A NEURO-INSPIRED AUTOENCODING DEFENSE AGAINST ADVERSARIAL ATTACKS

Can Bakiskan *Metehan Cekic* *Ahmet Dundar Sezer* *Upamanyu Madhow*

Department of Electrical and Computer Engineering
University of California Santa Barbara, Santa Barbara, CA 93106
{canbakiskan,metehancekic,adsezer,madhow}@ece.ucsb.edu

ABSTRACT

Deep Neural Networks (DNNs) are vulnerable to adversarial attacks: carefully constructed perturbations to an image can seriously impair classification accuracy, while being imperceptible to humans. The most effective current defense is to train the network using adversarially perturbed examples. In this paper, we investigate a radically different, neuro-inspired defense mechanism, aiming to reject adversarial perturbations *before* they reach a classifier DNN, using an encoder with characteristics commonly observed in biological vision, followed by a decoder restoring image dimensions that can be cascaded with standard CNN architectures. Unlike adversarial training, all training is based on *clean* images. Our experiments on the CIFAR-10 and a subset of Imagenet datasets show performance competitive with state-of-the-art adversarial training, and point to the promise of bottom-up neuro-inspired techniques for the design of robust neural networks.

***Index Terms*—** Adversarial, Machine learning, Robust, Image classification, Defense

1. INTRODUCTION

The susceptibility of neural networks to small, carefully crafted input perturbations raises great concern regarding their robustness and security. Since this vulnerability of DNNs was pointed out [1, 2], there have been numerous studies on how to generate these perturbations (adversarial attacks) [3, 4] and how to defend against them [4, 5, 6, 7]. Existing defenses that attempt to employ systematic or provable techniques either do not scale to large networks, or have been defeated by appropriately modified attacks [5, 6, 8]. State of the art defenses [4, 9, 10] employ adversarial training (i.e., training the model with adversarially perturbed examples), but there is little insight into how DNNs designed in this end-to-end, "top down" fashion provide robust performance.

Approach: In this paper, we turn to neuro-inspiration for defending against adversarial attacks, inspired by the observation that humans barely register adversarial perturbations devised for machines. While neuro-inspiration could ultimately provide a general framework for designing DNNs which are robust to a variety of perturbations, in this paper, we take a first step by focusing on the well-known ℓ^∞ bounded attack, which captures the concept of "barely noticeable" perturbation. Our architecture, illustrated in Figure 1, does not require adversarial training: it consists of (a) a neuro-inspired encoder learnt in purely unsupervised fashion, (b) a decoder which produces an output of the same size as the original image, (c) a standard CNN for classification. The decoder and classifier are trained in standard supervised fashion using *clean* images passed through our encoder.

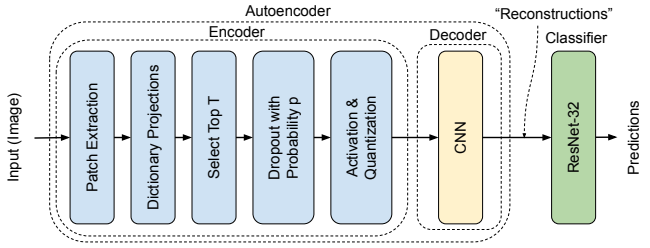

Fig. 1: Proposed autoencoding defense. Decoder restores input size but does not attempt to reconstruct the input in our nominal design (supervised decoder+classifier training).

The key features we incorporate into our encoder design are sparsity and overcompleteness, long conjectured to be characteristic of the visual system [11], lateral inhibition [12], synaptic noise [13], and drastic nonlinearity [14]. We use standard unsupervised dictionary learning [15] to learn a sparse, highly overcomplete (5-10X relative to ambient dimension) patch-level representations. However, we use the learnt dictionary in a non-standard manner in the encoder, not attempting patch-level reconstructions. Instead, we take the top T coefficients from each patch (lateral inhibition), randomly drop a fraction p of them (synaptic noise and lateral inhibition), and threshold and quantize them, retaining only their sign (drastic nonlinearity). We use overlapping patches, providing an additional degree of overcompleteness. The patch-level outputs, which have ternary quantized entries, are fed to a multi-layer CNN decoder whose output is the same size as the original RGB image input. This is then fed to a standard classifier DNN.

