RegionSparse: Leveraging Sparse Coding and Object Localization to Counter Adversarial Attacks

Yunjian Zhang[1,2], Yanwei Liu[1*], Liming Wang[1], Zhen Xu[1], Qiuqing Jin[1,2]
[1]*Institute of Information Engineering, Chinese Academy of Sciences*
[2]*School of Cyber Security, University of Chinese Academy of Sciences*
{zhangyunjian,liuyanwei,wangliming,xuzhen,jinqiuqing}@iie.ac.cn

Abstract—**Although deep neural networks have demonstrated exceptional performance in substantial computer vision tasks, they can be easily confused by carefully generated adversarial examples. Via a novel technique we call activation visualization, the particular characteristics of adversarial examples are analyzed in this paper. Observing that the dominant features of adversarial examples are distributed over a high-dimensional space, we propose a defense framework named RegionSparse that projects the images into a low-dimensional space to remove the influence of the adversarial perturbations on the performance of deep neural networks. In RegionSparse, after training a robust global dictionary, the region where pixels are highly related to classification is firstly located by an object localization mechanism, then the sparse coding is performed on the located object region, together with a perturbation suppression for the remaining region. Extensive experiments on ImageNet dataset for gray-box, black-box, and transferred attacks are performed and the results show that RegionSparse can eliminate up to 90% attacks delivered by strong attacks including *Momentum Iterative Fast Gradient Sign Method* and *Carlini-Wagner's* L_2 attack.**

Index Terms—**Adversarial examples, image classification, deep neural networks**

I. Introduction

Recent studies have revealed that deep neural networks (DNNs) are particularly vulnerable to well-tuned perturbations, and these perturbed instances are called *adversarial examples*. Worse still, these perturbations are normally imperceptible to human [1], [2], leading to severe consequences in many DNN-based applications, especially in security-critical systems, be it face recognition [3], autonomous navigation [4], or robotic [5] systems.

To deal with adversarial examples in DNNs, researchers have proposed several defense approaches. Defense approaches against adversarial examples typically fall into two categories: *model-specific* and *input-specific*. Model-specific approaches generally modify the architecture or training scheme of models to improve their robustness [6], [7]. One of the most effective model-specific approaches is adversarial training [8]. It feeds adversarial examples themselves into the training set. The performance of these approaches is limited because they make strong assumptions about the type of attacks [9]. Moreover, they tend to complicate models, and have not shown satisfying generalization ability on different models. In contrast, input-specific approaches aim to remove adversarial perturbations from the input data while keeping the network fixed. They require no knowledge about the target networks, and it makes them easily deployable to any DNN models. Existing input-specific approaches, including JPEG compression [10], [11], feature squeezing [12], total variance minimization [9], and ComDefend [13], have achieved acceptable results in previous works. However, these approaches did not explore the unique space where the adversarial perturbations express as the meaningful adversarial features that are critical to the misclassification of DNNs. Instead, they simply considered these perturbations as an ordinary type of noises (e.g. Gaussian noise, impulse noise), and attempted to utilize universal image denoising techniques to defend adversarial attacks, which seriously restricts their performances.

Inspired by the observation from activation visualization that adversarial images usually have more features in high-dimensional space than benign ones, we propose to use sparse coding to remove adversarial perturbations by projecting them to a low-dimensional space. This work offers a fresh perspective on understanding and countering adversarial examples, and to the best of our knowledge, none of the previous works have considered the high-dimensional features implicit in adversarial examples.

Our contributions are summarized as follows:

- We deeply analyze the high-dimensional characteristics of adversarial examples with activation visualization, and theoretically prove the feasibility of sparse coding to reduce the impact of adversarial perturbations.
- We propose a defense framework RegionSparse. This framework can be decomposed into two stages: *dictionary learning* and *sparse coding*. In *dictionary learning* stage, a novel dictionary learning scheme is designed to capture the most essential and representative natural features from a collection of benign images augmented by image smoothing and patch normalization. In *sparse coding* stage, a flexible image decomposition method is proposed to balance the compression levels over different regions of the image by combining sparse coding with object localization, which tremendously improves the effectiveness of defense.
- Comprehensive experiments is performed on ImageNet dataset, and the results show that RegionSparse outper-

This work was supported in part by National Natural Science Foundation of China under Grant 61771469 and Ningbo Natural Science Foundation under Grant 2019A610109.

