Towards Feature Space Adversarial Attack by Style Perturbation

Qiuling Xu, Guanhong Tao, Siyuan Cheng, Xiangyu Zhang
Department of Computer Science, Purdue University
305 N University St
West Lafayette, Indiana 47907
{xu1230, taog, cheng535, xyzhang}@purdue.edu

Abstract

We propose a new adversarial attack to Deep Neural Networks for image classification. Different from most existing attacks that directly perturb input pixels, our attack focuses on perturbing abstract features, more specifically, features that denote styles, including interpretable styles such as vivid colors and sharp outlines, and uninterpretable ones. It induces model misclassfication by injecting imperceptible style changes through an optimization procedure. We show that our attack can generate adversarial samples that are more natural-looking than the state-of-the-art unbounded attacks. The experiment also supports that existing pixel-space adversarial attack detection and defense techniques can hardly ensure robustness in the style related feature space. [1]

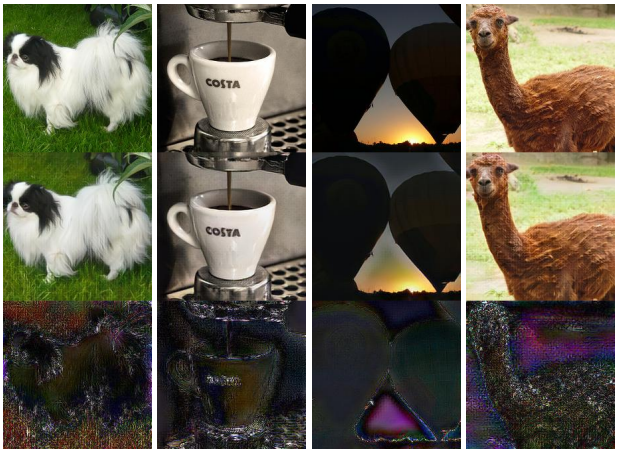
Introduction

Adversarial attacks are a prominent threat to the broad application of Deep Neural Networks (DNNs). In the context of classification applications, given a pre-trained model M and a benign input $\boldsymbol{x}$ of some output label y, adversarial attack perturbs $\boldsymbol{x}$ such that M misclassifies the perturbed $\boldsymbol{x}$. The perturbed input is called an *adversarial example*. Such perturbations are usually bounded by some distance norm such that they are not perceptible by humans. Since it was proposed in (Szegedy et al. 2014), there has been a large body of research that develops various methods to construct adversarial examples with different modalities (e.g., images (Carlini and Wagner 2017), audio (Qin et al. 2019), text (Ebrahimi et al. 2018), and video (Li et al. 2019)), to detect adversarial examples (Tao et al. 2018; Ma et al. 2019), and use adversarial examples to harden models (Madry et al. 2018; Zhang et al. 2019).

However, most existing attacks (in the context of image classification) are in the pixel space. That is, bounded perturbations are directly applied to pixels. In this paper, we illustrate that adversarial attack can be conducted in the style related feature space. The underlying assumption is that during training, a DNN may extract a large number of abstract features. While many of them denote critical characteristics of the object, some of them are secondary, for example, the different styles of an image (e.g., vivid colors versus pale colors, sharp outlines versus blur outlines). These secondary features may play an improperly important role in model prediction. As a result, feature space attack can inject such secondary features, which are not simple pixel perturbation, but rather functions over the given benign input, to induce model misclassification. Since humans are not sensitive to these features, the resulted adversarial examples look natural from humans' perspective. As many of these features are pervasive, the resulted pixel space perturbation may be much more substantial than existing pixel space attacks. As such, pixel space defense techniques may become ineffective for feature space attacks (see Evaluation section). Figure 1 shows a number of adversarial examples generated by our technique, their comparison with the original examples, and the pixel space distances. Observe that while the distances are much larger compared to those in pixel space attacks, the adversarial examples are natural, or even indistinguishable from the original inputs in humans' eyes. The contrast of the benign-adversarial pairs illustrates that the malicious perturbations largely co-locate with the primary content features, denoting imperceptible style changes.

Under the hood, we consider that the activations of an inner layer represent a set of abstract features, including those primary and secondary. Distinguishing the two types of features is crucial for the quality of feature-space attack. To avoid generating adversarial examples that are unnatural, we refrain from tampering with the primary features (or *content features*) and focus on perturbing the secondary *style features*. Inspired by the recent advance in style transfer (Huang and Belongie 2017), the *mean* and *variance* of activations are considered the style. As such, we focus on perturbing the means and variances while preserving the *shape* of the activation values (i.e., the up-and-downs of these values and the relative scale of such up-and-downs). We use gradient driven optimization to search for the style perturbations that can induce misclassification. Since our threat model is the same as existing pixel space attacks, that is, the attack is launched by providing the adversarial example to the model. An important step is to translate the activations with style changes back to a naturally looking pixel space example. We address the problem by considering the differences of any pair of training inputs of the same class as the possible style

[1] The appendix and code are available at https://arxiv.org/abs/2004.12385 and https://github.com/qiulingxu/FeatureSpaceAttack respectively.



(a) Spaniel	(b) Espresso	(c) Balloon	(d) Llama
ℓ_∞:121/255	192/255	149/255	183/255
ℓ_2:25.92	24.47	20.75	28.55

Figure 1: Examples by feature space attack. The top row presents the original images. The middle row denotes the adversarial samples. The third row shows the pixel-wise difference ($\times 3$) between the original and the adversarial samples. The ℓ_∞ and ℓ_2 norms are shown on the bottom.