We report on experiments on the CIFAR-10 and a subset of the ImageNet dataset ("Imagenette"), demonstrating the promise of a "bottom-up" neuro-inspired approach, in contrast

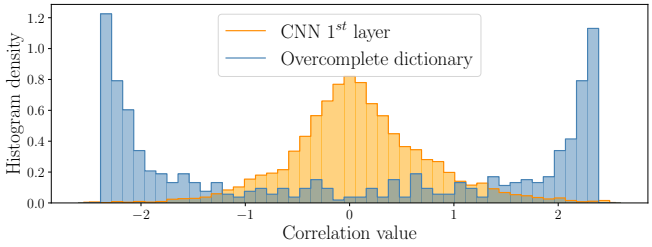

Fig. 2: Histogram of correlations for a typical patch with atoms of an overcomplete dictionary vs. that of activations through layer 1 filters of a standard classifier CNN.

to the top-down approaches that currently dominate adversarial machine learning. For state of the art PGD attacks tailored to our architecture, our attacked accuracy is slightly worse than that of adversarial training [4, 9] for CIFAR-10, while being on par or slightly better than these methods for ImageNette, showing that our approach scales to larger image sizes.

We invest significant effort into attacking our own defense: following the guidelines in [16], our strongest attack is tailored specifically to account for the structure of our defense while avoiding the gradient obfuscation problem exposed in [17]. The software for our defense, including the attack library we have created, is available at github.com/canbakiskan/neuro-inspired-defense.

2. ADVERSARIAL ATTACKS AND DEFENSES

Attacks: These can be broadly grouped into two categories: whitebox attacks, in which the attacker has access to both the structure and the parameters of the neural network; and blackbox attacks, which have access only to the network outputs. Given a classifier $f : \mathbf{x} \in \mathbb{R}^N \rightarrow \mathbf{y} \in \mathbb{R}^C$, the goal of an adversary is to find a perturbation $\mathbf{e}$ that maximizes the given loss function $\mathcal{L}$ for classification under some constraints. Typically, adversarial attacks are constrained in ℓ^p norm, with $p = \infty$ receiving the greatest attention because it can be tuned to be imperceptible to humans [3, 18]. Among the many attack methods, Projected Gradient Descent (PGD) appears to be the most effective first order ℓ^∞ bounded attack, and is therefore generally used to evaluate defense methods. PGD computes the perturbation iteratively as follows:

$$\mathbf{e}_{i+1} = \text{clip}_\epsilon\big[\mathbf{e}_i + \delta \cdot \text{sign}(\nabla_{\mathbf{e}}\mathcal{L}(\mathbf{f}(\mathbf{x}+\mathbf{e}_i), \mathbf{y}))\big] \quad (1)$$

where $\mathbf{e}_i$ corresponds to the value of the perturbation at iteration i with $\mathbf{e}_0 = \mathbf{0}$ or $\mathbf{e}_0$ with each element drawn from uniform distribution $\mathcal{U}(-\epsilon, \epsilon)$, ϵ is the overall ℓ^∞ attack budget, and δ is the step size for each iteration. Expectation Over Transformation (EOT) is suggested in [19] to make attacks robust against transformations, and [16] suggests using this method to evaluate defenses utilizing stochasticity. With EOT, PGD becomes:

$$\mathbf{e}_{i+1} = \text{clip}_\epsilon\big[\mathbf{e}_i + \delta \cdot \text{sign}(\sum_{r=0}^{N_E-1} \nabla_{\mathbf{e}}\mathcal{L}_r(\mathbf{f}(\mathbf{x}+\mathbf{e}_i), \mathbf{y}))\big] \quad (2)$$

where $\mathbf{e}_0 = \mathbf{0}$ and N_E corresponds to the number of runs used to average the gradients.

Defenses: State of the art adversarial training employs perturbations computed using variants of the original FGSM method [3] of gradient ascent on a cost function, including PGD [4] and recent enhancements such as the faster single-step R+FGSM scheme in [20], the use of a modified cost function aiming to trade off clean and adversarial accuracy (called TRADES) in [9], or the use of more unlabeled data in [10].

