# Defense Against Adversarial Images using Web-Scale Nearest-Neighbor Search: Supplementary Material

Abhimanyu Dubey*[1, 2], Laurens van der Maaten[2], Zeki Yalniz[2], Yixuan Li[2], and Dhruv Mahajan[2]

[1]Massachusetts Institute of Technology
[2]Facebook AI

## Appendix A: Adversarial Attack Methods

**Fast Gradient Sign Method (FGSM)** [1] is one of the earliest attack techniques that has been demonstrated to successfully produce adversarial samples. The FGSM attack produces adversarial samples using the update rule:

$$\mathbf{x}^*_{\textsf{FGSM}} = \mathbf{x} + \varepsilon \cdot \text{sign}(\nabla_{\mathbf{x}} L(\mathbf{x}, y)),$$

where $\mathbf{x}$ is the unperturbed input. When the model is available to the attacker (*white-box* setting), the attack can be run using the true gradient $\nabla_{\mathbf{x}} L(h(\mathbf{x}), y)$, however, in *gray-box* and *black-box* settings, the attacker may habe access to a surrogate gradient $\nabla_{\mathbf{x}} L(h'(\mathbf{x}), y)$, which in practice, has been shown to be effective as well. A stronger version of this attack is the **Iterative Fast Gradient Sign Method (I-FGSM)** [6], where the adversarial input is generated by iteratively applying the FGS update over $m = \{1, ..., M\}$ steps, following:

$$\mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} + \varepsilon \cdot \text{sign}(\nabla_{\mathbf{x}^{(m-1)}} L(\mathbf{x}^{(m-1)}, y)), \\ \text{where } \mathbf{x}^*_{\textsf{IFGSM}} = \mathbf{x}^{(M)}, \text{ and } \mathbf{x}^{(0)} = \mathbf{x}$$

A general version of the I-FGSM attack is the **Projected Gradient Descent (PGD)** [7] attack, which clips the gradients to project them back to the feasible image domain, and also includes random restarts in the optimization process. We use this attack over the I-FGSM attack due to its stronger nature. The FGSM, I-FGSM, and PGD attacks approximately minimize the Chebyshev distance between the input $\mathbf{x}$ and the generated adversarial sample $\mathbf{x}^*$. **Carlini-Wagner's $\ell_p$ (CW-Lp)** attack attempts to find a solution to an unconstrained optimization problem that jointly penalizes a differentiable surrogate for the model accuracy along with a distance measure for regularization, such as the $\ell_2$ or $\ell_\infty$ distance.

$$\mathbf{x}^*_{\textsf{CW-Lp}} = \min_{\mathbf{x}'} \big[\|\mathbf{x} - \mathbf{x}'\|_p^2 + \lambda_f \max(-\kappa, Z(\mathbf{x}')_{h(\mathbf{x})} \\ - \max\{Z(\mathbf{x}')_k : k \neq h(\mathbf{x})\})\big]$$

Herein, $\kappa$ denotes a margin parameter, and the parameter $\lambda_f$ relatively weighs the losses from the distance penalty and accuracy surrogate (hinge loss of predicting an incorrect class). The most common values for $p$ are $p = 2$ and $p = \infty$; we use the implementation of [2] to implement FGSM, IFGSM, PGD, and CW-L2. For all the above attacks, we enforce that the image remains within $[0, 1]^d$ by clipping values.

## Appendix B: Feature Construction

To evaluate the trade-off between robustness and accuracy of neural network features at different depths (layers), we use features extracted from different layers of ResNet-50 models [3] as the basis for retrieving nearest neighbors. Since layers are very high-dimensional, we reduce the final dimensionality of the feature vectors to 256 by performing a spatial average pooling followed by PCA (see Table 1 for a complete description). We found that spatial average pooling step helps in increasing accuracy as well as the computational efficiency of PCA.

| Feature Layer | Uncompressed Size | Pooled Size |
|---|---|---|
| conv_2_3 | $256 \times 56 \times 56$ | $256 \times 7 \times 7$ |
| conv_3_4 | $512 \times 28 \times 28$ | $512 \times 7 \times 7$ |
| conv_4_6 | $1024 \times 14 \times 14$ | $1024 \times 4 \times 4$ |
| conv_5_1 | $2048 \times 7 \times 7$ | $2048 \times 1 \times 1$ |

Table 1. Feature vector details. All features are finally compressed to a dimensionality of 256 by a PCA done over 3M samples.

The features are extracted using the PyTorch [8] framework; we use the SciPy [5] implemention of online PCA for dimensionality reduction. The PCA was computed on

*This work was done while Abhimanyu Dubey was at Facebook AI.

