Exploring "Feature Denoising for Improving Adversarial Robustness" - Network Depth and Image Similarity

Anthony Pineci
CMS, Caltech

Ayooluwa Odemuyiwa
CMS, Caltech

Randall Pulido
CMS, Caltech

June 5, 2021

Deep neural networks have been found to be vulnerable to adversarial attacks. Denoising models have been proposed[21] as methods of mitigating adversarial perturbations to intermediate feature maps. We explore how adversarial perturbations propagate through robust denoising systems by tracking the dissimilarity between the adversarial feature maps and those of the original image. We also compare the dissimilarity between the feature maps of a randomly perturbed image and its unmodified counterpart. We find that adversarial perturbations not only increase as an adversarial example propagates through a network, but also induces structural changes to the expected intermediate feature maps. This implies that the proposed denoising techniques do not adequately account for the induced changes in the feature maps of an adversarial example's forward pass.

1 INTRODUCTION

1.1 ADVERSARIAL MACHINE LEARNING

Adversarial machine learning is a technique in which a sample input is contrived in an attempt to deceive a machine learning model by forcing a misclassification. In the case of image classification models, a small amount of carefully selected noise may be applied to an image

in order to perturb it by either using the inner structure and parameter weights of the targeted model or by treating the model as a black-box. This perturbation can be imperceptible to the human eye, rendering the noisy image seemingly indistinguishable from the original. The exploited system is then fed the noisy image which it subsequently misclassifies.

1.2 Implications of Vulnerability

Adversarial attacks pose a significant risk to machine learning systems by leaving them vulnerable to manipulation by outside forces. Attacks could occur in benign scenarios, such as bypassing email spam filters, to dangerous situations such as fooling a weapons detection system in an airport or high security building. This threat is more alarming as black-box attacks have been shown to be possible through reverse engineering techniques. They are also generalizable in their effectiveness in exploiting a variety of machine learning methods.[8] Such vulnerabilities are growing,[15] and not only for image classification models. Widespread learning systems, such as systems that underlie automatic speech recognition, have been found to have vulnerabilities which allow extreme attack effectiveness.[4] Furthermore, security measures against adversarial attacks are limited and ineffective against increasingly efficient attacks.[15] Supposedly robust learning systems which are comprehensively fortified against attacks become quickly exploited by more vigorous adversarial machine learning models and attack methods. The rapid growth in popularity of machine learning and its ubiquity in the data age underscores the importance of the threat of adversarial attacks.

The existence of adversarial examples may also upend the conception that network models mimic the activity of the human brain. A neural net is able to perform resoundingly well on image classification tasks, yet will also fail to correctly predict on adversarial examples that humans can easily classify. This indicates that although their structure may be similar, the way that information is processed within an artificial neural network may differ drastically from the way information is processed in a biological one. A thorough understanding of the mechanisms underlying adversarial attacks may allow the design of new network architectures which are not only more robust, but also more similar to the biological recognition process which could lead to feature maps with more semantic meaning.

2 Background and Previous Work

2.1 Noise Propagation

It is necessary to understand the mechanisms within a network in order to understand its vulnerabilities and prevent it from being attacked. The way in which adversarial noise propagates through a model may elucidate its vulnerabilities. Network depth is a crucially important factor in image classification systems[16]. Intuitively, in image classification networks, the noise of the features introduced to an image should gradually increase as the image propagates through the network. It is hypothesized[21] that this is what ultimately leads to an adversarial example, generated by introducing a small amount of noise to an image to force it to be misclassified. By treating adversarial examples as noisy perturbations of unaltered exam-

ples, network architectures can be designed with noise robustness with the aim of improving adversarial robustness.

