the neurons in the deeper layers of the classifier. This refinement is purely based on the intermediate perturbations computed by our algorithm, which makes our technique model faithful - a desirable property for model explanation [14].

Besides explaining deep models in terms of salient visual features for class labels and highlighting the alignment of deep representation with human perception, our attack naturally suits to the low-level vision tasks of e.g. image generation, inpainting and interactive image manipulation using 'classifiers' [41]. We affirm the utility of our technique (and perturbations in general) beyond the adversarial objective by achieving significant visual improvements for these tasks over [41]. The major contributions of this work are summarized as:

- We propose the first attack on deep learning with input perturbation that explains a model instead of fooling it.
- By manifesting salient visual features of class labels in perturbations for 'non-robust' models, we drastically weaken the argument that deep representation is misaligned with the human perception.
- We demonstrate visually appealing image generation, inpainting and interactive image manipulation by attacking robust classifiers. Our results affirm the utility of perturbations beyond model fooling.

2. Related work

Adversarial perturbations are being actively investigated along the lines of attacking deep models and defending them against the adversarial attacks [2]. We first discuss the key contributions along these lines and then focus on the non-adversarial perspective of input perturbations.

Adversarial attacks: Additive adversarial perturbations that can arbitrarily alter the decisions of deep models made their first appearance in the seminal work of Szegedy et al. [45]. This discovery fueled the development of numerous techniques to attack deep visual models. Goodfellow et al. [16] devised the Fast Gradient Sign Method (FGSM) to craft adversarial perturbations in a single gradient ascent step over the model's loss surface for the input. Later, Kurakin et al. [25] advanced this scheme by introducing a mutli-step version called Iterative FGSM (I-FGSM). Further instances of the follow-up iterative algorithms for adversarial attacks include Momentum I-FGSM (MI-FGSM) [13], Diverse Input I-FGSM (DI^2-FGSM) [51] and Variance-Reduced I-FGSM (vr-IGSM) [48] etc.

The above-mentioned algorithms and other recent works [30, 42, 39, 11, 52, 15] compute image-specific adversarial perturbations. These perturbations appear noise to humans but completely fool the models. Moosavi-Dezfooli et al. [29] first demonstrated the possibility of fooling deep models simultaneously on a large number of images with Universal Adversarial Perturbations. Later, [33, 5, 22, 31] also devised techniques for computing effective universal perturbations. The pervasive susceptibility of deep models to adversarial perturbations is seen as a serious threat to practical deep learning [2] - an idea currently fueling the very high level of research activity in this area.

Adversarial defenses: On the flip side, numerous techniques have also surfaced to counter the adversarial attacks [20, 34, 36, 50, 44, 35, 1, 26]. These techniques aim at protecting deep model against both image-specific [34] and universal perturbations [1]. This is commonly done by either detecting the perturbation in an input image, or diluting the adversarial effects of perturbation signals by modifying the model or the input itself. Nevertheless, Carlini et al. [7, 6, 8] and later Athalye et al. [3] demonstrated that it is often possible to break the adversarial defenses by stronger adversarial attacks.

Non-adversarial perspective: Currently, there are also contributions in the literature (albeit very few) that hint towards the utility of perturbations beyond model fooling. For

instance, Tsipras et al. [46] observed the presence of salient visual features of the target class in the perturbation signals that fool 'adversarially robust' models. A similar observation is made by Woods et al. [47] for the models robustified with regularized gradients. Existence of salient visual features in perturbations indicate the potential of these signals in model explanation [28, 47]. However, their manifestation *uniquely* in the case of robustified models is interpreted as a misalignment between (non-robust) deep representation and the human perception [14, 46]. Potentially, the re-alignment is only achievable by adversarially robustifying the models at a serious cost of performance loss and amplified computational complexity [14, 46].

3. Attacking to explain

Let $\boldsymbol{I} \in \mathbb{R}^m$ be a sample of a distribution $\mathcal{I}$ over the natural images and $\mathcal{K}(\boldsymbol{I})$ be a deep visual classification model that maps $\boldsymbol{I}$ to its correct label ℓ_{true}. The common aim of generating perturbations in adversarial settings is to compute $\boldsymbol{p} \in \mathbb{R}^m$ that satisfies the constraint

$$\mathcal{K}(\boldsymbol{I}+\boldsymbol{p}) \rightarrow \ell_{\text{target}} \;\; \text{s.t. } \ell_{\text{target}} \neq \ell_{\text{true}}, ||\boldsymbol{p}||_p \leq \eta, \qquad (1)$$

where $||.||_p$ denotes the ℓ_p-norm that is restrained by a fixed 'η'. In (1), restricting ℓ_{target} to a pre-defined label results in a targeted adversarial attack.

According to (1), $\boldsymbol{p}$ can also be expressed as a function over $\boldsymbol{I}$ and $\mathcal{K}(.)$[1]. Given a fixed $\mathcal{K}(.)$, the objective of computing an image-specific perturbation confines the domain of $\boldsymbol{p}$, say $\text{Dom}(\boldsymbol{p})$ to the extreme case of a single image. With such restrictions the perturbation signal can only reflect peculiarities of a single data point *w.r.t.* $\mathcal{K}(.)$, that is hardly indicative of any general character of the classifier. This also calls into question the relevance of claiming human perceptual misalignment with deep representation by alluding to image-specific perturbations. To better encode the classifier information in the perturbation, the signal needs to be invariant to the input samples, which is achievable by broadening the domain of $\boldsymbol{p}$.