3 million randomly selected samplesfor the *IG* and *YFCC* datasets, and on the complete training set for ImageNet.

For nearest-neighbor matching, we construct a pipeline using the GPU implementation of the FAISS [4] library; we refer the readers to the original paper for more details about billion-scale fast similarity search.

## Appendix C: Hard versus Soft Combination of Predictions

As discussed in the main paper, we evaluate both hard and soft prediction combinations, as described below:

**Soft Combination (SC)**: We return the *weighted average* of the softmax probability vector of all the nearest neighbors. The predicted class is then the $\arg\max$ of this average vector. This corresponds to a "soft" combination of the model predictions over the data manifold.

**Hard Combination (HC):** In this case, each nearest neighbor votes for its "hard" predicted class with some weight. The final prediction is then taken as the most commonly predicted class.

In Table 2, we present the results of both these approaches with uniform weighing (UW) and confidence-based weighing (CBW). These results were obtained using the same experimental setup as described in the main paper.

## Appendix D: Results with FGSM and CWL-2

Below, we present results that measure the effectiveness of our defense strategy against the FGSM and CWL-2 attacks. All experiments follow the same experimental setup as described in the main paper (but use a different attack method).

**Effect of Number of Neighbors, $K$.** Figures 1 and 2 describe the effect of varying $K$ under FGSM and CWL-2 attacks. We observe similar trends as for the PGD attacks.

**Effect of Image Database Size.** Figure 3 and 4 describe the effect of varying the index size under FGSM and CWL-2 attacks. We observe similar trends as for the PGD attacks.

**Effect of Feature Space.** Figure 5 and 6 present classification accuracies obtained using CBW-D defenses based on four different feature representations of the images in the IG-1B-Targeted database, for the FGSM and CWL-2 attacks respectively. We observe a similar trend as in the case of PGD attacks.

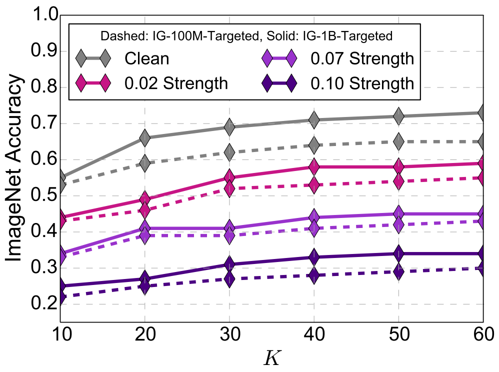

Figure 1. Classification accuracy of ResNet-50 using our CBW-D defense on FGSM adversarial ImageNet images, as a function of the normalized $\ell_2$ norm of the adversarial perturbation. Defenses are implemented via nearest-neighbor search using `conv_5_1` features on the IG-1B-Targeted (solid lines) and IG-100M-Targeted (dashed lines). Results are for the black-box setting.

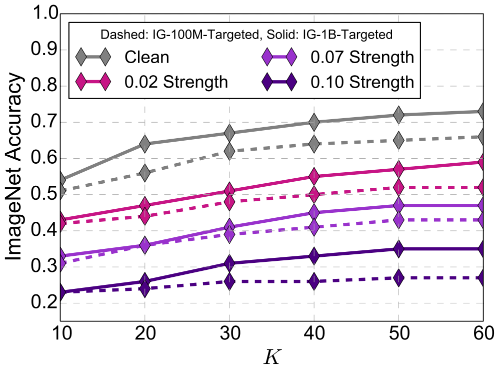

Figure 2. Classification accuracy of ResNet-50 using our CBW-D defense on CWL-2 adversarial ImageNet images, as a function of the normalized $\ell_2$ norm of the adversarial perturbation. Defenses are implemented via nearest-neighbor search using `conv_5_1` features on the IG-1B-Targeted (solid lines) and IG-100M-Targeted (dashed lines). Results are for the black-box setting.