[*] Corresponding author

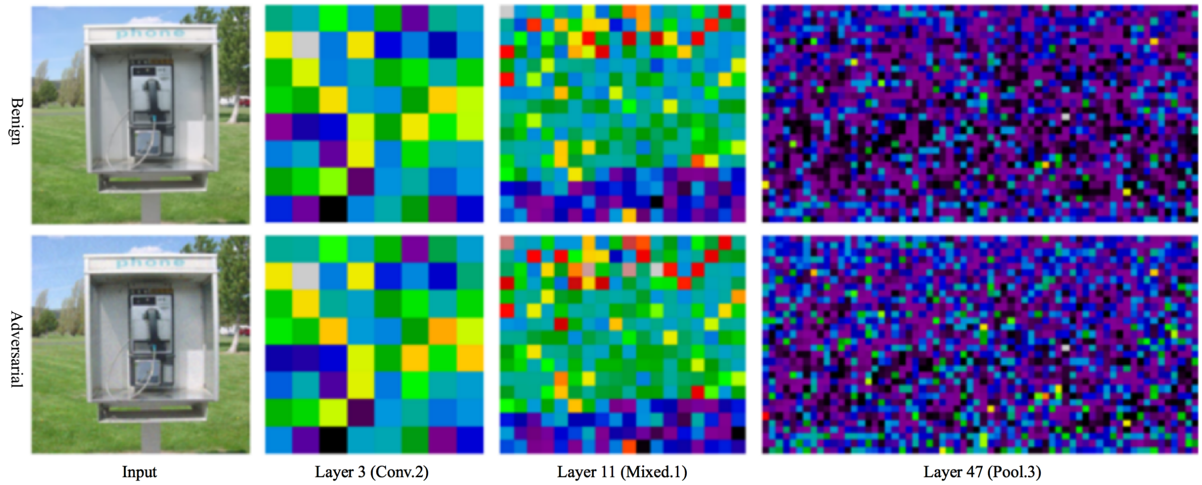

Fig. 1. Activation visualization of three different layers in Inception-v3 model. Color temperature map is used, in which lighter color indicates higher activation.

forms the state-of-the-art defense approaches. Furthermore, RegionSparse can be easily combined with adversarial training, which dramatically improve its capability for defending adversarial attacks.

II. Background

This section reviews the typical adversarial attacks and the popular input-specific defense approaches.

A. Adversarial Attacks

*a) **FGSM**:* FGSM [2] is the first successful one-step attack method. For a given image x, the adversarial examples are computed by

$$x^{adv} = x + \epsilon \cdot \operatorname{sign}(\nabla_x L(x, y)), \tag{1}$$

where ϵ governs the perturbation magnitude, y is the true label, and $L(\cdot)$ is the loss function used to train the network (e.g., cross-entropy loss).

*b) **BIM**:* BIM [14] generates adversarial examples by iteratively applying the FGSM method at a small step size:

$$x_m^{adv} = x_{m-1}^{adv} + \epsilon \cdot \operatorname{sign}(\nabla_{x_{m-1}^{adv}} L(x_{m-1}^{adv}, y)), \tag{2}$$

where $m = 1, ..., M$ indicates the m-th iteration, M is the maximum iteration number, $x_0^{adv} = x$, and $x^{adv} = x_M^{adv}$.

*c) **MI-FGSM**:* MI-FGSM [15] is another iterative version of FGSM attack. It improves FGSM by introducing a momentum term into gradient calculation:

$$g_m = \mu \cdot g_{m-1} + \frac{\nabla_x L(x_{m-1}^{adv}, y)}{\left\|\nabla_x L(x_{m-1}^{adv}, y)\right\|_1}, \tag{3}$$

where μ controls the influence of the previous gradient. The gradient is used to optimize the image

$$x_m^{adv} = x_{m-1}^{adv} + \epsilon \cdot \operatorname{sign}(g_m). \tag{4}$$

*d) **CW-L_2**:* CW-L_2 [16] formalizes the procedure of generating adversarial examples as an optimization problem selecting a good trade-off between the prediction accuracy and an L_2 penalty that determines the scale of perturbations.

B. Defenses

The classic image processing techniques used for removing adversarial perturbations are summarized as follows.

*a) **Total Variance Minimization (TVM)**:* TVM [9] randomly selects a subset of pixels from an image x, and reconstructs the “simplest” image x' that is consistent with the selected pixels by solving the following optimization problem:

$$\min_{x'} \|x' - x\|_2 + \lambda J(x'), \tag{5}$$

where λ denotes the trade-off between image similarity and total variance, and $J(x')$ is called *Total Variation*, defined by:

$$J(x') = \sum_{x'} |\nabla x'|. \tag{6}$$

*b) **Feature Squeezing**:* Xu et al. [12] proposed to deal with adversarial examples by “squeezing” out their features via local smoothing. Local smoothing usually runs a sliding window over each pixel in the image, and smooths it with its neighbors within the window. By making nearby pixels more similar, this method can effectively “squeezing” a amount of adversarial features out of the adversarial image and promotes the models to accurately classify the image.

*c) **JPEG Compression**:* JPEG compression is designed to reduce the imperceptible details in images, and it typically performs a simple quantization that can effectively remove small adversarial variations in pixel values from an image.