3. AUTOENCODING DEFENSE

The rationale behind our encoder design is as follows: An overcomplete dictionary for sparse coding results in large activations for a small fraction of the atoms, in contrast with filters learnt in the first layer of a traditional convolutional neural network where activations are clustered around zero; see Figure 2. We can therefore drop most of the activations, reducing the effective subspace available to the attacker. An attacker can still perturb the subset of top T coefficients in each patch. Randomly dropping a large fraction p of these coefficients allows the decoder and classifier to learn to be resilient to randomness in the sparse code, and to an attacker knocking a coefficient out of the top T. The thresholds for ternary quantization of the selected coefficients are selected to provably guarantee that the attacker cannot flip the sign of any nonzero entry in the sparse code. The hard thresholding ensures that the perturbation cannot add to a coefficient which would have been selected for a clean image. Rather, the attacker must invest the effort in pushing a smaller coefficient into the top T, and gamble on it being randomly selected.

3.1. Patch-Level Overcomplete Dictionary

We consider images of size $N \times N$ with 3 RGB channels, processed using $n \times n$ patches with stride S, so that we process $M = m \times m$ patches, where $m = \lfloor (N-n)/S \rfloor + 1$. Learning at the patch level allows for the extraction of sparse local features, effectively allowing reduction of the dimension of the space over which the adversary can operate for each patch.

We use a standard algorithm [15] (implemented in Python library `scikit-learn`), which is a variant of K-SVD [21]. Given a set of clean training images $\mathcal{X} = \{\mathbf{X}^{(k)}\}_{k=1}^K$, an overcomplete dictionary $\mathbf{D}$ with L atoms can be obtained by solving the following optimization problem [15]

$$\min_{\mathbf{D}\in\mathcal{C}, \{\boldsymbol{\alpha}^{(k)}\}_{k=1}^K} \sum_{k=1}^K \sum_{i,j} \Big(\frac{1}{2} \Big\| \mathbf{R}_{ij}\mathbf{X}^{(k)} - \mathbf{D}\boldsymbol{\alpha}_{ij}^{(k)} \Big\|_2^2 + \lambda \Big\| \boldsymbol{\alpha}_{ij}^{(k)} \Big\|_1 \Big) \quad (3)$$

where $\mathcal{C} \triangleq \big\{\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_L] \in \mathbb{R}^{\bar{n}\times L} \mid \|\mathbf{d}_l\|_2 = 1, \forall l \in \{1, \ldots, L\}\big\}$, λ is a regularization parameter, $\boldsymbol{\alpha}^{(k)}$ is an $m \times m \times L$ tensor containing the coefficients of the sparse decomposition, and $\mathbf{R}_{ij} \in \mathbb{R}^{\bar{n}\times\bar{N}}$ with $\bar{n} \triangleq 3n^2$ and $\bar{N} \triangleq 3N^2$

extracts the (ij)-th patch from image $\mathbf{X}^{(k)}$. The optimization problem in (3) is not convex, but its convexity with respect to each of the two variables $\mathbf{D}$ and $\{\boldsymbol{\alpha}^{(k)}\}_{k=1}^{K}$ allows for efficient alternating minimization [15, 21].

3.2. Sparse Randomized Encoder

Based on the overcomplete dictionary obtained from (3), we encode the image patch by patch. For given image $\mathbf{X}$, patch $\mathbf{x}_{ij} \in \mathbb{R}^{\bar{n}}$ is extracted based on the (ij)-th block of $\mathbf{X}$; that is, $\mathbf{x}_{ij} = \mathbf{R}_{ij}\mathbf{X}$, and then projected onto dictionary $\mathbf{D}$ in order to obtain projection vector $\bar{\mathbf{x}}_{ij}$, where $\bar{\mathbf{x}}_{ij} = \mathbf{D}^T\mathbf{x}_{ij}$. Since the dictionary is highly overcomplete, a substantial fraction of coefficients typically take large values, and a sparse reconstruction of the patch can be constructed from a small subset of these. However, our purpose is robust image-level inference rather than patch-level reconstruction, hence we use the dictionary to obtain a discrete sparse code for each patch using random "population coding," as follows.