2.2 Networks and Denoising

In their 2018 paper,[21] Xie et al. attempt to use feature denoising to improve adversarial robustness under the assumption that adversarial examples are explained as noisy perturbations. The authors insert denoising blocks at intermediate layers of convolutional networks. The models are trained end-to-end with these denoising blocks using an adversarial training[5] method where the model is given adversarial examples as training inputs. This training scheme allows the network to reduce feature map noise due to perturbations to the original input, including adversarial perturbations. The best preforming denoising block uses non-local means[3] which takes a weighted mean of features across all spatial locations in a feature map:

$$y_i = \frac{1}{\mathscr{C}(x)} \sum_{\forall j \in \mathscr{L}} f(x_i, x_j) \cdot x_j$$

where x_i is the value of the feature map at location i, $\mathscr{L}$ is the set of spatial locations, $f(x_i, x_j)$ is a feature dependent weighting function, and $\mathscr{C}(x)$ is a normalization function.

Two different approaches for the weighting function are explored.

- ***Gaussian:*** The first uses two learned embeddings $\theta(x)$ and $\phi(x)$ and a Gaussian weighting function: $f(x_i, x_j) = e^{\frac{1}{\sqrt{d}}\theta(x_i)^T \phi(x_j)}$ and $\mathscr{C} = \sum_{\forall j \in \mathscr{L}} f(x_i, x_j)$. This normalization scheme forces $f/\mathscr{C}$ to be a softmax function.
- ***Dot product:*** The next weighting function uses $f(x_i, x_j) = x_i^T x_j$ and $\mathscr{C}(x) = N$, the number of feature values in x. Note that this scheme does not enforce the weighted mean to sum to 1, but that experiments find that this is unnecessary when using a dot product version of non-local means.

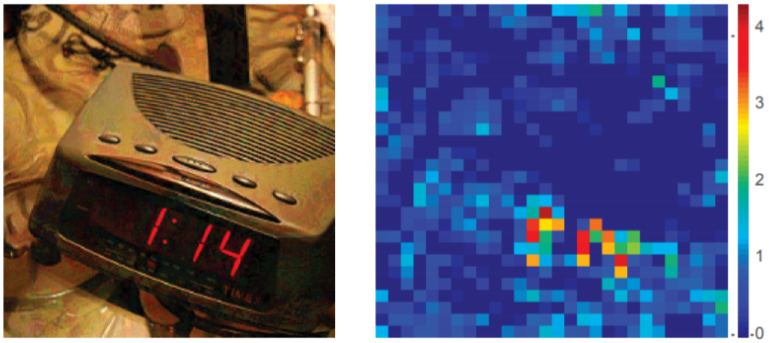
The dot product scheme requires no parameters, whereas the Gaussian weighting function requires learned embeddings for both θ, ϕ.

3 Models and Attackers

3.1 Resnet Architecture

Resnet[7] is widely accepted as a top-tier model after performing very well in many classification, detection, and localization competitions including ILSVRC 2015 and MS COCO 2015. As such, [21] chose variants of this architecture as a baseline that was modified with denoising layers. We use two variants of the Resnet architecture, Resnet-152 and Resnet-101 where the number denotes the total number of convolutional layers in the network. Each Resnet is divided into groups which can be further divided into blocks. Resnet-152 contains 4 groups, containing 3, 8, 36, and 3 blocks sequentially. Between each group, a convolutional layer with a stride of 2 and kernel size of 1 is used to reduce each dimension of the output by half.

(a) Original Image



(b) Adversarial Noisy Image

Figure 2.1: A sample image of a digital clock (top) and its noisy counterpart (bottom)[21]. To the right of the images are individual feature maps extracted from a Resnet model applied to the images on the left. The adversarial noise applied to the bottom image was produced using projected gradient descent[10] with maximum perturbation $\epsilon = 16$ (out of 256). The model predicted the original image correctly as "digital clock" while the adversarial image was misclassified as "space heater."

The number of channels in the feature map is also doubled between each group. Each block consists of 3 convolutional layers with filter sizes of 1, 3, 1 respectively. The input to the block is added to the output of these 3 layers with a skip connection, after which a ReLU activation function is applied. After the final group output, an average pooling layer is applied and connected to a linear layer which is used as the classification output of the model. For the purposes of this paper, all feature map outputs are taken as block outputs.