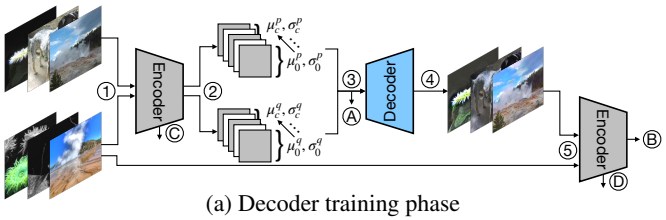
differences, and pre-training a decoder that can automatically impose styles in the pixel space based on the style feature variation happening in an inner layer. We propose two concrete feature space attacks, one to enhance styles and the other to impose styles constituted from a set of pre-defined style prototypes.

We evaluate our attacks on 3 datasets and 7 models. We show that feature space attacks can effectively generate adversarial samples. The generated samples have natural, and in many cases, human imperceptible style differences compared with the original inputs. Our comparative experiment with recent attacks on colors (Hosseini and Poovendran 2018) and semantics (Bhattad et al. 2020) shows that our generated samples are more natural-looking. We also show that 7 state-of-the-art detection/defense approaches are ineffective to our attack as they focus on protecting the pixel space. Particularly, our attack reduces the detection rate of a state-of-the-art pixel-space approach (Roth, Kilcher, and Hofmann 2019) to 0.04% on the CIFAR-10 dataset, and the prediction accuracy of a model hardened by a state-of-art pixel-space adversarial training technique (Xie et al. 2019) to 1.25% on ImageNet. Moreover, we observe that despite the large distance introduced in the pixel space (by our attack), the distances in feature space are similar or even smaller than those in ℓ-norm based attacks. Note that the intention of these experiments is not to claim our attack is superior, but rather to illustrate that new defense and hardening techniques are needed for feature space protection.

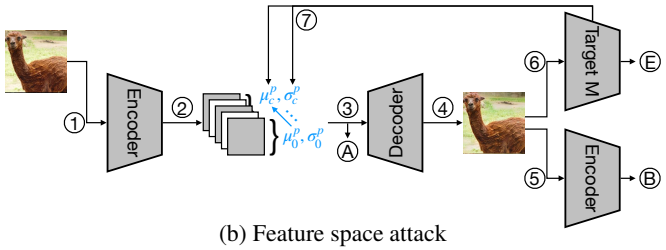
Background and Related Work

Style Transfer. Huang and Belongie (2017) proposed to transfer the style from a (source) image to another (target) that may have different content such that the content of the target image largely retains while features that are not essential to the content align with those of the source image. Specifically, given an input image, say the portrait of actor Brad Pitt, and a style picture, e.g., a drawing of painter Vincent van Gogh, the goal of style transfer is to produce a portrait of Brad Pitt that looks like a picture painted by Vincent van Gogh. Existing approaches leverage various techniques to achieve this purpose. Gatys, Ecker, and Bethge (2016) utilized the feature representations in convolutional layers of a DNN to extract *content features* and *style features* of input images. Given a random white noise image, the algorithm feeds the image to the DNN to obtain the corresponding content and style features. The content features from the white noise image are compared with those from a content image, and the style features are contrasted with those from a style image. It then minimizes the above two differences to transform the noise image to a content image with style. Due to the inefficiency of this optimization process, researchers replace it with a neural network that is trained to minimize the same objective (Li and Wand 2016; Johnson, Alahi, and Fei-Fei 2016). Further study extends these approaches to synthesize more than just one fixed style (Dumoulin, Shlens, and Kudlur 2017; Li et al. 2017). Huang and Belongie (2017) introduced a simple and yet effective approach, which can efficiently enable arbitrary style transfer. It proposed an *adaptive instance normalization* (AdaIN) layer that aligns the mean and variance of the content features with those of the style features.

Adversarial Attacks beyond Pixel Space. The exploration beyond ℓ-norm based attacks is rising. Inkawhich et al. (2019) found that simulating feature representation of target label improves transferability. Hosseini and Poovendran (2018) proposed to modify the HSV color space to generate adversarial samples. The method transforms all pixels by a non-parametric function uniformly. Differently, our feature space attack changes colors of objects or background and the transformation is learned from images of the same object with different styles. It is hence more imperceptible. Laidlaw and Feizi (2019) proposed to change the lighting condition and color (like (Hosseini and Poovendran 2018)) to generate adversarial examples. Prabhu and UnifyID (2018) produced art-style images as adversarial samples. It does not restrict the feature space such that the generated samples are not natural looking, especially compared to ours. Bhattad et al. (2020) generated semantic adversarial examples by modifying color and texture. It advocates not to restrict attack space and is hence considered unbounded. As such, it is difficult to control the attack to avoid generating unrealistic samples. In contrast, our attack has a well-defined attack space while being unbounded in the pixel space. It implicitly learns to modify lighting condition, color and texture, it tends to be more general and capable of transforming subtle (and uninterpretable) features (see Evaluation). Unlike in Song et al. (2018), where a vanilla GAN-based attack generates samples over a distribution of limited support, and has little control of the generated samples, our encoder-decoder based structure enables attacking individual samples with controlled content. Stutz, Hein, and Schiele (2018) proposed to perturb the latent embedding of VAE-GAN to generate adversarial samples.



(a) Decoder training phase



(b) Feature space attack

Figure 2: Procedure of feature space adversarial attack. Two phases are involved during the attack generation process: (a) decoder training phase and (b) feature space attack phase.

Since it does not distinguish primary and secondary features, the perturbation on primary features substantially degrades the quality of generated adversarial samples. In contrast, our feature space perturbation is more effective.