Incidentally, universal perturbations [29] are computed with a broader domain as per our formulation. Inline with our reasoning, those perturbations exhibit much more regular geometric patterns as compared to the image-specific perturbations. However, those patterns still remain far from salient visual features of any object. This is because universal perturbations map all the input images to random class labels. For a given $\mathcal{K}(.)$, broadening the perturbation domain with a 'targeted' objective is more likely to induce the geometric patterns in $\boldsymbol{p}$ that are actually considered salient features of ℓ_{target} by $\mathcal{K}(.)$.

Further to the above argument, we can alternately describe the objective of (1) as maximizing the probability of a perturbed sample being mapped to ℓ_{target} by $\mathcal{K}(.)$. For $|\text{Dom}(\boldsymbol{p})| > 1$, where $|.|$ is the set cardinally, this maximization must incorporate all the relevant samples. Hence, we re-cast (1) into the following constraint for our objective of explaining a deep model with $\boldsymbol{p}$:

$$\mathbb{E}\big[P\left(\mathcal{K}(\boldsymbol{I}+\boldsymbol{p}) \rightarrow \ell_{\text{target}}\right)\big] \geq \gamma, \;\; s.t. \qquad (2)$$
$$\text{Dom}(\boldsymbol{p}) = \{\forall \boldsymbol{I} | \boldsymbol{I} \sim \mathcal{I}\}, |\text{Dom}(\boldsymbol{p})| \gg 1, ||\boldsymbol{p}||_p \leq \eta,$$

where $P(.)$ denotes probability and $\gamma \in [0, 1]$ is a pre-defined constant. As compared to the commonly computed adversarial perturbations, a $\boldsymbol{p}$ satisfying (2) is expected to reveal clear information about e.g. what constitutes discriminative visual features of objects for a model?, what semantics are attached to a given label index of the model?, and do these features and semantics are human-meaningful? etc.

4. Algorithm

We compute the desired perturbations in two phases. In the first phase of *perturbation estimation*, discriminative features of the target class (as perceived by the classifier) are induced in the perturbation in a holistic manner. Later, in the phase of *perturbation refinement*, the technique focuses more on the image regions that cause high neural activity in the model to refine the perturbation.

Perturbation estimation: To expand our perturbation domain, we need to sample a distribution of images. Considering $\mathcal{I}$, we define a set $\Im = \{\boldsymbol{d}\} \cup \overline{\mathcal{D}}$ of the samples from that distribution. Here, $\boldsymbol{d} \in \mathbb{R}^m$ denotes a 'seed' image, whereas each element of $\overline{\mathcal{D}}$ is also a sample from $\mathcal{I}$. We adopt this formalization to explicate the role of distribution and seed choice in the subsequent text.

The procedure to estimate the perturbation is summarized as Algorithm 1, that solves the optimization problem below with a guided stochastic gradient descent strategy

$$\max_{\boldsymbol{p}} \wp = \mathop{\mathbb{E}}_{\boldsymbol{I} \sim \Im}\big[P\left(\mathcal{K}(\boldsymbol{I}+\boldsymbol{p}) \rightarrow \ell_{\text{target}}\right)\big] \; s.t. \; ||\boldsymbol{p}||_2 \leq \eta. \quad (3)$$

At its core, the algorithm employs mini-batches of the distribution samples for a multi-step traversal of the visual model's cost surface to solve (3). We bias this traversal with the seed image. The algorithm iteratively steps in the direction of increasing '$\wp$' by computing the gradient of the surface *w.r.t.* mini-batches and utilizing the gradient moments for efficient optimization. Instead of aiming at the optimal solution, based on (2), we accept any solution for which $\wp \geq \gamma$. Below, we describe this procedure in detail, following the sequence in Algorithm 1.

We compute the desired perturbation expecting the inputs mentioned in Algorithm 1. Briefly ignoring the initialization on line-1, the algorithm first randomly selects $b-1$ samples to form a set $\mathcal{D}$ and clips these samples and the input seed $\boldsymbol{d}$ after perturbing them with the current estimate of the perturbation (line 3&4). The clipping is performed to

[1] We assume that the algorithm to generate $\boldsymbol{p}$ is fixed.

Algorithm 1 Perturbation estimation

Input: Classifier $\mathcal{K}$, seed $\boldsymbol{d}$, raw samples $\overline{\mathcal{D}}$, target label ℓ_{target}, perturbation norm η, mini-batch size b, probability threshold γ.

Output: Perturbation $\boldsymbol{p} \in \mathbb{R}^m$.