*d) **ComDefend**:* ComDefend [13] utilized an encoder-decoder architecture to compress the 24-bits image to 12-bits image to make the classifiers easier to simulate the image distribution. The model consists of a compression CNN and a reconstruction CNN, the former is used to reduce the bit

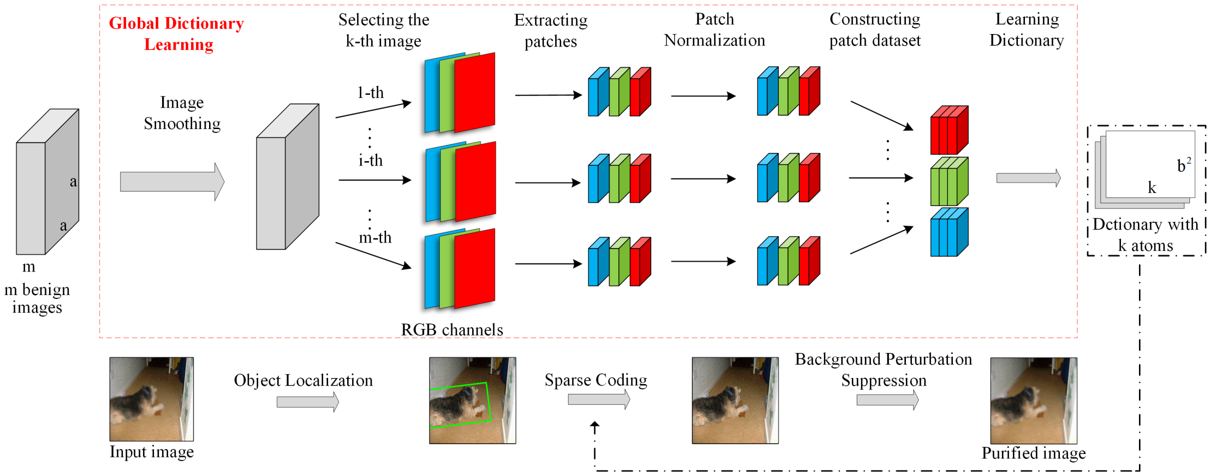

Fig. 2. Two stages of RegionSparse. The top flowchart describes the dictionary learning scheme, and the bottom one presents the sparse coding process. a and b denote the widths of the original image and the extracted patches, respectively.

depth of the input image from 24 bit to 12 bit, and the later is used to reconstruct a purified image from the compressed image. Noting that Gaussian noise is added to the compressed 12-bits image for better performance.

III. Understanding Adversarial Features with Activation Visualization

Achievements of feature visualization techniques [17], [18] motivate researchers to illustrate the differences between benign and adversarial images by analyzing their feature maps. For instance, [19] attempted to apply a two-sample test on the feature maps to distinguish adversarial examples from benign ones. Li et al. [20] trained a cascaded classifier with the feature maps to recognize adversarial examples. However, these methods require additional statistical analyses, increasing the computational complexity and lacking a visually intuitive insight into the adversarial features.

According to [21], each channel of convolutional filters can be seen as an individual feature detector, for it is sensitive to a specific feature. Therefore, from the perspective of DNNs, we believe that the number of activated feature detectors for an image directly reflects the features contained in the image. Motivated by this understanding, we propose a novel visualization technique that directly visualizes the number of activated detectors rather than the statistics of feature maps. Specifically, a max-pooling operation is applied on every feature map, and the size of the pooling kernel is equal to that of the corresponding feature map, which allows the activation of a feature detector to be represented as a single value. The activation value of a feature detector indicates the significance of the corresponding feature in the input image. With the aid of activation visualization, we gain a deeper and more intuitive insight into how adversarial examples affect DNNs.

Fig. 1 shows an example of activation visualization of three different layers in Inception-v3 model [22]. In the relative low layers (e.g. Conv.2, Mixed.1), the activation of the adversarial image does not clearly distinguish from that of the benign one. Nevertheless, the visualization result of the adversarial image in the relative deep layer (e.g. Pool.3) becomes apparently lighter than that of the benign image. This observation shows that the lower layers detect similar features from both benign and adversarial examples, while the deeper layers detect far more features from adversarial images than benign ones.

To further validate this observation, we quantify the total activation of a layer as the weighted sum of the activation value of the feature detectors in this layer. The total activation of l-th layer is formulated by

$$V_l = \sum_{(p_l^i-\gamma)>0} p_l^i \cdot \mathrm{sign}(p_l^i-\gamma), \tag{7}$$

where i denotes the i-th feature detector, p denotes the activation value of the detector, and γ is the activation threshold. Detectors with an activation value below γ are not considered activated.

We calculate the ratio of the total activation of adversarial images to that of benign images at each layer. In the layers whose indexes are lower than 30, the ratios are almost equal to 1.0, indicating that the similar features are lying in adversarial and benign images. Nevertheless, in the deeper layers, the ratios are increase rapidly, especially those close to the output layer, the ratios are up to 1.15, which suggests that adversarial images have much more features in these layers. [23] observed that shallow layers in the networks can detect angles and edges, while deep layers are sensitive to high-level features like outlines and objects. Because high-level features indicate more detailed and complex spatial structures, they can be considered high-dimensional. Therefore, adversarial images can be considered to have more high-dimensional features than benign ones. It explains how the adversarial examples mislead DNNs: they generate vast high-dimensional features to activate superabundant detectors and ultimately confuse the networks.

Based on the analyses above, we propose to utilize sparse coding to compress redundant high-dimensional features of

TABLE I
TABLE OF IMPORTANT NOTATIONS.