We empirically compare with these attacks later in the paper.

Feature Space Attack

Overview. We aim to demonstrate that perturbation in the feature space can lead to model misbehavior, which existing pixel space defense techniques cannot effectively defend against. The hypothesis is that during training, the model picks up numerous features, many of which do not describe the key characteristics (or *content*) of the object, but rather human imperceptible features such as styles. These subtle features may play an improperly important role in model prediction. As a result, injecting such features to a benign image can lead to misclassification. However, the feature space is not exposed to attackers such that they cannot directly perturb features. Therefore, a prominent challenge is to derive the corresponding pixel space mutation that appears natural to humans while leading to the intended feature space perturbation, and eventually the misclassification. In particular, the attack comprises two phases: (1) training a decoder that can translate feature space perturbation to pixel level changes that look natural for humans; (2) launching the attack by first using gradient based optimization to identify feature space perturbation that can cause misclassification and then using the decoder to generate the corresponding adversarial example. Inspired by style transfer techniques, we consider a much confined feature perturbation space – *style perturbation*. Specifically, as in Huang and Belongie (2017), we consider the *mean* and *variance* of the activations of an inner layer denote the style of the features in that layer whereas the activations themselves denote the content features. We hence perturb the mean and variance of content features by performing a predefined transformation that largely preserves the shape of the features while changing the mean and variance. The decoder then decodes the perturbed feature values to an image closely resembles the original image with only style differences that appear natural to humans but causing model misclassification.

Fig. 2 illustrates the workflow of the proposed attack. In the Decoder training phase (a), a set of image pairs with each pair from the same class (and hence their differences can be intuitively considered as style differences) are fed to a fixed Encoder that essentially consists of the first a few layers of a pre-trained model (e.g., VGG-19) (step ①). The Encoder produces the internal embeddings of the two respective images, which correspond to the activation values of some inner layer in the pre-trained model, e.g., conv4_1 (step ②). Each internal embedding consists of a number of matrices, one for every channel. For each embedding matrix, the mean and variance are computed. We use these values from the two input images to produce the integrated embedding Ⓐ (step ③), which will be discussed in details later in this section. Intuitively, it is generated by performing a shape-preserving transformation of the upper matrix so that it retains the content features denoted by the upper matrix while having the mean and variance of the lower matrix (i.e., the style denoted by the lower matrix). We employ a Decoder to reconstruct a raw image from Ⓐ at step ④, which is supposed to have the content of the upper image (called the *content image*) and the style of the lower image (called the *style image*). To enable good reconstruction performance, two losses are utilized for optimizing the Decoder. The first one is the *content loss*. Specifically, at step ⑤ the reconstructed image is passed to the Encoder to acquire the reconstructed embedding Ⓑ, and then the difference between the integrated embedding Ⓐ and the reconstructed embedding Ⓑ is minimized. The second one is the *style loss*. Particularly, the means and variances of a few selected internal layers of the Encoder are computed for both the generated image and the original style image. The difference of these values of the two images is minimized. The Decoder optimization process is conducted on the original training dataset of target model M (under attack). Intuitively, the decoder is trained to understand the style differences so that it can decode feature style differences to realistic pixel space style differences, by observing the possible style differences.

When launching the attack ((b) in Fig. 2), a test input image is fed to the Encoder and goes through the same process as in the Decoder training phase. The key differences are that only one input image is required and the Decoder is fixed in this phase. Given a target model M (under attack), the reconstructed image is fed to M at step ⑥ to yield prediction Ⓔ. As the attack goal is to induce M to misclassify, the difference between prediction Ⓔ and a target output label (different from Ⓔ) is considered the *adversarial loss* for launching the attack. In addition, the content loss between Ⓐ and Ⓑ is also included. The attack updates the means and variances of embedding matrices at step ⑦ with respect to the adversarial loss and content loss. The final reconstructed image that induces the target model M to misclassify is a successful adversarial sample.

Definitions

In this section, we formally define feature space attack. Considering a typical classification problem, where the samples $\boldsymbol{x} \in \mathbb{R}^d$ and the corresponding label $y \in \{0, 1, \ldots, n\}$ jointly obey a distribution $\mathcal{D}(\boldsymbol{x}, y)$. Given a classifier $M : \mathbb{R}^d \to \{0, 1, \ldots, n\}$ with parameter θ. The goal of training is to find the best parameter $\arg\max_\theta P_{(\boldsymbol{x},y)\sim\mathcal{D}}[M(\boldsymbol{x}; \theta) = y]$. Empirically, people associate a continuous loss function $\mathcal{L}_{M,\theta}(\boldsymbol{x}, y)$, e.g. cross-entropy, to measure the difference between the prediction and the true label. And the goal is rewritten as $\arg\min_\theta \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}_{M,\theta}(\boldsymbol{x}, y)]$. We use $\mathcal{L}_M$ in short for $\mathcal{L}_{M,\theta}$ in the following discussion. In adversarial learning, the adversary can introduce a perturbation $\boldsymbol{\delta} \in \mathbb{S} \subset \mathbb{R}^d$ to a natural samples $(\boldsymbol{x}, y) \sim \mathcal{D}$. For a given sample $\boldsymbol{x}$ with label y, an adversary chooses the most malicious perturbation $\arg\max_{\boldsymbol{\delta}\in\mathbb{S}} \mathcal{L}_M(\boldsymbol{x} + \boldsymbol{\delta}, y)$ to make the classifier M predict incorrectly. Normally $\mathbb{S}$ is confined as an ℓ_p-ball centered on 0. In this case, the ℓ_p norm of pixel space differences measures the distance between adversarial samples (i.e., $\boldsymbol{x} + \boldsymbol{\delta}$ that causes misclassification) and the original samples. Thus we refer to this attack model as the *pixel space attack*. Most existing adversarial attacks fall into this category. Different from adding bounded perturbation in the pixel space, feature space attack applies perturbation in the feature space such that an encoder (to extract the feature representation of the benign input) and a decoder function (that translates perturbed feature values to a natural-looking image that closely resembles the original input in humans' perspective).