1: Initialize $\boldsymbol{p}_0$, $\boldsymbol{\mu}_0$, $\boldsymbol{\sigma}_0$ to $\mathbf{0} \in \mathbb{R}^m$ and $t = \wp = 0$
Set $\alpha = 0.9$, $\beta = 0.999$ and $\overline{\boldsymbol{d}} = \boldsymbol{d}$.
2: **while** $\wp < \gamma$ **do**
3: $\quad \mathcal{D} \sim \overline{\mathcal{D}}$, s.t. $|\mathcal{D}| = b - 1$
4: $\quad \mathcal{D} \leftarrow \text{Clip}\left(\mathcal{D} \ominus \boldsymbol{p}_t\right), \boldsymbol{d} \leftarrow \text{Clip}\left(\boldsymbol{d} \ominus \boldsymbol{p}_t\right)$,
5: $\quad t \leftarrow t + 1$
6: $\quad \xi \leftarrow \frac{||\nabla_{\boldsymbol{d}} \mathcal{J}(\boldsymbol{d}, \ell_{\text{target}})||_2}{\mathop{\mathbb{E}}_{\boldsymbol{d}_i \in \mathcal{D}}[||\nabla_{\boldsymbol{d}_i} \mathcal{J}(\boldsymbol{d}_i, \ell_{\text{target}})||_2]}$
7: $\quad \boldsymbol{g_t} \leftarrow \frac{1}{2}\nabla_{\boldsymbol{d}} \mathcal{J}(\boldsymbol{d}, \ell_{\text{target}}) + \frac{\xi}{2} \mathop{\mathbb{E}}_{\boldsymbol{d}_i \in \mathcal{D}}\left[\nabla_{\boldsymbol{d}_i} \mathcal{J}(\boldsymbol{d}_i, \ell_{\text{target}})\right]$
8: $\quad \boldsymbol{\mu}_t \leftarrow \alpha \boldsymbol{\mu}_{t-1} + (1 - \alpha)\boldsymbol{g}_t$
9: $\quad \boldsymbol{\sigma}_t \leftarrow \beta \boldsymbol{\sigma}_{t-1} + (1 - \beta)(\boldsymbol{g}_t \odot \boldsymbol{g}_t)$
10: $\quad \boldsymbol{\rho} \leftarrow \left(\boldsymbol{\mu}_t \sqrt{1 - \beta^t}\right) \odot \left(\sqrt{\boldsymbol{\sigma}_t}(1 - \alpha^t)\right)^{-1}$
11: $\quad \mathcal{D}^+_{\boldsymbol{\rho}} \leftarrow \overline{\mathcal{D}} \ominus \left(\boldsymbol{p}_{t-1} + \frac{\boldsymbol{\rho}}{||\boldsymbol{\rho}||_\infty}\right)$
12: $\quad \mathcal{D}^-_{\boldsymbol{\rho}} \leftarrow \overline{\mathcal{D}} \ominus \left(\boldsymbol{p}_{t-1} - \frac{\boldsymbol{\rho}}{||\boldsymbol{\rho}||_\infty}\right)$
13: $\quad \varrho^+ \leftarrow \mathbb{E}\left[P(\mathcal{K}(\mathcal{D}^+_{\boldsymbol{\rho}}) \rightarrow \ell_{\text{target}})\right]$
14: $\quad \varrho^- \leftarrow \mathbb{E}\left[P(\mathcal{K}(\mathcal{D}^-_{\boldsymbol{\rho}}) \rightarrow \ell_{\text{target}})\right]$
15: $\quad$ **if** $\varrho^+ \geq \varrho^-$ **then**
16: $\quad\quad \boldsymbol{p_t} \leftarrow \boldsymbol{p}_{t-1} + \boldsymbol{\rho}$
17: $\quad$ **else**
18: $\quad\quad \boldsymbol{p_t} \leftarrow \boldsymbol{p}_{t-1} - \boldsymbol{\rho}$
19: $\quad$ **end if**
20: $\quad \boldsymbol{p_t} \leftarrow \boldsymbol{p_t} \odot \min\left(1, \frac{\eta}{||\boldsymbol{p}_t||_2}\right)$
21: $\quad \Im_p \leftarrow \text{Clip}(\{\overline{\boldsymbol{d}} \cup \overline{\mathcal{D}}\} \ominus \boldsymbol{p_t})$
22: $\quad \wp \leftarrow \mathbb{E}\left[P(\mathcal{K}(\Im_p) \rightarrow \ell_{\text{target}})\right]$
23: **end while**
24: return

confine the dynamic range of the resulting samples to $[0, 1]$. The $\ominus$ symbol indicates that perturbation is being applied to a sample or individual elements of a set. For a given iteration, clipped $\boldsymbol{d} \cup \mathcal{D}$ forms a mini-batch that is used by our stochastic gradient descent strategy.

The seed is introduced in our algorithm to allow variation in the perturbations by changing this input. We do not assume any restrictions over the input samples, implying that the elements of $\mathcal{D}$ and $\boldsymbol{d}$ can be widely different. This also means that the gradients of $\boldsymbol{d}_i \in \mathcal{D}$ in the direction of ℓ_{target} - denoted by $\nabla_{\boldsymbol{d}_i} \mathcal{J}(\boldsymbol{d}_i, \ell_{\text{target}})$ - can significantly differ from their counterpart computed for $\boldsymbol{d}$. To account for this difference, line-6 of the algorithm computes the ratio between the gradient norm for $\boldsymbol{d}$ and the Expected gradient norm for $\boldsymbol{d}_i \in \mathcal{D}$. The ratio is later used to fuse the gradients on line-7, giving higher relevance to the seed gradient.

Given the fused gradient, we estimate its first and second raw moment on line-8 & 9 using the exponential running average controlled by the hyper-parameters 'α' and 'β'. In our algorithm, the use of adaptive moments is inspired by the Adam algorithm [23] that employs this scheme for model parameter optimization. After empirically verifying the qualitative similarity between the effects of these hyper-parameters on our algorithm and Adam, we fix their values to those proposed in [23]. This is indicated on line-1, where the other parameters are initialized to null values and a copy of the seed is created for subsequent processing.