Notation	Meaning	Notation	Meaning
x	natural image	x'	adversarial image
K	sparsity	Ψ	sparse dictioanry
f	DNN model	D	benign patch dataset
F	compression process	C	compression function
x_o	object region	x_b	background region
θ	tradeoff between x_o and x_b	h	degree of filtering

adversarial images to correct the predictions for them.

IV. THE PROPOSED APPROACH: REGIONSPARSE

Fig. 2 shows the whole framework of RegionSparse. In the dictionary learning stage, image smoothing is firstly applied on a collection of benign images. After that, abundant patches are extracted from the smoothed images, and then we normalize the patches extracted from the same images. Further, we construct a patch dataset, and learn a global dictionary with it.

In the sparse coding stage, we utilize object localization technique to locate the region to be sparsely coded in the image. Then, sparse coding is performed in the located region with the dictionary obtained from the dictionary learning stage. Finally, we apply background perturbation suppression on the remaining area of the image. The processed image is then fed into DNN to be classified.

A. Rationale of Sparse Coding

For a natural image x, its corresponding adversarial example x' can be formulated by $x' = x + e$, where e denotes the adversarial perturbation. The sparse representation $S_K(\cdot)$ is defined as

$$S_K(x) = H_K(\Psi^T x), \tag{8}$$

where the function $H_K(\cdot)$ enforces K-sparsity by retaining the K coefficients largest in magnitude and zeroing out the rest.

Assuming that sparse coding is operated in the *high SNR regime*, according to [24], the additive perturbations do not shift the K-dimensional subspace of x, then we can get

$$H_K(\Psi^T(x+e)) = H_K(\Psi^T x) + e', \tag{9}$$

where $e'_k = \begin{cases} \psi_k^T e & \text{if k} \in S_K(x) \\ 0 & \text{otherwise} \end{cases}$. The output of sparse coding is:

$$x'_s = x_s + \Psi^T e. \tag{10}$$

The energy function of adversarial attack is defined as the norm between the prediction of natural image and adversarial image:

$$E(x') = \|f(x') - f(x)\|_2^2 + ([f(x')] - [f(x)])^2, \tag{11}$$

where the right term is a constraint condition for avoiding energy confusion (e.g. the energy of setting $f(x) = 0.6, f(x') = 0.4$ is smaller than that of setting $f(x) = 0.6, f(x') = 1.0$). The gradient for energy function is calculated by

$$\frac{\partial E(x')}{\partial x'} = 2(f(x') - f(x)) \cdot \frac{\partial f(x')}{\partial x'}. \tag{12}$$

It is worth noting that DNNs tend to behave linear in high dimension [2], [25], so we assume that $f(x') = W^T x'$, where W is a generalized parameter matrix of the DNN, and the rank of W is greater than the number of labels N, so $\frac{\partial f(x')}{\partial x'} = W^T \neq 0$. Obviously, the minimum of the energy function occurs when $f(x) = f(x')$, and we can get the optimal solution:

$$f(x_s + \Psi^T e) = f(x_s), \tag{13}$$

that is, $\Psi^T e = 0$, which means that the space spanned from the dictionary Ψ is orthogonal to the space occupied by adversarial perturbations. In general, sparse coding projects the adversarial perturbations e to a subspace of the space spanned from the overcomplete normalization dictionary. Noting that e has a higher dimension than the dictionary space, so $\left\|\Psi^T e\right\| < \|e\|$, resulting in a weak impact for the model accuracy of the adversarial perturbations.

B. Robust Global Dictionary Learning

Our learning scheme is based on *Online Dictionary Learning* [26] that is widely applied for image denoising. For denoising tasks, it is common to utilize the noised images to learn dictionaries when the original images are not available. However, this approach is not suitable for adversarial image countering tasks, because it needs to learn individual dictionaries for every adversarial image, and leads to dictionaries containing adversarial features, resulting in low classification accuracy. We overcome this problem by learning a global dictionary with the following innovations.

a) ***Constructing Benign Patch Dataset.:*** In our scheme, we randomly select substantial patches from a benign image dataset to learn a global dictionary. Thus, there is no need to retrain dictionaries when processing adversarial images. For m benign images $[x_1, ..., x_m]$, we extract n patches from the image x_k processed by *Image Smoothing*. Then the n patches are flattened and concatenated to a matrix

$$\mathbf{P_k} = [x_k^1, ..., x_k^n]^T. \tag{14}$$

After that, we apply *Patch Normalization* on $\mathbf{P_k}$ and concatenate them to the final benign patch dataset:

$$\mathbf{D} = [\mathbf{P_1}, ..., \mathbf{P_m}]^T. \tag{15}$$

b) ***Image Smoothing.:*** In order to acquire a highly representative dictionary, we should eliminate individual features from different benign images by filtering out high frequency elements in images. All blurring methods are alternative. In this work, we select Non-local median (NLM) [27] to smooth benign images.