1) Top T selection: We keep only T elements of the projection vector with largest absolute values and zero out the remaining elements. The surviving coefficients are denoted by $\check{\mathbf{x}}_{ij}$.

2) Dropout: Each of the top T coefficients is dropped with probability p, leaving surviving outputs

$$\tilde{\mathbf{x}}_{ij}(l) = \begin{cases} 0, & \text{with probability } p \\ \check{\mathbf{x}}_{ij}(l), & \text{with probability } 1-p \end{cases}, \qquad (4)$$

for all $l \in \{1, \ldots, L\}$.

Note that we use dropout for both train and test time, as opposed to only during training in its standard usage.

3) Activation/Quantization: Finally, we obtain sparse codes with discrete values by applying binary quantization with a dead zone designed to reject perturbations.

$$\hat{\mathbf{x}}_{ij}(l) = \begin{cases} \operatorname{sign}\left(\tilde{\mathbf{x}}_{ij}(l)\right) \|\mathbf{d}_l\|_1, & \text{if } \frac{|\tilde{\mathbf{x}}_{ij}(l)|}{\epsilon \|\mathbf{d}_l\|_1} \geq \beta \\ 0, & \text{otherwise} \end{cases}, \qquad (5)$$

for all $l \in \{1, \ldots, L\}$, where $\beta > 1$ is a hyperparameter.

Rationale: By Hölder's inequality, an attacker with ℓ^∞ budget ϵ can perturb the kth basis coefficient by at most $\epsilon \|\mathbf{d}_k\|_1$. By choosing $\beta > 1$, we guarantee that an attacker can never change the sign of a nonzero element of the sparse code. Thus, the attacker can only demote a nonzero element to zero, or promote a zero element to a nonzero value. As discussed, a large dropout probability alleviates the impact of both demotions and promotions.

Another consequence of choosing $\beta > 1$ is that weak patches whose top T coefficients are not large enough compared to the maximum perturbation $\epsilon \|\mathbf{d}_k\|_1$ get killed, thereby denying the adversary the opportunity to easily perturb the patch-level sparse code. The scaling of the surviving ± 1 outputs by $\|\mathbf{d}_l\|_1$ allows basis coefficients surviving a larger ℓ^1 norm based threshold to contribute more towards the decoder input, but could be omitted, since the decoder can learn the appropriate weights.

		PGD with EOT		
	Clean	Whitebox	PW-T	Blackbox
Our defense	80.06	61.28	**39.53**	57.76

Table 1: Accuracies for our defense method under different attacks (CIFAR-10, $\epsilon = 8/255$)

Following patch-level processing with stride S, the encoder outputs an image level sparse code which is an $m \times m \times L$ tensor.

3.3. CNN-based Decoder

We employ a CNN-based decoder architecture employing three transposed convolutional layers, each followed by ReLU activation function, clipped at the end to produce output with dimension $N \times N \times 3$ equal to that of the original RGB image.

3.4. Ensemble Processing

In order to utilize the full potential of the randomization employed in the encoder, we allow for ensemble processing in which an input image is processed using E random realizations of our encoder during inference, with classifier softmax outputs averaged across the realizations.

4. EXPERIMENTS, RESULTS AND DISCUSSION

Our main focus is on evaluating our defense on the CIFAR-10 dataset ($N = 32$), for which there are well-established benchmarks in adversarial ML. In order to verify that our approach scales to larger images, we also consider the Imagenette dataset: 9469 train and 3925 validation RGB images, cropped to size 160×160 ($N = 160$). Both datasets contain images from 10 classes. For CIFAR-10, we use 4×4 patches ($n = 4$) and an overcomplete dictionary with $L = 500$ atoms. The stride $S = 2$, so the encoder output is a $15 \times 15 \times 500$ tensor ($m = 15$, $L = 500$). The regularization parameter in (3) is set to $\lambda = 1$ and the number of iterations is chosen as 1000 to ensure convergence. The hyperparameters for Imagenette are: 8×8 ($n = 8$) patches and an overcomplete dictionary with $L = 1000$ atoms, stride $S = 4$ which gives encoder outputs of size $38 \times 38 \times 1000$ ($m = 38$, $L = 1000$). The number L of dictionary atoms is 10 times the ambient dimension for CIFAR-10, and 5 times the ambient dimension for ImageNette. The number of iterations in dictionary learning is set to 10000, and in order to promote sparsity, the regularization parameter λ is set to 0.5, in the upper range of values resulting in convergence.