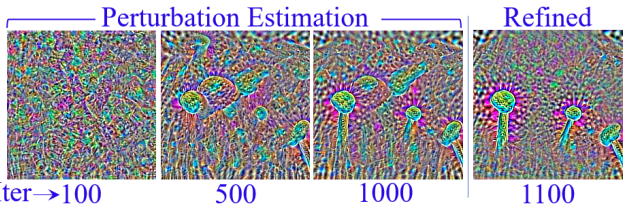

Figure 2. Visually salient geometric patterns emerge with more iteration of Algorithm 1 that are further refined with Algorithm 2. The refined perturbation is shown after post-refinement 100 iterations of the former. The 'Nail' patterns are computed for VGG-16 with $\eta = 10$. We follow [45] for perturbation visualization.

We combine the running averages on line-10 and then perform a binary search for the resulting intermediate perturbation update signal $\boldsymbol{\rho}$ on lines 11-19. The search monitors if changing the direction of $\boldsymbol{\rho}$ is more conducive for our ultimate objective. Stochasticity can cause our optimization to significantly deviate from the eventual objective in a given iteration. On one hand, the binary search inhibits this case. On the other, it introduces more variety in the perturbation that is desirable for better model explanation. We project the updated perturbation to the ℓ_2-ball of radius 'η' on line-20, and estimate '$\wp$' on the perturbed clipped distribution samples on line-21 & 22.

Whereas the ℓ_p-norm of perturbation is restricted in adversarial settings for imperceptibility, this constraint plays a different role in our technique. By iterative back-projection and clipping, we keep amplifying those geometric patterns in the perturbation that strongly influence $\mathcal{K}(.)$ to predict ℓ_{target} as the label of all the input samples. With successive back-projections, visually salient feature of ℓ_{target} start to emerge in our perturbations (Fig. 2) that are subsequently refined for better visualization, as discussed below.

Perturbation refinement: The holistic treatment of perturbation in Algorithm 1 results in an unrestricted spread of energy over the signal. To achieve finer patterns we let the technique focus more on the relevant regions with an adaptive filtration mechanism summarized in Algorithm 2. A key property of this mechanism is that it upholds model fidelity of the perturbation by assuming no external priors.

To refine the perturbation, it is fed to the convolutional base $\bar{\mathcal{K}}(.)$ of the classifier (line-2). The output Ω of the base is a set of low resolution 2D signals, which is reduced to an average signal $\boldsymbol{a}$ on line-3. This signal captures rough

Algorithm 2 Perturbation refinement

Input: Classifier $\mathcal{K}$, perturbation $\boldsymbol{p} \in \mathbb{R}^{\mathrm{m}}$
Output: Refined perturbation $\boldsymbol{p}$

1: Initialize $\boldsymbol{f}$ to $\mathbf{0} \in \mathbb{R}^{\mathrm{m}}$
   Set $\bar{\mathcal{K}}$ = convolutional base of $\mathcal{K}$, scale factor $\lambda = 5$
2: $\Omega \leftarrow \bar{\mathcal{K}}(\boldsymbol{p}) : \Omega \in \mathbb{R}^{H \times W \times C}$
3: $\boldsymbol{a} \leftarrow \frac{1}{C} \sum_{n=1}^{\mathrm{C}} \Omega^n$
4: $\tau \leftarrow \Psi(\boldsymbol{a})$
5: **if** $\boldsymbol{a}(\mathrm{x},\mathrm{y}) > \tau$ **then** $\boldsymbol{a}(\mathrm{x},\mathrm{y}) = \lambda$ **else** $\boldsymbol{a}(\mathrm{x},\mathrm{y}) = 0$
6: $\boldsymbol{f} \leftarrow$ upsample $(\boldsymbol{a}) : \boldsymbol{f} \in \mathbb{R}^{\mathrm{m}}$
7: $\boldsymbol{p} \leftarrow \mathrm{Clip}(\boldsymbol{p} \odot \boldsymbol{f})$
8: return

silhouette of the salient regions in the input perturbation, which makes it a useful spatial filter for our technique. On line-4, $\Psi(.)$ computes the Otsu threshold [32] for the average signal, that is subsequently used to binarize the image on line-5. We empirically set $\lambda = 5$ in this work. The resulting image is up-sampled by bicubic interpolation [21] on line-6 to match the dimensions of the input perturbation $\boldsymbol{p}$. The scaled mask is applied to the perturbation, which is subsequently clipped to the valid dynamic range.

The output of Algorithm 2 is further processed by Algorithm 1 to again highlight any salient patterns that might be diminished with filtration. The final perturbation is computed by iterating between the two algorithms.

5. Experimentation

We experiment with the proposed algorithm for model explanation in § 5.1, and to perform low-level image processing in § 5.2. The former uses standard 'non-robust' classifiers, whereas 'adversarially robust' classifiers are used for the latter.

5.1. Model explanation

Setup: We assume $\mathcal{I}$ to be a distribution of natural images and create our set $\overline{\mathcal{D}}$ by randomly sampling 256 images from the validation set of ILSVRC 2012 dataset [12]. Random samples are used for each experiments separately. We consider visual models trained on the ImageNet as our classifiers and arbitrarily select the target label ℓ_{target}. A mini-batch size of $b = 32$ is used. To compute the perturbations, we set the probability threshold $\gamma = 0.8$ and perturbation norm $\eta = 10$. The value of 'γ' is chosen based on the visual clarity of salient patterns in the final perturbations. Higher 'γ' tends to generate clearer patterns at higher computational cost. We keep 'η' comparable to the existing techniques for adversarial perturbation generation [29, 1]. NVIDIA Titan V GPU with 12 GB RAM is used.