The Resnet-101 architecture is very similar to the Resnet-152 architecture, except that it uses 4 groups with block sizes of 3, 4, 23, and 3 respectively.

3.2 Baseline and Denoising Models

We use three models for evaluation which were designed and trained in [21]. Each model was trained using adversarial training [5] where networks are trained on adversarial examples

rather than an unperturbed example. The three models are:

- ***Resnet:*** An unmodified Resnet-152[7] architecture used as a baseline
- ***ResnetDenoise:*** A modified Resnet-152 architecture with Gaussian feature denoising blocks after each of the 4 Resnet groups
- ***ResNeXtDenoiseAll:*** A modified Resnet-101 architecture with Dot product denoising blocks after each Resnet block in all 4 Resnet groups.

It is important to note that the ResNeXt101 DenoiseAll model was the winning model of the Competition on Adversarial Attacks and Defenses 2018 (CAAD2018[1]) competition, achieving 50.6%[21] accuracy against 48 unkown attackers, beating the second place winner by ~ 10%.

3.3 Adversarial Attacker

All adversarial attacks in this paper refer to the method of Projected Gradient Descent[11] (PGD). This attack method uses the gradients of the model in order to compute an adversarial perturbation, making it a white-box attack method. PGD is a constrained optimization problem where the target image is perturbed to maximize the model's loss while keeping the size of the perturbation at most ϵ under the L^∞ norm, where ϵ is a user-defined parameter. This optimization occurs in a very similar way to backpropogation of a model's weights, except that an iteration's gradient updates are made to the input image which maximize the loss of the model. After one step of PGD, the perturbation is clipped to stay within the ϵ bound. This process is repeated for a set number of iterations or until convergence. In our case, models are evaluated on a 100 step PGD attack.

4 Image Similarity Metrics

In measuring the similarity between images and their adversarial examples, we used various image similarity metrics. These similarity metrics are outlined in the table below, along with their range of values and popular uses:

Image Similarity Metrics		
Metric	Range	Uses
Frobenius	[0, inf)	Directly comparing pixel intensity values [18]
SSIM	[0,1]	Comparison with respect to luminance, structure, and contrast [13]
MSE	[0, inf)	Directly comparing pixel insensity values [19]
SRE	[0, inf)	Comparing images of varying brightness[9]

4.1 Frobenius Distance

For two input n-dimensional images A and B, the Frobenius distance, also referred to as the Euclidean distance, is defined as :

$$F(A,B) = ||A-B||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}-b_{ij}|^2}$$

Where a_{ij} and b_{ij} represent individual pixel intensities in A and B's image matrices. For two images, this equation can be utilized to compute the distance using only pixel intensity values. The raw outputs of this metric vary as a function of the observed magnitudes and are very sensitive to distances between individual pixels. This Frobenius distance does not take into consideration spatial relationships and is thus not a popular metric for identifying image quality differences. It is, however, a popular image similarity metric because of its simple formulation.[18]

4.2 SSIM

The Structural Similarity Index Measure [19] is an image similarity metric that utilizes image contrast, structure, and luminance to compute an index between 0 and 1, with a value of 1 indicating that images are very structurally similar or identical and a value of 0 indicating that images are very different. SSIM requires a reference image and a processed version of that image to make these comparisons. The equation for computing this index, utilizing formulas for contrast, structure, and luminance are defined as follows for two images A and B, where either A or B is the reference.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where μ_i is the mean of image i, σ_i is the variance of image i, and C_1 and C_2 are used to stabilize division.

From this formual for SSIM, one can see the individual components of the previously mentioned quantities, since luminance per-pixel is defined as :

$$l(x,y) = \frac{2\mu_A\mu_B + C_1}{\mu_A^2 + \mu_B^2 + C_1},$$

structure is defined as :

$$s(x,y) = \frac{\sigma_{AB} + C_3}{\sigma_A\sigma_B + C_3}$$

and contrast is defined as:

$$c(x,y) = \frac{2\sigma_A\sigma_B + C_2}{\sigma_A^2 + \sigma_B^2 + C_2}$$

While SSIM produces useful results in some cases, it can generate counterintuitive results in others. Research has shown that when images have low luminance values and when when the local distribution of pixel values differ very little, outputs can be counterintuitive [13].