Formally, consider an encoder function $f : \mathbb{R}^d \to \mathbb{R}^e$ and a decoder function $f^{-1} : \mathbb{R}^e \to \mathbb{R}^d$. The former encodes a sample to an embedding $b \in \mathbb{R}^e$ and the latter restores an embedding back to a sample. A perturbation function $a \in \mathbb{A} : \mathbb{R}^e \to \mathbb{R}^e$ transforms a given embedding to another. For a given sample $\boldsymbol{x}$, the adversary chooses the best perturbation function to make the model M predict incorrectly.

$$\max_{a\in\mathbb{A}} \mathcal{L}_M[f^{-1} \circ a \circ f(\boldsymbol{x}), y]. \tag{1}$$

Functions f and f^{-1} need to satisfy additional properties to ensure the attack is meaningful. We call them the *wellness properties* of encoder and decoder.

Wellness of Encoder f. In order to get a meaningful embedding, there ought to exist a well-functioning classifier g based on the embedding, with a prediction error rate less than δ_1.

$$\begin{aligned} &\exists g : \mathbb{R}^e \to \{0, 1, \ldots, n\}, P_{(\boldsymbol{x},y)\sim\mathcal{D}}[g(f(\boldsymbol{x})) = y] \\ &\geq 1 - \delta_1, \text{ for a given } \delta_1. \end{aligned} \tag{2}$$

In practice, this property can be easily satisfied as one can construct g from a well-functioning classifier M, by decomposing $M = M_2 \circ M_1$ and take M_1 as f and M_2 as g.

Wellness of Decoder f^{-1}. Function f^{-1} is essentially a translator that translates what the adversary has done on the embedding back to a sample in $\mathbb{R}^d$. We hence require that for all possible adversarial transformation $a \in \mathbb{A}$, f^{-1} ought to retain what the adversary has applied to the embedding in the restored sample.

$$\begin{aligned} &\forall a \in \mathbb{A}, \text{ let } B^a = a \circ f(\boldsymbol{x}),\ E_{(\boldsymbol{x},y)\sim\mathcal{D}} \\ &||f \circ f^{-1}(B^a) - B^a||_2 \leq \delta_2, \text{ for a given } \delta_2. \end{aligned} \tag{3}$$

This ensures a decoded (adversarial) sample induce the intended perturbation in the feature space. Note that f^{-1} can always restore a benign sample back to itself. This is equivalent to requiring the identity function in the perturbation function set $\mathbb{A}$.

Given $(f, f^{-1}, \mathbb{A})$ satisfying the aforementioned properties, we define Eq. (1) as a feature space attack. Under this definition, pixel space attack is a special case of feature space attack. For an ℓ_p-norm ϵ-bounded pixel space attack, i.e., $\mathbb{S} = \{||\boldsymbol{\delta}||_p \leq \epsilon\}$, we can rewrite it as a feature-space attack. Let encoder f and decoder f^{-1} be an identity function and let $\mathbb{A} = \cup_{||\boldsymbol{\delta}||_p\leq\epsilon}\{a : a(\boldsymbol{m}) = \boldsymbol{m} + \boldsymbol{\delta}\}$.

One can easily verify the wellness of f and f^{-1}. Note that the stealthiness of feature space attack depends on the selection of $\mathbb{A}$, analogous to that the stealthiness of pixel space attack depending on the ℓ_p norm. Next, we demonstrate two stealthy feature space attacks.

Attack Design

Decoder Training. Our decoder design is illustrated in Fig. 2a. It is inspired by style transfer in (Huang and Belongie 2017). To train the decoder, we enumerate all the possible pairs of images in each class in the original training set and use these pairs as a new training set. We consider each pair has the same content features (as they belong to the same class) and hence their differences essentially denote style differences. By training the decoder on all possible style differences (in the training set) regardless the output classes, we have a general decoder that can recognize and translate arbitrary style perturbation. Formally, given a normal image $\boldsymbol{x}_p$ and another image $\boldsymbol{x}_q$ from the same class as $\boldsymbol{x}_p$, the training process first passes them through a pre-trained Encoder f (e.g., VGG-19) to obtain embeddings $B^p = f(\boldsymbol{x}_p), B^q = f(\boldsymbol{x}_q) \in \mathbb{R}^{H\cdot W\cdot C}$, where C is the channel size, and H and W are the height and width of each channel. For each channel c, the mean and variance are computed across the spatial dimensions (step ② in Fig. 2a). That is,

$$\begin{aligned} \mu_{B_c} &= \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} B_{hwc} \\ \sigma_{B_c} &= \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(B_{hwc} - \mu_{B_c})^2}\,. \end{aligned} \tag{4}$$