To compute a perturbation, we first let Algorithm 1 run to achieve the desired '$\wp$'. Then, we apply Algorithm 2 for refinement. Subsequently, Algorithm 1 is again applied such that a refinement is carried out after every 50^{th} iteration until 300 iterations.

Salient visual features: Model-gradient based adversarial perturbations are known to generate noise-like patterns [45, 16, 30] or motifs that seem meaningless to humans [29, 1]. However, by accumulating such perturbations under a slightly different objective, our attack is able to discover visually salient features of the target labels in those signals. In Fig. 3, we show representative examples of the perturbations computed by our algorithm for VGG-16 model. Notice the clear geometric patterns that humans can associate with the target class labels. These patterns emerge without assuming any priors on the perturbation, distribution samples (in $\overline{\mathcal{D}}$), or the model itself.

Firstly, from the figure, it is apparent that our technique can (qualitatively) explain a model in terms of 'what human-meaningful semantics are attached to its output neurons?'. This is useful e.g. in the settings where an unknown model is available and one must discover the labels of its output layer. Secondly, the perturbations are explaining 'what geometric patterns are perceived as the discriminative features of a given class *by the classifier*?'. Interestingly, these patterns align very well with the human perception, and we compute them with the same tool (i.e. gradient based perturbation) that is used to promote the argument of misalignment between human perception and deep representation [14, 46].

Diversity of the salient patterns: We provide two representative perturbations for each target class in Fig. 3, where the difference in the perturbations is caused by selecting different seeds. Besides ascertaining the effective role of seed in our algorithm, the diverse patterns that remain visually salient, affirm that the model has learned the general (human-meaningful) semantics for the target label. We emphasize that we ignored the target class while creating $\overline{\mathcal{D}}$ for Fig. 3. Hence, the patterns are completely based on the visual model, which also highlights the potential of standard classifiers for the task of diverse image generation.

Region specific semantics: Intrigued by the spatial distribution of the salient patterns in perturbations, we also explore the possibility of extracting model semantics associated with specific regions in the image space. This is possible by increasing the correlation between the pixels in those regions across the distribution samples that are input to our algorithm. This leads the gradients for the individual samples to be in the same direction for the specified regions. Which reinforces the signal for those regions with back-projection while weak signals in the other regions get suppressed with refinement. We emulate this scenario by replacing the image regions of interest with 64×64 patches for all the samples, where all patch pixels are generated with the mean pixel value of the sampled images.

In Fig. 4, we show the perturbations for a representative

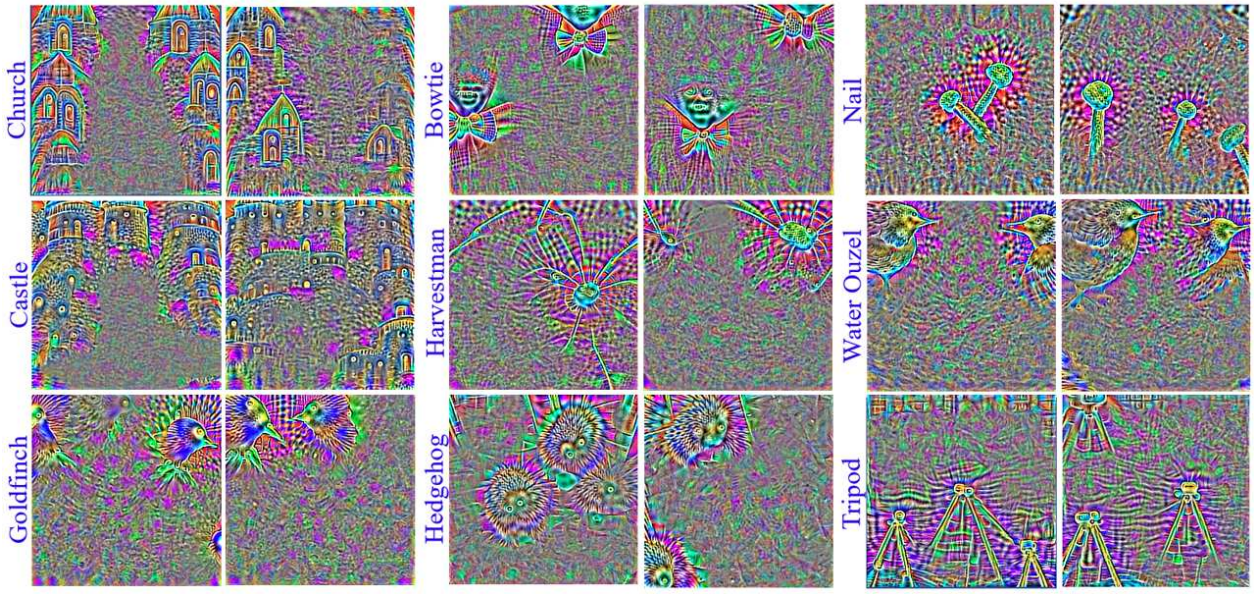

Figure 3. Visually salient features of the target class (label given) emerge by accumulating the gradient based perturbations with explanation objective. The shown perturbations are computed for VGG-16 with ImageNet samples, excluding the target class samples. Perturbations for the same target are generated with different seeds for variety.