NLM smooths a pixel over the whole image area. For a given patch $I(i)$ with pixel i as the center, NLM finds several similar patches in the image and smooths the pixel i by the central pixels of these patches. The weight corresponding to the central pixel i and j can be obtained by:

$$\omega(i,j) = e^{-\frac{\|I(i)-I(j)\|_2^2}{h}}, \tag{16}$$

where h controls the amount of details preserved after filtering.

*c) **Patch Normalization.**:* A problem in patch extraction is that the patches come from different regions of images, resulting in the nonidentical patch distributions. To deal with it, we utilize *z-score* normalization to normalize patches that are extracted from the same images.

For the $\mathbf{P_k}$ consisting of n patches sized of b×b, we can take it as a b×b-dimensional matrix, that is, $\mathbf{P_k} = (\mathbf{P_k^{(1)}}, ..., \mathbf{P_k^{(b\times b)}})$. Each dimension is normalized as

$$\mathrm{Nor}(\mathbf{P_k^{(i)}}) = \frac{\mathbf{P_k^{(i)}} - \mathrm{E}[\mathbf{P_k^{(i)}}]}{\sqrt{Var[\mathbf{P_k^{(i)}}]}}, \tag{17}$$

where the expectation $\mathrm{E}[\mathbf{P_k}]$ and variance $Var[\mathbf{P_k}]$ are computed over each dimension.

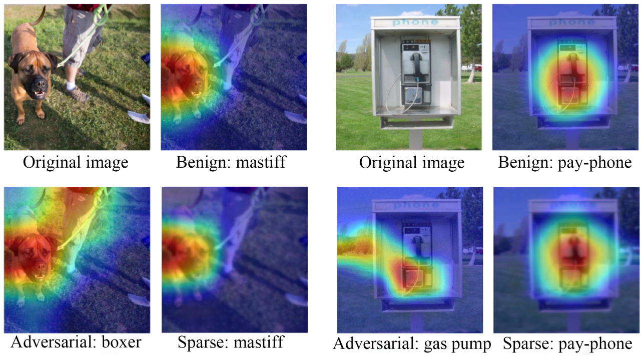

Fig. 3. Visual attention maps of different images. Attention of benign images is concentrated on the object regions, while that of adversarial ones deviates from the normal areas. RegionSparse can recover the attention misled by the adversarial attacks.

C. RegionSparse Coding

We have so far learnt a robust dictionary containing basic features of natural images. Before utilizing this dictionary to process images, we should consider a balance among the compression levels over different regions of the image, because high compression level can reduce more adversarial perturbations, but too much compression could reduce the accuracy on benign images by introducing numerous artifacts. The compression process can be formulated by

$$F(x) = \sum_i w_i C(x_i), \tag{18}$$

where x_i is the i-th region of the image, w_i controls the compression level for the i-th region. We divide the image into object region x_o and background region x_b, then the compression process can be simplified to

$$F(x) = \theta C(x_o) + (1-\theta)C(x_b), \tag{19}$$

In this paper, we use *object localization* and *background perturbation suppression* to decide a satisfying θ.

*a) **Object Localization.**:* We visualize the attention areas that contribute most to the classification for input images in Fig. 3, and notice that the pixels highly related to the prediction are gathered near the object instead of the background areas. This observation shows that denoising in the object regions may be effective to reduce the impact of adversarial examples to a large extent, which avoids applying sparse coding on the whole images and introduces less artifacts.

To select the region to be sparsely coded, we apply object localization mechanism on input images. Firstly, we adopt *Sobel* operator [28] on an image to calculate the gradients G_x of x axis and G_y of y axis. To preserve the region with high horizontal gradient and low vertical gradient, we calculate the image gradient $G = G_x - G_y$. Next, the image is binarized and the morphological operation is applied on the binary image. Finally, the remaining outline of the binary image is marked as the boundary of the object region.

*b) **Sparse Coding.**:* Sparse coding can be approximately seen as a linear decomposition of an image x by solving the following convex problem:

$$l(x,\Psi) = \min_{\alpha\in\mathbb{R}^{m\times k}} \frac{1}{2}\|x-\Psi\alpha\|_2^2 + \beta\|\alpha\|_1, \tag{20}$$

where m is the dimension of the image, β is a regularization parameter, Ψ is the trained dictionary with k atoms, in which each column represents an atom, and α is the sparse approximation of the original image.

Orthogonal Matching Pursuit (OMP) [29] is generally used to solve the problem above. Given a signal $\vec{y}$ and a dictionary Ψ, the decomposition of $\vec{y}$ described by Ψ is $\vec{y} = \Psi\vec{s} + \vec{\varepsilon}$, where $\vec{s}$ represents the coefficients and ε is the error. Then OMP minimizes the error by iteratively selecting the columns in the dictionary to represent $\vec{y}$.

*c) **Background Perturbation Suppression.**:* Although sparse coding has removed plentiful perturbations, a mild filtering is still required, in order to reduce the impact of perturbations distributed in the background region and avoid high-frequency jump between the sparsely coded and unprocessed regions. Considering the capability of bilateral filter to smooth image gradients while preserving edge features, we utilize it to suppress background perturbations in RegionSparse.