## References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[2] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017. 1

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017. 1

[5] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed ¡today¿]. 1

[6] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 1

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks.

| | Clean | | | | | | Gray box | | | | | | Black box | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Soft Combination | | | Hard Combination | | | Soft Combination | | | Hard Combination | | | Soft Combination | | | Hard Combination | | |
| **Image database** | UW | CBW-E | CBW-D | UW | CBW-E | CBW-D | UW | CBW-E | CBW-D | UW | CBW-E | CBW-D | UW | CBW-E | CBW-D | UW | CBW-E | CBW-D |
| IG-50B-All (`conv_5_1-RMAC`) | 0.632 | 0.644 | **0.676** | 0.637 | 0.642 | 0.649 | 0.395 | 0.411 | **0.427** | 0.402 | 0.403 | 0.414 | 0.448 | 0.459 | **0.491** | 0.457 | 0.462 | 0.473 |
| IG-1B-Targeted (`conv_5_1`) | 0.659 | 0.664 | **0.681** | 0.668 | 0.671 | 0.673 | 0.415 | 0.429 | **0.462** | 0.418 | 0.423 | 0.437 | 0.568 | 0.574 | **0.587** | 0.554 | 0.561 | 0.571 |
| IN-1.3M (`conv_5_1`) | 0.472 | 0.469 | 0.471 | **0.475** | 0.472 | 0.473 | 0.285 | 0.286 | 0.286 | 0.291 | 0.289 | **0.293** | 0.311 | 0.312 | 0.312 | **0.316** | 0.309 | 0.314 |

Table 2. ImageNet classification accuracies of ResNet-50 on PGD-generated images with a normalized $\ell_2$ distance of $0.06$, using our nearest-neighbor defenses with three different image databases on both soft and hard combination techniques, with three different weighing strategies (UW, CBW-E and CBW-D), and $K = 50$. Accuracies on clean images are included for reference.

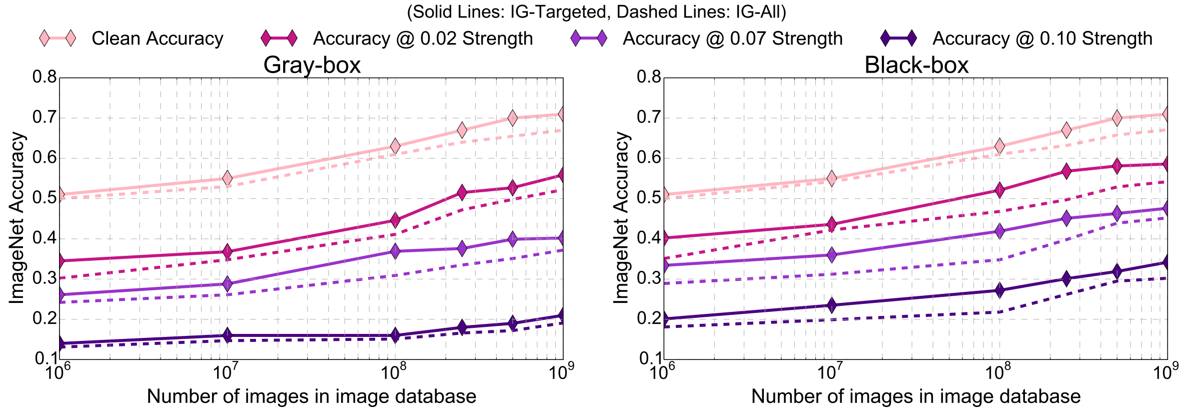

Figure 3. Classification accuracy of ResNet-50 using the CBW-D defense on FGSM adversarial ImageNet images, using the IG-$N$-Targeted database (solid lines) and IG-$N$-All database (dashed lines) with different values of $N$. Results are presented in the gray-box (left) and black-box (right) settings.

*arXiv preprint arXiv:1706.06083*, 2017. 1

[8] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 1

[9] Y. Wang, S. Jha, and K. Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. *arXiv preprint arXiv:1706.03922*, 2017.