4.3 MSE

The Mean Squared Error (MSE) is a very popular metric for image similarity and is defined for two images A and B as:

$$MSE(A,B) = \frac{1}{n}\sum_{i}^{n}\sum_{j}^{m}(a_{ij} - b_{ij})^2$$

where a_{ij} and b_{ij} represent the pixel intensities of the image vectors for A and B. Similar to Euclidean distance, MSE is very sensitive to small changes in pixel intensity and is not good at identifying image quality. [19]

4.4 SRE

Signal to Reconstruction Error Ratio (SRE) [9] is an image similarity metric that measures errors relative to the mean image pixel intensity. For an n-dimensional reference image matrix A and an n-dimensional reconstructed image matrix B, their SRE is computed as:

$$SRE(A,B) = 10log_{10}\frac{\mu_A^2}{||B-A||^2/n}$$

where μ_A^2 is the squared average value of A, and $||B-A||$ is the Frobenius norm between A and B. As this metric explores errors relative to mean intensity, it is typically used to compute errors for images of varying brightness. Similar to the Euclidean and MSE metrics, SRE does not take into consideration spatial relationships between pixels and thus can produce counterintutive results when evaluating images with slight pixel transformations.[9]

5 Experiments

5.1 Setup

Three pretrained models were acquired from the associated github repository for [21]. All models were trained on the ILSVRC-2012 dataset[14] using an adversarial training method where each network was given adversarially perturbed images from a PGD attacker. The

models used for evaluation are Resnet, ResnetDenoise, and ResNeXtDenoiseAll as described in section 3.2

We evaluated 100 images from the ILSVRC-2012 validation dataset on each of the three models. These images were evaluated in three different settings: the original image, perturbed with random noise, and perturbed with adversarial noise. This noise is bounded by an L^∞ norm of $\epsilon = 16$ out of the possible 256 values of each pixel. The adversarial noise is computed using a 100-step PGD attack. Random noise is generated by a random perturbation of each pixel within the L^∞ bound for the same ϵ as the adversarial noise case. For each evaluated image, we extracted the layer outputs for each Resnet block.

For each output layer of a Resnet block, we calculated the similarity metric between each of the two noisy images and the corresponding layer output of the original image. This similarity was calculated by treating the feature maps of the original input as the true image and the feature maps of perturbed inputs as reconstructions. This distinction is necessary to account for similarity metrics which are not symmetric with respect to the image inputs.

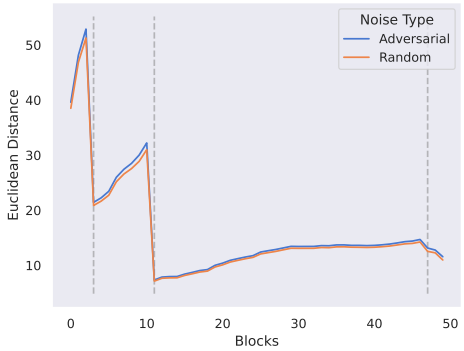
Numpy[6] was used to evaluate the Euclidean and MSE similarity metrics, while the image-similarity-measures[12] package was used to evaluate SRE and skimage[17] evaluated the SSIM metric.