V. Experimental Results

We performed several experiments to test the efficacy of RegionSparse. Three scenarios were considered in the experiments:

- Gray-box scenario: The adversary has access to the model architecure and parameters but is unaware of the defense strategy.

- Black-box scenario: The adversary has no knowledge about the model.
- Transferred attack scenario: The adversary is expected to achieve attack with adversarial examples generated from another model.

We considered two types of attacks on these scenarios: L_2 attack (CW-L_2, MI-FGSM) and L_∞ attack (FGSM, BIM).

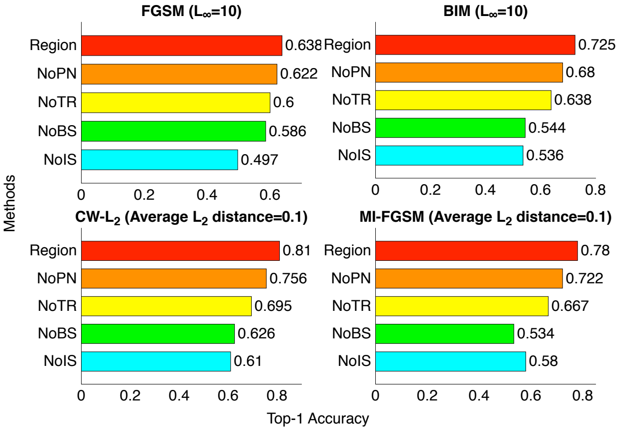

Fig. 4. Top-1 accuracy of Inception-v3 model tested on the sparsely coded images. Longer bar implies higher accuracy.

A. Experimental Setup

Our experiments were performed on Inception-v3 model with validation set in the ImageNet dataset. To remove the influence of the inherent inaccuracy of the model, we reconstructed the dataset with correctly classified images. We simulated the attacks using *CleverHans*. FGSM and BIM attacks were performed with $L_\infty \in [2, 22]$. For CW-L_2 attack, we used $\kappa = 0$, $\lambda_f = 0.1$, and the perturbations were multiplied by a constant $\tau \geq 1$ to alter their magnitude. The average L_2 distance of CW-L_2 and MI-FGSM were adjusted to $d \in [0.01, 0.2]$. Our code will be released on the website.

B. Visualizing the Attention of Network

Before analyzing the performance of RegionSparse, we tested its effectiveness in protecting the network attention area by characterizing images from the perspective of DNNs. Class Activation Mapping (CAM) [30] was applied here to localize the attention areas detected by DNNs.

Fig. 3(a)-(d) and Fig. 3(e)-(f) respectively show two groups of attention visualization results. Attention of benign images is concentrated on object areas (e.g. left: the dog, right: the phone), and it is the prerequisite of correct classifications. In contrast, adversarial images successfully mislead the attention of DNNs: the attention areas include irrelevant background regions, resulting in wrong predictions. Fig. 3(d) and (f) show that the attention of adversarial images processed by RegionSparse is similar to that of benign images, and it suggests that RegionSparse has ability to correct the classifications misled by adversarial attacks.

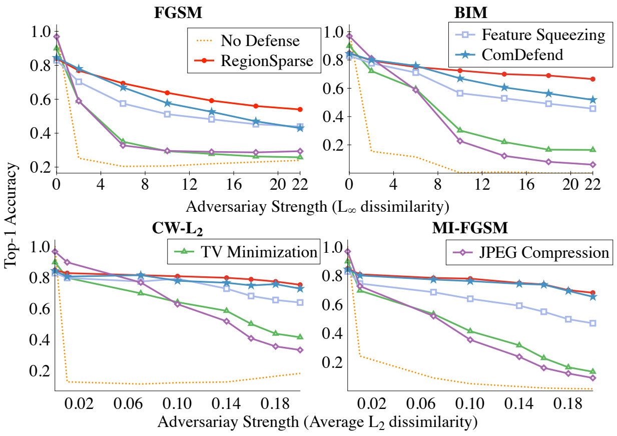

Fig. 5. Top-1 accuracy of Inception-v3 model tested on the transformed images produced by four kinds of attacks in a gray-box setting. The orange dotted line shows the accuracy on images without any defenses, and the attack strength of zero corresponds to benign images.

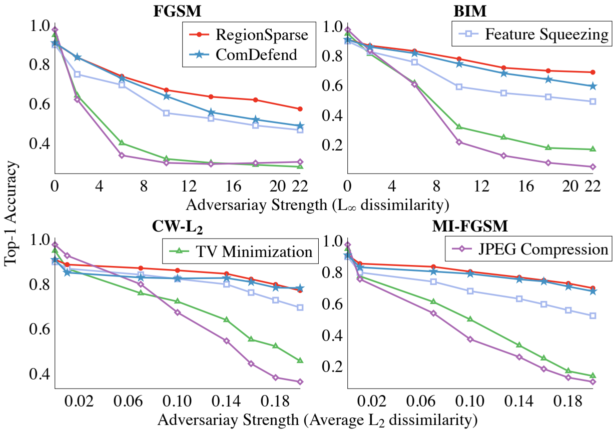

Fig. 6. Top-1 accuracy of Inception-v3 trained on the transformed benign images and tested on the transformed adversarial images generated in a black-box setting.