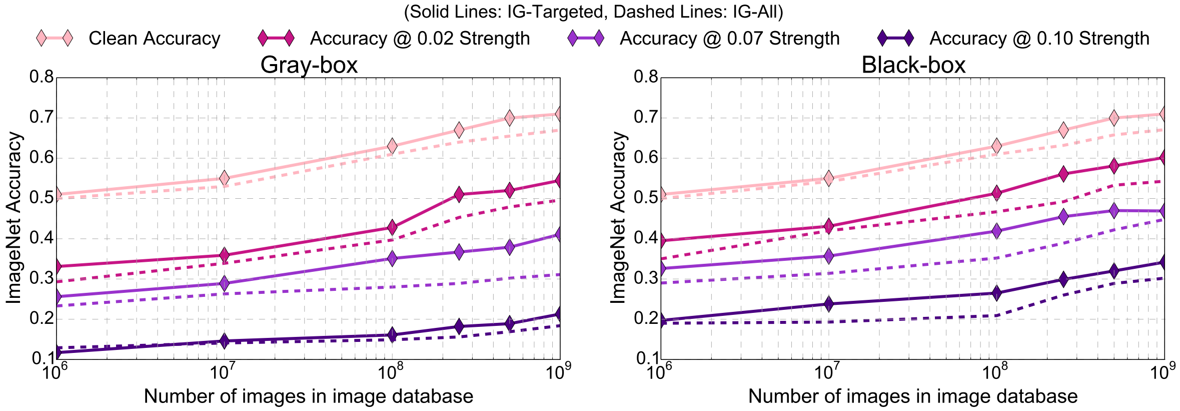

Figure 4. Classification accuracy of ResNet-50 using the CBW-D defense on CWL-2 adversarial ImageNet images, using the IG-$N$-Targeted database (solid lines) and IG-$N$-All database (dashed lines) with different values of $N$. Results are presented in the gray-box (left) and black-box (right) settings.

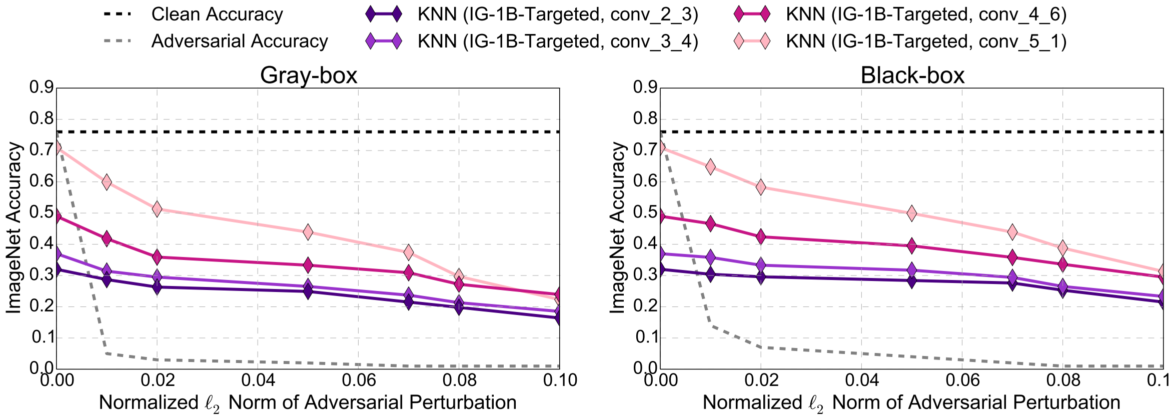

Figure 5. Classification accuracy of ResNet-50 using the CBW-D defense on FGSM adversarial ImageNet images, as a function of the normalized $\ell_2$ norm of the adversarial perturbation. Defenses use four different feature representations of the images in the IG-1B-Targeted image database. Results are presented for the gray-box (left) and black-box (right) settings.

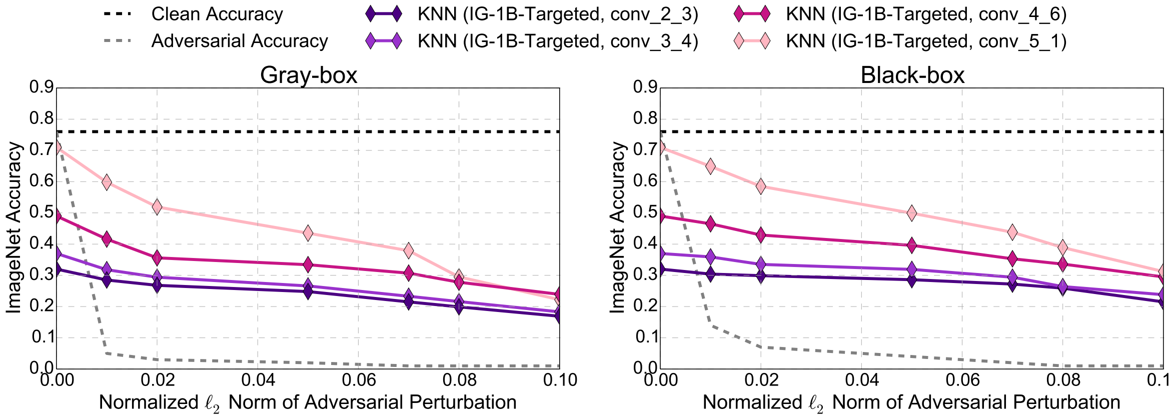

Figure 6. Classification accuracy of ResNet-50 using the CBW-D defense on CWL-2 adversarial ImageNet images, as a function of the normalized $\ell_2$ norm of the adversarial perturbation. Defenses use four different feature representations of the images in the IG-1B-Targeted image database. Results are presented for the gray-box (left) and black-box (right) settings.