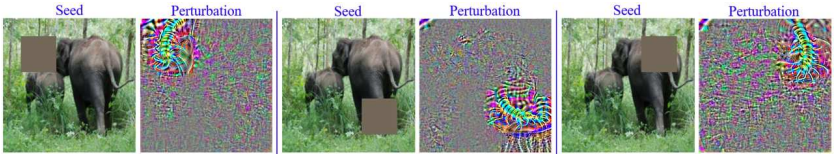

Figure 4. Obstructing samples with a uniform patch (seed shown) lets the algorithm focus on/near the pre-specified region for extracting model semantics. Perturbations for 'Centipede' are computed for VGG-16.

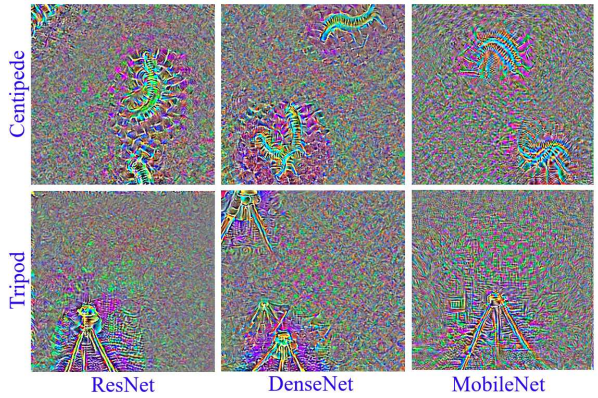

Figure 5. Salient pattern emergence is a general phenomenon. Patterns for two random labels are shown for different models.

label ('Centipede') with three random choices of regions. A region of interest is depicted with seed only. As can be observed, our attack is able to focus much better near the specified regions. Interestingly, the model is generally able to associate similar discriminative features of the target label to different regions in a coherent manner, further strengthening the notion of human perception alignment with the deep representation.

Patterns for different models: Above, we mainly presented the patterns for VGG for their visual clarity after resizing. However, emergence of the salient visual features in our perturbations is a general phenomenon for the deep visual classifiers. In Fig. 5, we also show representative perturbations for ResNet-50 [17], DenseNet-121 [18] and MobileNet-V2 [40] for two random classes used in our experiments. The perturbations clearly depict the features of the target labels for all these model.

To demonstrate perceptual alignment of deep representation over different models, we classify the 'perturbations' generated for one model with other models. High confidence of multiple models for the intended target label indicates that the extracted patterns are commonly seen as discriminative visual features of the target class.

5.2. Leverage in low-level tasks

Santurakar et al. [41] recently showed that adversarially robust deep classifiers can be exploited beyond classification. They demonstrated image generation, inpainting and image manipulation etc. by attacking a robust ResNet with the PGD attack [28]. The key notion exploited by Santurakar et al. is the presence of salient visual features in the 'adversarial' perturbations computed for the 'robust' classifiers. Relating this concept to our findings, their study indicates an excellent test bed for our attack, where successful results not only ascertain the implicit model explaining nature of our perturbations, but also improves the state-of-