5.2 Results

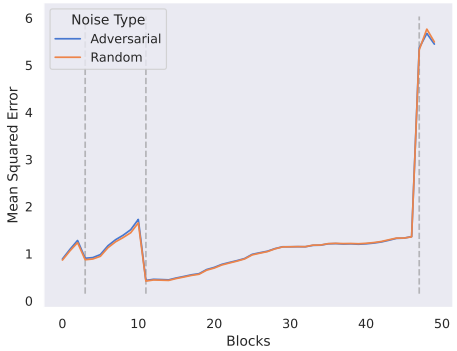
Figure 5.1 shows computed image similarity values for each block as a function of the block depth. Figure 5.1*a* shows a positive relationship between block depth and distance between the original input, but the relative distance changes at each new Resnet group. The Resnet architecutre suggests that this difference between Resnet blocks is due to changes in the feature map dimensions. A similar trend can be seen in 5.1*b* where mean squared error drastically increases in the last group. Signal Reconstruction Error in 5.1*c* follows the same relationship but has slightly greater error for random noise throughout deeper layers in the third and largest Resnet group. Note that Figure 5.1*d*'s SSIM metric has an inverted relationship to the rest of the metrics, where lower values indicate lower similarity. This plot shows the opposite relationship between random and adversarial noise as 5.1*c* since adversarial noise has less similarity in deep blocks of the large Resnet group whereas the SRE ratio was lower in these same locations.

We next compare the effects of denoising models with the baseline models in Figure 5.2. Each model in 5.2*a* exhibits the same relationship between block depth and Euclidean distance, but at different relative scales. Random and adversarial noise closely match for each model. Figure 5.2*b* also shows a similar trend with different scales across each model. SRE in 5.2*c* changes significantly between the different models in the largest Resnet group between 20 and 3 blocks to the end of the network. For both adversarial and random noise types, the baseline Resnet model's error increases with depth, ResnetDenoise's error decreases within this group, while ResNeXtDenoiseAll's error stays at a constant low value. Figure 5.2*d* shows an inversion of the relationship between random and adversarial noise for the denoising models

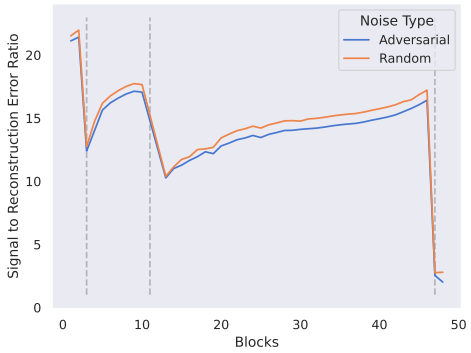
in the largest Resnet group. Whereas the denoising models achieve a higher similarity for the random noise type compared with the Resnet baseline, their similarity is lower than the baseline for the adversarial noise type.