C. Experiments of Variant Sparse Methods

To test the functions of additional steps added to the original sparse coding, we here compared the whole RegionSparse (Region) with those methods lacking different added steps: trained dictionary (noTR), image smoothing (noIS), patch normalization (noPN), and background perturbation suppression (noBS). Fig. 4 reports the results of this experiments. Apparently, RegionSparse outperforms the other methods. For L_∞ attacks with $L_\infty = 10$, it can correct up to 70% adversarial examples and surpass other methods with the improvement at most 20%. For L_2 attacks with $d = 0.1$, RegionSparse can successfully recognize more than 80% adversarial images. We also notice that smoothing steps, including image smoothing and background perturbation suppression, exert an important impact on defense capability with the improvement up to 10%.

TABLE II
RESULTS FOR ADVERSARIAL TRAINING EXPERIMENTS ($L_\infty = 2/14/22, L_2 = 0.01/0.14/0.2$).

Network	Defensive method	Clean	FGSM (L_∞)	BIM (L_∞)	CW-L2 (L_2)	MI-FGSM (L_2)
Inception-v3	Normal	100%/100%/100%	25%/22%/24%	15%/0%/0%	14%/13%/18%	24%/3%/1%
	FGSM Adversarial	83%/83%/83%	70%/52%/51%	76%/52%/42%	79%/74%/66%	74%/61%/51%
	RegionSparse	84%/84%/84%	75%/60%/55%	78%/70%/67%	83%/80%/76%	81%/75%/68%
	RegionSparse + FGSM Adversarial	91%/91%/91%	85%/73%/69%	85%/79%/76%	86%/85%/82%	85%/81%/79%

D. Gray Box: RegionSparse Coding at Test Time

Fig. 5 shows the top-1 accuracy of Inception-v3 model tested on the transformed images as a function of the attack strength. When there is no defense, the attacks successfully mislead the model, even though the model performs well on benign images. For L_∞ attacks, RegionSparse successfully corrects 55-85% images, and for L_2 attacks, it recovers 70-85% images. In addition, RegionSparse is robust as the attack strength increases. The maximum accuracy decrease of RegionSparse is lower than 30%, and that for CW-L_2 attack is even less than 10%. When confronting strong attacks, the performance of RegionSparse is apparently superior to any other methods, including the state-of-the-art method ComDefend. Other methods, by contrast, all suffer severe performance loss against strong attack. Another observation is that in the case of weak attacks, RegionSparse does not exhibit more excellent performance. A reasonable explanation is that the model is trained with high quality images, while RegionSparse projects images to a low-dimensional space by removing more features than other methods, which introduces artifacts and results in low quality images.

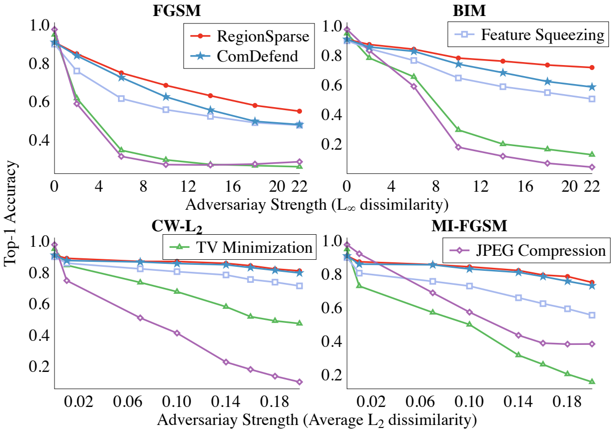

Fig. 7. Top-1 accuracy of retrained models tested on transformed images produced by four attacks, and the attacks have access to the resulting models.

E. Black Box: RegionSparse Coding at Training and Test Time

In this experiment, we randomly selected a subset from ImageNet dataset and retrained the models with transformed images. We collected the results of the retrained models on the adversarial images produced in gray-box setting. Fig. 6 shows that applying transformation methods at training time, indeed, conspicuously improves the effectiveness of defenses. RegionSparse benefits the most from this operation, and it gains accuracy improvements of 10% against attacks compared with gray-box setting experiments. Although its superiority on small-perturbation images is not distinct, the over 90% accuracy is ample for a majority of classification tasks. Overall, RegionSparse, with the most competitive precision and robustness, is still preferable to any other method.

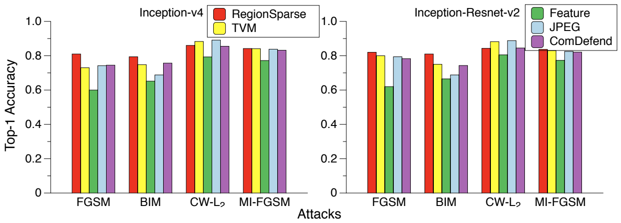

Fig. 8. Top-1 accuracy of Inception-v3 model tested on the transformed adversarial images generated from Inception-v4 and Inception-Resnet-v2 model with $L_\infty = 22$ for FGSM and BIM, $d = 0.1$ for CW-L_2 and MI-FGSM.