We combine the embeddings B^p, B^q from the two input images using the following equation:

$$\forall c \in [1, 2, ..., C], B^o_c = \sigma_{B^q_c}\Big(\frac{B^p_c - \mu_{B^p_c}}{\sigma_{B^p_c}}\Big) + \mu_{B^q_c}, \tag{5}$$

where B^o_c is the result embedding of channel c. Intuitively, the transformation retains the shape of B^p while enforcing the mean and variance of B^q. B^o is then fed to the Decoder f^{-1} for reconstructing the image with the content of $\boldsymbol{x}_p$ and the style of $\boldsymbol{x}_q$ (steps ③ & ④ in Fig. 2a). In order to generate a realistic image, the reconstructed image is passed to Encoder f to acquire the reconstructed embedding $B^r = f \circ f^{-1}(B^o)$ (step ⑤). The difference between the combined embedding

B^o and the reconstructed embedding B^r, called the *content loss*, is minimized using the following equation during the Decoder training:

$$\mathcal{L}_{\text{content}} = ||B^r - B^o||_2. \tag{6}$$

Note that the similarity between the input and the output is implicitly ensured by the fact that the encoder is relatively shallow and well-trained. In addition, some internal layers of Encoder f are selected, whose means and variances (computed by Equation 4) are used for representing the style of input images. The difference of these values between the style image $\boldsymbol{x}_q$ and the reconstructed image $\boldsymbol{x}_r$, called the *style loss*, is minimized when training the Decoder. It is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{style}} = &\sum_{i \in L} ||\mu(\phi_i(\boldsymbol{x}_q)) - \mu(\phi_i(\boldsymbol{x}_r))||_2 \\ &+ \sum_{i \in L} ||\sigma(\phi_i(\boldsymbol{x}_q)) - \sigma(\phi_i(\boldsymbol{x}_r))||_2\end{aligned} \tag{7}$$

where $\phi_i(\cdot)$ denotes layer i of Encoder f and L the set of layers considered. In this paper, L consists of conv1_1, conv2_1, conv3_1 and conv4_1 for the ImageNet dataset, and conv1_1 and conv2_1 for the CIFAR-10 and SVHN datasets. $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and the variance, respectively. The Decoder training is to minimize $\mathcal{L}_{\text{content}} + \mathcal{L}_{\text{style}}$.

Two Feature Space Attacks. Recall in the attack phase (Fig. 2b), the encoder and decoder are fixed. The style features of a benign image are perturbed while the content features are retained, aiming to trigger misclassification. The pre-trained decoder then translates the perturbed embedding back to an adversarial sample. During perturbation, we focus on minimizing two loss functions. The first one is the adversarial loss $\mathcal{L}_M$ whose goal is to induce misclassification. The second one is similar to the content loss in the Decoder training (Eq. 6). Intuitively, although the decoder is trained in a way that it is supposed to decode with minimal loss, arbitrary style perturbation may still cause substantial loss. Hence, such loss has to be considered and minimized during style perturbation.

With two different sets of transformations $\mathbb{A}$, we devise two respective kinds of feature space attacks, *feature augmentation attack* and *feature interpolation attack*. For feature augmentation attack, attacker can change both the mean and standard deviation of each channel of the benign embedding independently. The boundary of increments or decrements are set by ℓ_∞-norm under logarithmic scale (to achieve stealthiness). Specifically, given two perturbation vectors $\boldsymbol{\tau}^\mu$ for the mean and $\boldsymbol{\tau}^\sigma$ for the variance, both have the same dimension C as the embedding (denoting the C channels) and are bounded by ϵ, the list of possible transformations $\mathbb{A}$ is defined as follows.

$$\begin{aligned}\mathbb{A} = \cup_{||\boldsymbol{\tau}^\sigma||_\infty \leq \epsilon \text{ and } ||\boldsymbol{\tau}^\mu||_\infty \leq \epsilon,\ \boldsymbol{\tau}^\sigma \text{ and } \boldsymbol{\tau}^\mu \in \mathbb{R}^C} \\ \left\{ a : a(B)_{h,w,c} = e^{\boldsymbol{\tau}^\sigma_c}(B_{h,w,c} - \mu_{B_c}) + e^{\boldsymbol{\tau}^\mu_c}\mu_{B_c} \right\}\end{aligned} \tag{8}$$

Note that μ_B denotes the means of embedding B for the C channels. The subscript c denotes a specific channel. The transformation essentially enlarges the variance of the embedding at channel c by a factor of $e^{\boldsymbol{\tau}^\sigma_c}$ and the mean by a factor of $e^{\boldsymbol{\tau}^\mu_c}$.

For the feature interpolation attack, the attacker provides k images as the style feature prototypes. Let $\mathbb{S}_\mu, \mathbb{S}_\sigma$ be the simplex determined by $\cup_{i \in [1,2,\ldots,k]} \mu_{f(\boldsymbol{x}_i)}$ and $\cup_{i \in [1,2,\ldots,k]} \sigma_{f(\boldsymbol{x}_i)}$ respectively. The attacker can modify the vectors of μ_B and σ_B to be any point on the simplex.

$$\mathbb{A} = \cup_{\substack{\sigma_i \in \mathbb{S}_\sigma \\ \mu_i \in \mathbb{S}_\mu}} \left\{ a : a(B)_{h,w,c} = \sigma_i \cdot \frac{B_{h,w,c} - \mu_{B_c}}{\sigma_{B_c}} + \mu_i \right\} \tag{9}$$

Intuitively, it enforces a style constructed from an interpolation of the k style prototypes. Our optimization method is a customized iterative gradient method with gradient clipping (see Appendix).