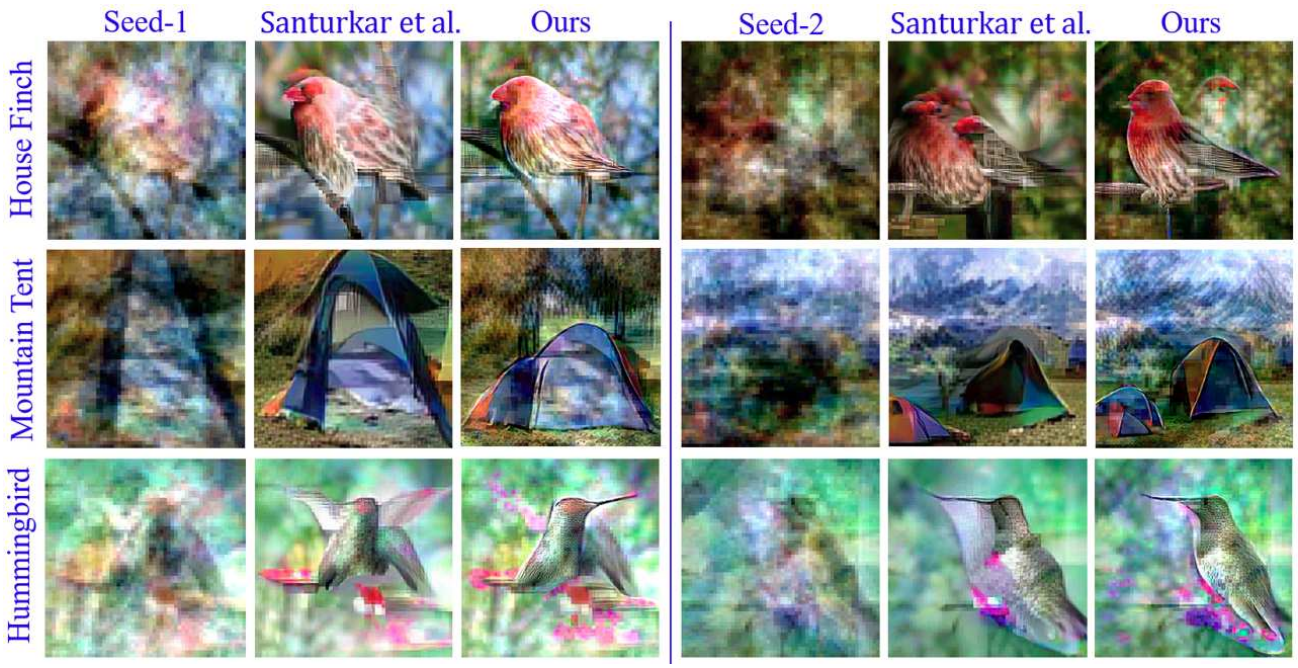

Figure 6. Image generation by attacking adversarially robust ResNet. The generated images are adversarial examples of the shown seeds. The intended class labels are mentioned. Setup of Santurkal et al. [41] is followed.

the-art for the newly found place of the robust classifiers in the broader Machine Learning context.

To demonstrate improvements in the results, we follow [41] closely in terms of the used classifier, perturbation budget and the underlying evaluation procedure. In the experiments to follow, we create the set $\overline{\mathcal{D}}$ by sampling a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}_I, \Sigma_I)$, where $\boldsymbol{\mu}_I \in \mathbb{R}^m$ is the mean value of an image set $\boldsymbol{I}_{i=1,\dots,n} \sim \mathcal{I}^{\text{target}}$. Here, $\mathcal{I}^{\text{target}}$ is the distribution of a target class images, emulated by ImageNet. We compute $\Sigma_I = \mathbb{E}[(\boldsymbol{I}_i - \boldsymbol{\mu}_I)^\intercal (\boldsymbol{I}_i - \boldsymbol{\mu}_I)]$. For computational reasons, the multivariate Gaussian is computed by $4\times$ downsampling of the original images. Random 256 distribution samples are later upsampled to match the network input and used to create the set $\overline{\mathcal{D}}$. In the following experiments, where the image processing tasks are performed holistically, we do not use the refinement step.

5.2.1 Image Generation

In Fig. 6, we show representative examples of images generated by our technique and compare those with Santurkal et al. [41]. We use the author-provided code for [41] and strictly follow the guidelines to achieve the best results of their method. In the context of adversarial attacks, the generated images are adversarial examples of the seed images. We show two images per class, generated with the shown seeds for the mentioned target label. Our technique is clearly able to generate more refined and coherent images. Notice the details in the backgrounds as well. Theoretically, Santurkar et al. [41] used the strongest gradient-based iterative adversarial attack [28] in their method. Hence, our improved performance can be easily attributed to the model explaining nature of the perturbations computed by the proposed attack. We use the same perturbation budget $\eta = 40$ for both the techniques.

The variety in the images generated with different seeds, their textural details and clear semantic coherence strengthen the broader idea that robust classifiers are capable of more than simple classification [41] - a worth exploring venue for the future research.

5.2.2 Inpainting

Image inpainting [4] restores information in large corrupt regions of images while upholding the perceptual consistency. We demonstrate improved inpainting performance with robust classifiers using the proposed attack.

For this task, we treat the corrupted image as the seed, where its corrupt region is identified as a binary mask $F \in \{0,1\}^m$. Let $\Im$ contain the seed and samples form our above-mentioned multivariate Gaussian distribution $\mathcal{N}(.)$. Keeping the robust classifier parameters fixed, we minimize the following loss:

$$\mathcal{L}(\boldsymbol{p}) = \mathbb{E}\big[\mathcal{J}(\Im_p, l_{target}) + \beta \left(\boldsymbol{p} \odot (1 - F)\right)\big], \qquad (4)$$

where $\Im_p = \Im \ominus \boldsymbol{p}$, $\mathcal{J}(.)$ is the cross-entropy loss of the classifier and $\beta = 10$ is an empirically chosen scaling factor. The designed loss function allows the perturbation signal to grow freely for the corrupt region while restricting it in the other regions.

In Fig. 7, we show representative examples of corrupt images restored with our technique and Santurkar et al. [41]

Figure 7. Representative inpainting results. The Masked image is the seed. Both approaches restore images using the same robust model provided by Santurkar et al. [41]. using the same perturbation budget.

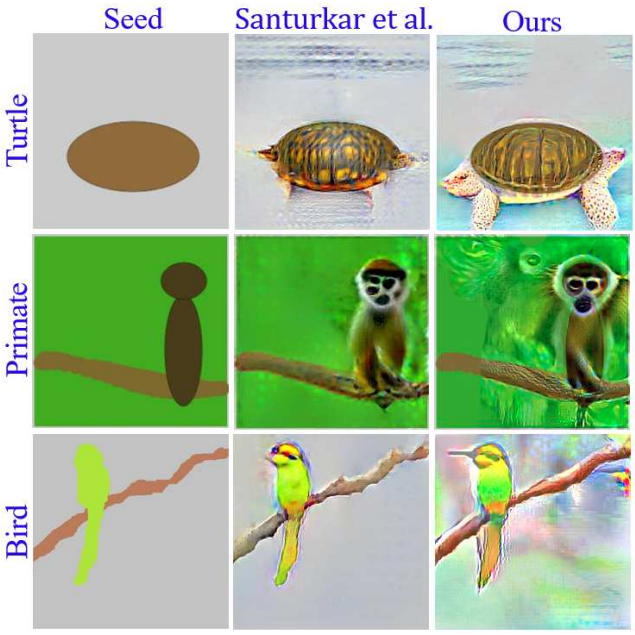

Figure 8. Representative examples of interactive image manipulation. The seed is a raw image required to be manipulated into an image of the target category. Both techniques use the same robust classifier with perturbation budget 60, optimized for the images.

using the robust ResNet provided by the authors. We use the same perturbation budget $\eta = 21$ for both techniques. The restoration quality of our technique is visibly better. The shown images and mask placements are randomly selected.