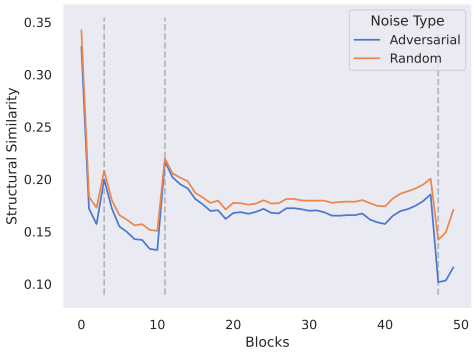
(a) Euclidean



(b) Mean Squared Error



(c) Signal to Reconstruction Error Ratio



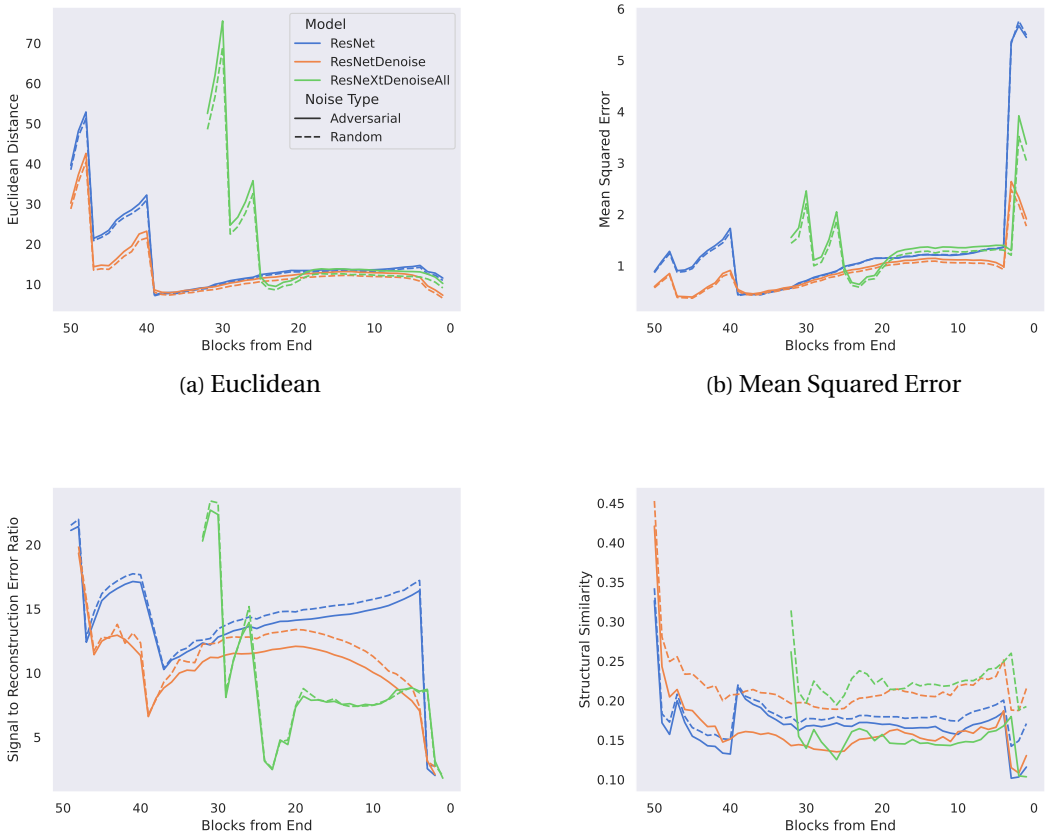
(d) Structural Similarity Index Measure

Figure 5.1: Feature map distance of the baseline Resnet at each block in the model architecture. Distances for a specific block are averaged over all channels of 100 example inputs. Distances are shown for random perturbations to the input (orange) and for adversarial perturbations (blue). Vertical gray lines delineate separation of Resnet groups. Subfigures $5.1a - 5.1d$ denote distance metrics for Euclidean, MSE, SRE, and SSIM respectively. Low values correspond to more similar feature maps in all metrics except SSIM, where lower values indicate less similarity.

6 Discussion

6.1 Resnet baseline model

Exploring the relationship between the feature map distances of the baseline Resnet and each block in the model's architecture reveals a lot about the nature of the propagation of adversar-



(a) Euclidean

(b) Mean Squared Error

(c) Signal to Reconstruction Error Ratio

(d) Structural Similarity Index Measure

Figure 5.2: Comparisons for feature map distances among 3 different Resnet models. Distances are averaged over all channels of the example set. Models are shown in different colors following the legend in 5.2*a*. Both adversarial and random noise perturbations are shown as solid and dashed lines respectively. The x-axis aligns the final block of each network to account for Resnet architectures of various block depths.

ial noise throughout our model relative to random noise. As outlined in [21], the perturbation of the features induced by an adversarial image gradually increases as the image is propagated through the network. This fact is highlighted in the results of Figure 5.1, in which for each similarity metric, we notice decreasing similarity trends between block groups of the Resnet model.

Similarity trends in these figures are split between Resnet groups because of the feature reduction that occurs between each of these groups. Changes to the scaling of the metric at each group can be attributed to halving of the image size. As images are assumed to be from the same distribution, with less pixels used to evaluate similarity, values are decreased between groups.

While the plots for each image similarity metric in the Resnet baseline model (Figure 5.1) exhibit similar trends with respect to Resnet group structure, the relationship between the randomly perturbed image and the adversarial perturbed image differs between the plots. This relationship can be explained by the nature of the metrics. Notably, the Euclidean plot (Figure 5.1*a*) and MSE plot (Figure 5.1*b*) have random noise similarity metrics that are almost completely identical to that of their adversarial counterparts. The inability of these metrics to distinguish between noise types is attributed to the method of solely computing distances between pixel intensity values and are are thus not able to give any information about image degradation or quality.