Evaluation

Three datasets are employed in the experiments: CIFAR-10 (Krizhevsky et al. 2009), ImageNet (Russakovsky et al. 2015) and SVHN (Netzer et al. 2011). The feature space attack settings can be found in Appendix . We use 7 state-of-the-art detection and defense approaches to evaluate the proposed feature space attack. Detection approaches aim to identify adversarial samples while they are provided to a DNN. They often work as an add-on to the model and do not aim to harden the model. We use two state-of-the-art adversarial example detection approaches proposed by (Roth, Kilcher, and Hofmann 2019) and (Papernot and McDaniel 2018) to test our attack. Defense approaches, on the other hand, harden models such that they are robust against adversarial example attacks. Existing state-of-the-art defense mechanisms either use adversarial training or certify a bound for each input image. We adopt 5 state-of-the-art defense approaches in the literature (Madry et al. 2018; Zhang et al. 2019; Xie et al. 2019; Song et al. 2019; Lecuyer et al. 2019a) for evaluation. Note that while these techniques are intended for pixel space attacks, their effectiveness for our attack is unclear. We are unaware of any detection/defense techniques for the kind of feature attacks we are proposing.

Quality of Feature Space Adversarial Examples by Human Study and Distance Metrics

In the first experiment, we conduct a human study to measure the quality of feature space attack samples. We follow the same procedure as in (Zhang, Isola, and Efros 2016; Bhattad et al. 2020). Users are given 50 pairs of images, each pair consisting of an original image and its transformed version (by feature space attack). They are asked to choose the realistic one from each pair. The images are randomly selected and used in the following trials. Each pair appears on screen for 3 seconds, and is evaluated by 10 users. Every user has 5 chances for practice before the trials begin. In total, 110 users completed the study. We repeat the same study for different feature space attack scales on ResNet-50 as shown in Fig. 3. On average, 41.9% of users choose our adversarial samples over original images. This indicates that the feature space attack is largely imperceptible.

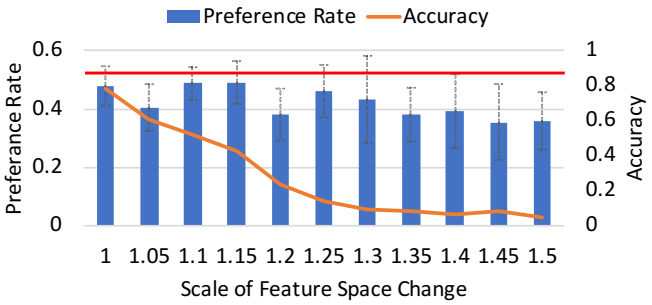


Figure 3: Human preference evaluation. The left y-axis (and blue bar) represents the percentage of user preference towards feature space attack images. The right y-axis (and orange line) denotes the test accuracy of models under feature space attack. The x-axis presents the scale of feature space perturbation e^{ϵ} in Eq. (8). The horizontal red line denotes where users cannot distinguish between adversarial samples and original images.

We also carry out a set of human studies to qualitatively measure images generated by different attacks: PGD (Madry et al. 2018), feature space attack, semantic attack (Bhattad et al. 2020), HSV attack (Hosseini and Poovendran 2018), manifold attack (Stutz, Hein, and Schiele 2018) and art attack (Prabhu and UnifyID 2018). The results are shown in Table 1. The first columns show the two attacks in comparison. The second column presents the human preference rate. The third column is the attack success rate. In the first table, the two attacks are conducted on a model with the denoise(t,1) defense (Xie et al. 2019) (in order to avoid 100% attack success rate). In the following four tables, the attacks are performed on a ResNet-50. We observe that the quality of feature space attack samples is comparable to that of PGD attack and the former has a higher attack success rate. Feature space attack also outperforms semantic attack, HSV attack, manifold attack and art attack in visual quality while achieving a higher/comparable attack success rate. That is, 67% or more users prefer feature space attack samples to the others. The comparison with art attack supports the benefit of leveraging feature-space during attack.The comparison with manifold attack stresses the importance of manipulating secondary features. The generated images by these attacks and comparison details can be seen in Appendix E.

We study the ℓ-norm distances in both the pixel space and the feature space for both pixel space attacks and feature space attacks. We observe that in the pixel space, the introduced perturbation by feature space attack is much larger than that of the PGD attack. In the feature space, our attack has very similar distances as PGD. Fig. 1 and 4 (in Appendix) show that the adversarial samples have only style differences that are natural or even human imperceptible. Detailed discussion can be found in Appendix C.We also study the characteristics of the adversarial samples generated by different feature space attacks and attack settings. Please see Appendix D.

	Pref.	Succ.
PGD	60	58
Feature	40	88

	Pref.	Succ.
Semantic	33	100
Feature	67	100

	Pref.	Succ.
HSV	20	64
Feature	79	100

	Pref.	Succ.
Manifold	27	100
Feature	73	100

	Pref.	Succ.
Art	11	100
Feature	89	100

Table 1: Human preference and success rate for different attacks. Feature represents feature space attack.

Attack against Detection Approaches

We use two state-of-the-art adversarial sample detection approaches "The Odds are Odd" (O2) (Roth, Kilcher, and Hofmann 2019) [2] and feature-space detection method "Deep k-Nearest Neighbors" (DkNN) (Papernot and McDaniel 2018).

O2 detects adversarial samples by adding random noise to input images and observing activation changing at a certain layer of a DNN. Specifically, O2 uses the penultimate layer (before the logits layer) as the representation of input images. It then defines a statistical variable that measures pairwise differences between two classes computed from the penultimate layer. The authors observed that adversarial samples differ significantly from benign samples regarding this variable when random noise is added. By performing statistical test on this variable, O2 is able to detect PGD attacks (Madry et al. 2018) with over 99% detection rate on CIFAR-10 with bound $\ell_\infty = 8/255$ and on ImageNet with $\ell_\infty = 2/255$. It also has over 90% detection rate against PGD and C&W (Carlini and Wagner 2017) attacks under ℓ_2 metric on CIFAR-10.