5.2.3 Interactive Image Manipulation

An interesting recent application of deep networks, especially GANs [10] is to turn crude sketches into realistic images. Santurkar et al. [41] demonstrated the possibility of such interactive image manipulation by attacking/fooling robust classifiers. We advance this direction by demonstrating that our alternate objective of model explanation is more suitable for the problem.

Using the raw sketch as the seed and creating the set $\overline{D}$ with the multivariate Gaussian, we manipulate the seed similar to image generation. However, this time we also apply the refinement procedure. Representative results of our attack are shown in Fig. 8. Compared to [41], images generated with our technique appear much more realistic. Such a refined manipulation of crude sketches with a classifier affirms the ability of our attack to highlight human-meaningful visual patterns learned by the classifier.

The three low-level image processing tasks discussed above not only demonstrate the utility of perturbations beyond model fooling (in general), but also ascertain that our attack is a positive step forward in that direction.

6. Conclusion

We present the first attack on deep learning that has an objective of explaining the model instead of fooling it. To compute the perturbation, our attack performs a stochastic gradient search on the cost surface of the model to increase the log-probability of a 'distribution' of images to be classified as a particular target. By iterative back-projection of the gradients and refinement with adaptive attention, our attack finds geometric patterns in the perturbations that are deemed salient by the classifier. We find that these patterns align well with the human perception, which weakens the argument of misalignment between human perception and deep representation - in the context of adversarial perturbations. Besides demonstrating perturbations with visually salient features for multiple state-of-the-art classifiers, we also perform low-level image manipulation with our technique using robust classifiers. Realistic image generation, inpainting and interactive image manipulation ascertain the model explaining nature of our attack, and advance the state-of-the-art in these newly found classifier utilities.

Acknowledgment This research was supported by ARC Discovery Grant DP190102443, DP150100294 and DP150104251. The Titan V used in our experiments was donated by NVIDIA corporation.

References

[1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3389–3398, 2018. 1, 2, 5

[2] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 1, 2

[3] Anish Athalye and Nicholas Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018. 2

[4] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000. 7

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 2

[6] Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016. 2

[7] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017. 2

[8] Nicholas Carlini and David Wagner. Magnet and" efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*, 2017. 2

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[10] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018. 8

[11] Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. *arXiv preprint arXiv:1909.05040*, 2019. 1, 2

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 5

[13] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1, 2

[14] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Learning perceptually-aligned representations via adversarial robustness. *arXiv preprint arXiv:1906.00945*, 2019. 2, 3, 5

[15] Aditya Ganeshan and R Venkatesh Babu. Fda: Feature disruptive attack. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8069–8079, 2019. 2

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2, 5

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019. 1

[20] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6084–6092, 2019. 2

[21] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981. 5

[22] Valentin Khrulkov and Ivan Oseledets. Art of singular vectors and universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8562–8570, 2018. 2

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[25] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1, 2

[26] Jiayang Liu, Weiming Zhang, Yiwei Zhang, Dongdong Hou, Yujia Liu, Hongyue Zha, and Nenghai Yu. Detection based defense against adversarial examples from the steganalysis point of view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4825–4834, 2019. 1, 2

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 3, 6, 7

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 2, 3, 5

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1, 2, 5

[31] Konda Reddy Mopuri, Aditya Ganeshan, and Venkatesh Babu Radhakrishnan. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[32] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. 5

[33] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431, 2018. 2

[34] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8571–8580, 2018. 1, 2

[35] Yuxian Qiu, Jingwen Leng, Cong Guo, Quan Chen, Chao Li, Minyi Guo, and Yuhao Zhu. Adversarial defense through network profiling based path extraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4777–4786, 2019. 2

[36] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6528–6537, 2019. 1, 2

[37] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[39] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019. 2

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 6

[41] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Computer vision with a single (robust) classifier. *arXiv preprint arXiv:1906.09453*, 2019. 1, 2, 6, 7, 8

[42] Yucheng Shi, Siyu Wang, and Yahong Han. Curls & whey: Boosting black-box adversarial attacks. *arXiv preprint arXiv:1904.01160*, 2019. 1, 2

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[44] Bo Sun, Nian-hsuan Tsai, Fangchen Liu, Ronald Yu, and Hao Su. Adversarial defense by stratified convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11447–11456, 2019. 2

[45] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2, 4, 5

[46] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, number 2019, 2019. 3, 5

[47] Walt Woods, Jack Chen, and Christof Teuscher. Reliable classification explanations via adversarial attacks on robust networks. *arXiv preprint arXiv:1906.02896*, 2019. 3

[48] Lei Wu, Zhanxing Zhu, Cheng Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018. 2

[49] Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton van den Hengel. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1367–1381, 2017. 1

[50] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019. 1, 2

[51] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2

[52] Tianhang Zheng, Changyou Chen, and Kui Ren. Distributionally adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2253–2260, 2019. 2