We set $T = 50$, $p = 0.95$, $E = 10$ for our nominal defense based on ablation studies (omitted due to space restrictions), with hyperparameter $\beta = 3$ for the threshold in (5). We train the CNN-based decoder in supervised fashion in tandem with the classifier, using the standard cross-entropy loss. For unsupervised (US) decoder training, we use ℓ^2 distance-squared regression over 50 epochs. In all cases, we use a cyclic learning

rate scheduler [22] with a minimum and maximum learning rate of $\eta_{min} = 0$ and $\eta_{max} = 0.05$, respectively. In order to provide a consistent evaluation, we employ the ResNet-32 classifier used in [4] for CIFAR-10, and use EfficientNet-B0 [23] for Imagenette. The number of epochs for supervised training is 70 for CIFAR-10 and 100 for Imagenette.

Attacks: We report on three attacks on our nominal defense: (1) *whitebox*, where every differentiable operation is differentiated. For the non-differentiable activation/quantization, we take a smooth backward pass approximation. (2) *Pseudo-Whitebox - Transfer (PW-T)*, where we generate whitebox attacks for an *unsupervised-trained decoder* with the same encoder, with standard supervised training of the classifier. This attack is adapted specifically for our defense, and does not apply to the benchmark defenses that we compare against. *Blackbox*, where the adversarial attack is generated based on a standard adversarially trained surrogate classifier. For attacks, we consider PGD and PGD with EOT if it is applicable. Different from the existing EOT implementation, we use $\delta \cdot \text{sign}\left(\mathbf{E}_r\left[\nabla_x/||\nabla_x||_2\right]\right)$ in each step to compute the expectation, which we find yields stronger attacks. Unless otherwise stated, we use the following parameters for ℓ^∞ bounded PGD with EOT for CIFAR-10 trained models: an attack budget of $\epsilon = 8/255$, a step size of $\delta = 1/255$, a number of $N_S = 20$ steps, a number of $N_R = 1$ restarts, and a number of $N_E = 40$ realizations for EOT. The same default attack parameters are used for attacking models trained on Imagenette, but given the lack of standard benchmarks, we test several attack budgets $\epsilon \in \{2/255, 4/255, 8/255\}$.

*Benchmarks:*Our benchmarks are the PGD adversarially trained (AT) [4], R+FGSM adversarially trained [20], and TRADES [9] defenses for the same classifier architecture. We reimplement these, to enable stress-testing these defenses with attacks of varying computational complexity. We train these models for 100 epochs with the same cyclic learning rate that we use for our models, and verify, for ResNet-32 classifier for CIFAR-10 and EfficientNet-B0 for Imagenette, that we can reproduce results obtained using the original code. For both PGD AT and TRADES, training hyperparameters are $\epsilon = 8/255$, $\delta = 1/255$, $N_S = 10$, $N_R = 1$ additionally, for TRADES $\lambda_{\text{TRADES}} = 1/6$. For RFGSM AT, they are $\epsilon = 8/255$, $\alpha = 10/255$. We also report on naturally trained (NT) networks (i.e., no defense).

Note that the classifier CNN used in our paper is "simpler" ResNet-32 rather than the wide ResNet-32, both of which are utilized in [4] and other studies in the literature. The choice of the smaller ResNet-32 network makes evaluation of attacks computationally more feasible.

Robustness against Defense-Adapted Attacks: We first investigate the performance of our defense under the different attack types specified earlier. Table 1 provides clean and adversarial accuracies for the different attack types. We note that the worst-case attack for it is *not* a white box attack, rather, it is the pseudo-whitebox transfer (PW-T) attack. While this result is surprising at first, it is intuitively pleasing. An attack succeeds only to the extent to which it can change the identities of the top T coefficients in the encoder. Since the latter is designed to preserve information about the original image, providing an unsupervised decoder might provide better guidance to the attacker by giving it a reproduction of the original image to work with.