Table 2 shows the results of O2 on detecting different input samples. The first two columns are the datasets and models used for evaluation. The third column denotes the prediction accuracy of models on normal inputs. The following three columns present the detection rate of O2 on normal inputs, PGD adversarial samples, and feature space adversarial samples, respectively. The detection rate on normal inputs indicates that O2 falsely recognizes normal inputs as adversarial, which are essentially false positives. We can observe that O2 can effectively detect PGD attack on both datasets, but fails to detect feature space attack. Particularly, O2 has only 0.04% detection rate on CIFAR-10, which indicates that *O2 can be evaded by feature space attack*. As for ImageNet, O2 can detect 25.30% of feature space adversarial samples but at the cost of a 19.20% false positive rate[3]. The results show that O2 is ineffective against feature space attack.

Table 3 shows the results for Deep K Nearest Neighbour.

[2] O2 is recently bypassed by (Hosseini, Kannan, and Poovendran 2019), where the attacker already knows the existence of the defense. In our case, however, we are able to evade the detection method without knowing its existence or mechanism.

[3] The parameters used for ImageNet are not given in the original paper. We can only reduce to this false positive rate after parameter tuning.

Dataset	Accuracy	Detection Rate		
		Normal	PGD	Feature Space
CIFAR-10	91.95	0.95	99.61	**0.04**
ImageNet	75.20	19.20	99.40	**25.30**

Table 2: O2 detection rate on normal inputs and adversarial samples. We use ResNet-18 on CIFAR-10 and ResNet-50 on ImageNet.

Dataset	Model	Accuracy	Detection Rate	
			PGD	Feature Space
CIFAR-10	CNN+MLP	53.93	3.92	**1.95**
	ResNet-18	81.51	11.32	**5.42**

Table 3: DkNN detection rate on normal inputs and adversarial samples.

Attack	SVHN	CIFAR-10		
	Adaption	Madry	TRADES	Pixel-DP [4]
None	84.84	77.84	84.97	44.3
PGD	52.84	41.43	54.02	30.7
Decoder	84.81	77.35	84.01	50.0
Feature Space	**2.56**	**7.05**	**8.64**	**0.0**

Attack	ImageNet		
	Denoise (t,1)	Denoise (u,1)	Denoise (u,5)
None	61.25	61.25	78.12
PGD	42.60	12.50	27.15
Decoder	64.68	64.00	82.37
Feature Space	**11.41**	**1.25**	**1.25**

Table 4: Evaluation of adversarial attacks against various defense approaches.

Due to memory limits, we only test on the CIFAR-10 dataset. The second column denotes models employed for evaluation including the default one used in the original paper (CNN+MLP). The third column shows model accuracy on benign inputs. The last two columns present detection rate on PGD and feature space attacks. We can observe that DkNN has much lower detection rate on feature space attack compared to PGD, despite the fact that DkNN uses feature space data for detecting adversarial samples.

Attack against Defense Approaches

We evaluate our feature space attack on 5 state-of-the-art adversarial training approaches: Madry (Madry et al. 2018), TRADES (Zhang et al. 2019), Denoise (Xie et al. 2019), Adaption (Song et al. 2019), and Pixel-DP (Lecuyer et al. 2019b). For Denoise, the original paper only evaluated on targeted attacks. We conduct experiments on both targeted and untargeted attacks. We use Denoise (t,1) to denote the top-1 accuracy of hardened model on targeted attack and Denoise (u,5) the top-5 accuracy on untargeted attack. We launch the PGD ℓ_∞ attack as well as our feature space attack on the four defense approaches. The results are shown in Table 4. The first column denotes attack methods, where "None" presents the model accuracy on benign inputs and "Decoder" denotes the samples directly generated from the decoder without any feature space perturbation. The latter is to show that the Decoder can generate faithful and natural images from embeddings. The following columns show different defense approaches (second row) applied on various datasets (first row). We can see that the PGD attack can reduce model accuracy to some extent when defense mechanisms are in place. *Feature space attack, on the other hand, can effectively reduce model accuracy down to less than 12%, and most results are one order of magnitude smaller than PGD.* Especially, model accuracy on ImageNet is only 1.25% when using untargeted attack, even in the presence of the defense technique.

From the aforementioned results, we observe that existing pixel space detect/defense techniques are largely ineffective as they focus on pixel space. While it may be possible to extend some of these techniques to protect feature space, the needed extension remains unclear to us at this point. We hence leave it to our future work. For example, it is unclear how to extend O2, which leverages the penultimate layer to detect anomaly and hence should have been effective for our attack in theory.

Towards Feature Space Adversarial Training. We conduct a preliminary study on using feature space attack to perform adversarial training. For comparison, we also perform the PGD adversarial training and use semantic attack to perform adversarial training. We evaluate the adversarially trained models against feature space (FS) attack, HSV attack, semantic (SM) attack, and PGD attack, with the first three in the feature space. We find that PGD adversarial training is most effective against PGD attack (55% attack success rate reduction) and has effectiveness against SM attack too (22% reduction), but not FS or HSV attack. Feature space adversarial training can reduce the FS attack success rate by 27% and the HSV attack by 13%, but not others. Adversarial training using semantic attack can reduce semantic attack success rate by 34% and PGD by 38%, but not others. This suggests that different attacks aim at different spaces and the corresponding adversarial trainings may only enhance the corresponding target spaces. Note that the robustness improvement of feature space adversarial training is not as substantial as PGD training in the pixel space. We believe that it is because either our study is preliminary and more setups need to be explored; or, feature space adversarial training may be inherently harder and demand new methods. We will leave it to our future study. More details (e.g., ablation study) can be found in Appendix F.