F. Extended Gray-Box: RegionSparse Coding at Training and Test Time

This experiment investigated the robustness of the retrained models when attackers had access to them. Fig. 7 shows the results of this experiment. In comparison with gray-box setting, defenses under extended gray-box setting can achieve improvements about 10% in classification accuracy. Among all methods, RegionSparse maintains a remarkable competitive edge with accuracy more than 60%.

G. Black Box: Transferred Attacks

We investigated the effectiveness of RegionSparse against attacks transferred from other models. The results presented in Fig. 8 show that each strategy is effective on the transferred attacks. RegionSparse, who can accurately classify more than 80% adversarial images, achieves the most competitive performance. An interesting observation is that TVM and JPEG Compression are more accurate than RegionSparse against CW-L_2 attack, and an explanation for it is that the two methods are more accurate in small-perturbation cases as they remove less features, as shown in Fig. 5.

H. Combined with Adversarial Training

A superiority of RegionSparse is that it can be easily combined with model-specific approaches such as adversarial training, remarkably improving its robustness. The results presented in Table II indicate that combining adversarial training and RegionSparse achieves satisfying accuracy. Generally, it can correct at least 70% adversarial images in any attack strength. This apparent improvement encourages defenders to apply more aggressive approximation (e.g. Larger sparseness) to handle adversarial examples.

I. Analysis of RegionSparse

The capability of RegionSparse comes from the favorable tradeoff of the compression levels between the object region and the background region. Sparse coding can aggressively suppress perturbations in the object region, and in the meanwhile bilateral filter mildly reduces perturbations in background and introduces less artifacts. This allows RegionSparse to achieve high accuracy by selectively handling perturbations spread over different regions of the image. An excellent property of RegionSparse is that it only needs a few images to capture benign features, greatly reducing its complexity. Moreover, RegionSparse is still effective in adaptive white-box setting where the attacker even has access to the defense deployed. It is difficult to attack RegionSparse by back propagating the model due to two reasons: i) RegionSparse is not differentiable because it divides the image into two regions and applies different processes on them. ii) Sparse coding compresses the image at patch granularity, increasing the difficulty to calculate the gradient of the model.

VI. Conclusion

This paper demonstrated that adversarial features are mainly lying in high-dimensional spaces by activation visualization. Based on that, an adversarial feature compression framework RegionSparse was proposed to counter adversarial attacks. The experimental results show that RegionSparse can achieve satisfying performance with accuracy up to 80-90%. Furthermore, integrating RegionSparse with adversarial training can enhance the robustness of models by immuning DNNs from artifacts introduced by sparse coding. Besides accuracy, flexibility is another advantage of RegionSparse, which allows it to be deployed into models without any modification. Our future work will focus on extending RegionSparse with model-specific approaches to effectively stabilize DNNs.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[3] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference (BMVC)*, 2015.

[4] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[5] P. Mirowski, M. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, A. Zisserman, R. Hadsell *et al.*, "Learning to navigate in cities without a map," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[6] C. Lyu, K. Huang, and H.-N. Liang, "A unified gradient regularization family for adversarial examples," in *IEEE International Conference on Data Mining (ICDM)*, 2015.

[7] N. Papernot and P. D. McDaniel, "Extending defensive distillation," in *arXiv preprint arXiv:1705.05264*, 2017.

[8] F. Tramer, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. Mcdaniel, "Ensemble adversarial training: Attacks and defenses," in *International Conference on Learning Representations (ICLR)*, 2018.

[9] C. Guo, M. Rana, M. Cisse, and L. V. Der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations (ICLR)*, 2018.

[10] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," in *arXiv preprint arXiv:1608.00853*, 2016.

[11] N. Das, M. Shanbhogue, S. Chen, F. Hohman, S. Li, L. Chen, M. E. Kounavis, and D. H. Chau, "Shield: Fast, practical defense and vaccination for deep learning using jpeg compression," in *ACM Knowledge Discovery and Data Mining (SIGKDD)*, 2018.

[12] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *ISOC Network and Distributed System Security Symposium, (NDSS)*, 2018.

[13] X. Jia, X. Wei, X. Cao, and H. Foroosh, "Comdefend: An efficient image compression model to defend adversarial examples." in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *arXiv preprint arXiv:1607.02533*, 2017.

[15] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy (SP)*, 2017.

[17] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *arXiv preprint arXiv:1506.06579*, 2015.

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision (ECCV)*, 2014.

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

[20] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[21] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, vol. 3, no. 3, p. e10, 2018.

[24] Z. Marzi, S. Gopalakrishnan, U. Madhow, and R. Pedarsani, "Sparsity-based defense against adversarial attacks on linear classifiers," in *IEEE International Symposium on Information Theory (ISIT)*, 2018.

[25] N. Akhtar and A. S. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[26] J. Mairal, F. R. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *International Conference on Machine Learning (ICML)*, 2009.

[27] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[28] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of Solid-state Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[29] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems and Computers*, 1993.

[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.