	Clean	Adv. (Worst case)
NT	**93.10**	0.00
PGD AT [4]	79.41	42.05
RFGSM AT [20]	80.86	42.42
TRADES [9]	75.17	**45.79**
Our defense	80.06	39.53

Table 2: Comparison of our defense with other defense techniques (CIFAR-10, $\epsilon = 8/255$). Attack details are: PGD with $N_S = 100$, $N_R = 50$ for the first 4 rows and PGD EOT with $N_S = 20$, $N_R = 1$, $N_E = 40$ for the last row.

Comparison with benchmarks: Table 2 lists *worst-case* accuracies for each defense, where we vary the computational burden of attack on the benchmarks up to a point that is comparable to the default settings for our own EOT/PGD attack. NT denotes natural training (no defense). The worst-case adversarial accuracy for our defense is 39.53%, a little worse than the worst-case accuracies of 42-46% for the benchmark defenses. On the other hand, the worst-case accuracy of our defense (again achieved by the PW-T attack) is slightly better than for the benchmark defenses for Imagenette, as reported in Table 3. For NT, PGD AT, and TRADES, we use PGD attack with default parameters.

5. CONCLUSIONS

While our results demonstrate the potential of neuro-inspiration and bottom-up design of robust DNNs, there is significant scope for further improvement. We attenuate perturbations in a single, rather drastic, encoding step, but spreading the burden across more layers may help with both clean and attacked accuracy. Our separation of decoder and classifier enables reuse of standard classifier architectures, but there might be better options. Finally, the efficacy of the transfer attack designed specifically for our defense highlights the need for further research on adaptive attacks for novel defenses.

	Clean	Adversarial ($\epsilon = x/255$)		
		$x = 2$	$x = 4$	$x = 8$
NT	**89.35**	11.44	0.28	0.00
PGD AT	80.97	75.31	68.81	53.32
TRADES	80.08	75.67	70.75	59.46
Our defense	79.36	**76.03**	**72.81**	**65.45**

Table 3: Accuracies for Imagenette dataset

6. REFERENCES

[1] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.

[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[4] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[5] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten, "Countering adversarial images using input transformations," *arXiv:1711.00117*, 2017.

[6] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu, "ME-Net: Towards effective adversarial robustness with matrix estimation," *arXiv:1905.11971*, 2019.

[7] Can Bakiskan, Soorya Gopalakrishnan, Metehan Cekic, Upamanyu Madhow, and Ramtin Pedarsani, "Polarizing front ends for robust CNNs," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4257–4261.

[8] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[9] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan, "Theoretically principled trade-off between robustness and accuracy," *arXiv:1901.08573*, 2019.

[10] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang, "Unlabeled data improves adversarial robustness," in *Advances in Neural Information Processing Systems*, 2019, pp. 11192–11203.

[11] Bruno A Olshausen and David J Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision research*, vol. 37, no. 23, 1997.

[12] Colin Blakemore, Roger HS Carpenter, and Mark A Georgeson, "Lateral inhibition between orientation detectors in the human visual system," *Nature*, vol. 228, no. 5266, pp. 37–39, 1970.

[13] Steven A Prescott and Yves De Koninck, "Gain control of firing rate by shunting inhibition: Roles of synaptic noise and dendritic saturation," *Proceedings of the National Academy of Sciences*, vol. 100, no. 4, 2003.

[14] Ryan Prenger, Michael C-K Wu, Stephen V David, and Jack L Gallant, "Nonlinear V1 responses to natural scenes revealed by neural network analysis," *Neural Networks*, vol. 17, no. 5-6, pp. 663–679, 2004.

[15] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 689–696.

[16] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry, "On adaptive attacks to adversarial example defenses," *arXiv:2002.08347*, 2020.

[17] Anish Athalye, Nicholas Carlini, and David Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, "Adversarial examples in the physical world," in *ICLR Workshop*, 2017.

[19] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok, "Synthesizing robust adversarial examples," *arXiv:1707.07397*, 2017.

[20] Eric Wong, Leslie Rice, and J Zico Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv:2001.03994*, 2020.

[21] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[22] Leslie N Smith, "Cyclical learning rates for training neural networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017.

[23] Mingxing Tan and Quoc V Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv:1905.11946*, 2019.