Conclusion

We propose feature space adversarial attack on DNNs. It is based on perturbing style features and retaining content features. Such attacks inject natural style changes to input images to cause model misclassification. Since they usually cause substantial pixel space perturbations and existing detection/defense techniques are mostly for bounded pixel space attacks, these techniques are not effective for feature space attacks.

Acknowledgments

This research was supported, in part by NSF 1901242 and 1910300, ONR N000141712045, N000141410468 and N000141712947, and IARPA TrojAI W911NF-19-S-0012. Any opinions, findings, and conclusions in this paper are those of the authors only and do not necessarily reflect the views of our sponsors.

Ethics Statement

Our work illustrates a new vulnerable aspect of deep learning models. As the vulnerability mainly lies in the internal style related feature space, most existing defense and detection techniques are not effective. Our work will provide motivation and insights for better protecting deep learning applications.

References

Bhattad, A.; Chong, M.; Liang, K.; Li, B.; and Forsyth, D. 2020. Unrestricted Adversarial Examples via Semantic Manipulation. In *International Conference on Learning Representations ICLR Conference, 2020*.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of 38th IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2017. A learned representation for artistic style. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.

Ebrahimi, J.; Rao, A.; Lowd, D.; and Dou, D. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 31–36.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423.

Hosseini, H.; Kannan, S.; and Poovendran, R. 2019. Are Odds Really Odd? Bypassing Statistical Detection of Adversarial Examples. *CoRR* abs/1907.12138. URL http://arxiv.org/abs/1907.12138.

Hosseini, H.; and Poovendran, R. 2018. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1614–1619.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1501–1510.

Inkawhich, N.; Wen, W.; Li, H.; and Chen, Y. 2019. Feature Space Perturbations Yield More Transferable Adversarial Examples. 7059–7067. doi:10.1109/CVPR.2019.00723.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 694–711.

Krizhevsky, A.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Laidlaw, C.; and Feizi, S. 2019. Functional adversarial attacks. In *Advances in Neural Information Processing Systems*, 10408–10418.

Lecuyer, M.; Atlidakis, V.; Geambasu, R.; Hsu, D.; and Jana, S. 2019a. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, 656–672. IEEE.

Lecuyer, M.; Atlidakis, V.; Geambasu, R.; Hsu, D.; and Jana, S. 2019b. Certified robustness to adversarial examples with differential privacy. In *Proceedings of 40th IEEE Symposium on Security and Privacy (SP)*.

Li, C.; and Wand, M. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision (ECCV)*, 702–716.

Li, S.; Neupane, A.; Paul, S.; Song, C.; Krishnamurthy, S. V.; Chowdhury, A. K. R.; and Swami, A. 2019. Adversarial perturbations against real-time video classification systems. In *Proceedings of 26th Annual Network and Distributed System Security Symposium (NDSS)*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Diversified texture synthesis with feed-forward networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3920–3928.

Ma, S.; Liu, Y.; Tao, G.; Lee, W.-C.; and Zhang, X. 2019. NIC: Detecting Adversarial Samples with Neural Network Invariant Checking. In *Proceedings of 26th Annual Network and Distributed System Security Symposium (NDSS)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of 6th International Conference on Learning Representations (ICLR)*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Papernot, N.; and McDaniel, P. D. 2018. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. *CoRR* abs/1803.04765. URL http://arxiv.org/abs/1803.04765.

Prabhu, V. U.; and UnifyID, J. W. 2018. Art-attack ! On style transfers with textures , label categories and adversarial examples. In *The Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security (CV-COPS 2018)*. URL http://vision.soic.indiana.edu/bright-and-dark-workshop-2018/cvcops_2018_extended_abstracts/art_attack.pdf.

Qin, Y.; Carlini, N.; Goodfellow, I.; Cottrell, G.; and Raffel, C. 2019. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 5231–5240.

Roth, K.; Kilcher, Y.; and Hofmann, T. 2019. The Odds are Odd: A Statistical Test for Detecting Adversarial Examples. In *International Conference on Machine Learning (ICML)*, 5498–5507.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3): 211–252.

Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2019. Improving the Generalization of Adversarial Training with Domain Adaptation. In *International Conference on Learning Representations (ICLR)*.

Song, Y.; Shu, R.; Kushman, N.; and Ermon, S. 2018. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, 8312–8323.

Stutz, D.; Hein, M.; and Schiele, B. 2018. Disentangling Adversarial Robustness and Generalization. *CoRR* abs/1812.00740. URL http://arxiv.org/abs/1812.00740.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing Properties of Neural Networks. In *International Conference on Learning Representations (ICLR)*.

Tao, G.; Ma, S.; Liu, Y.; and Zhang, X. 2018. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems (NeurIPS)*, 7717–7728.

Xie, C.; Wu, Y.; Maaten, L. v. d.; Yuille, A. L.; and He, K. 2019. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 501–509.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. *CoRR* abs/1605.07146. URL http://arxiv.org/abs/1605.07146.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *International Conference on Machine Learning (ICML)*, 7472–7482.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful Image Colorization. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, 649–666. Springer. doi:10.1007/978-3-319-46487-9_40. URL https://doi.org/10.1007/978-3-319-46487-9_40.