Randomly perturbed images in Figure 5.1*c* appear to have higher SRE values than the adversarial perturbed images because SRE values are determined by the ability of the perturbed images to "reconstruct" themselves to align to their original image. As the baseline Resnet model is trained on adversarial examples, it is better able to reconstruct the original feature maps of the adversarial input more frequently.

The Structural Similarity Index measure for the Resnet baseline model (Figure 5.1*d*) shows trends that imply feature maps from randomly perturbed inputs are *more* similar to the original feature maps than their adversarial counterparts. Contrary to Euclidean, MSE, and SRE similarity metrics, SSIM evaluates local difference in image structure and is very sensitive to local changes between perturbed images and original images. This relationship between adversarial and random noise as seen in our figure suggests that adversarial noise that is applied to images causes increased local changes over the depth of the Resnet neural network compared to random noise applied to images.

6.2 Comparing the Three Resnet Models

Figure 5.2 explores the effectiveness of denoising as evaluated by our image similarity metrics. While the relationship trends for Euclidean and MSE appear to be quite similar, SRE and SSIM reveal new and interesting properties of the adversarial noise throughout neural networks.

The results from evaluating the denoising models with both the Euclidean similarity metric (Figure 5.2*a*) and the MSE metric (Figure 5.2*b*) align with results cited in [21]. As expected, the Resnet denoising models (ResNetDenoise and ResNeXtDenoiseAll) have feature maps that are more similar to the original feature maps relative to the baseline Resnet model. As with the Resnet baseline model, both the Euclidean and MSE metrics don't seem to distinguish well between noise type as a result of these metrics relying solely on distances between image pixel intensity values.

The SRE metric (Figure 5.2*c*), unlike the Euclidean and MSE metrics, exhibits different behavior with respect to depth of the neural network for the denoising models. While the ResnetDenoise model initially has a higher SRE, it becomes smaller than the ResneXtDenoiseAll's error by the end of the network. As ResnetDenoise relies on less denoising periods than ResneXtDenoiseAll, this suggests that the Resnet models learn to better reconstruct original feature maps from the

image after occasional denoising. Since this trend follows for both random and adversarial noise, this suggests that denoising models learn to reconstruct original feature maps for images perturbed by any kind of noise.

As seen in Figure 5.2*d*, structural similarity for a randomly perturbed image appears to be higher than that of its adversarial image for both of the denoising models. As the SSIM evaluates image similarity at a local level with respect to structure, this behavior suggests that the denoising models are better able to reconstruct original features for random noise than for adversarial noise. Additionally, the denoising models appear to increase the difference in structural similarity between random noise and adversarial noise relative to the Resnet baseline model. Denoising is effective for random noise and is able to increase structural similarity as intended, but fails to follow the same trend with adversarial noise. This suggests that adversarial noise makes changes to feature structure that are not resolved with the denoising models. This indicates that adversarial perturbations cause changes to feature maps that are not explained purely as noise.

7 Conclusion

Our results show us both properties of adversarial examples that align with state-of-the-art results and new results that point toward further analysis of adversarial noise and their changes to intermediate features.

Euclidean distance and MSE didn't allow for similarity comparisons that revealed new information. The use of these image similarity metrics to compare layer outputs failed to give meaningful interpretations of changes in structure beyond pixel intensity. While denoising models are effective in minimizing both Euclidean distance and MSE, these metrics were unable to distinguish between images that were perturbed with random noise and those that were perturbed with adversarial noise through our model.

The SRE metric was able to better distinguish between random noise and adversarial noise. The results from evaluating with respect to this metric suggest that the ResNet models learn to better reconstruct original features from the image after occasional denoising.

Perhaps the most notable result in this experiment is from the results of the SSIM metric. This metric's sensitivity to local pixel changes and image structure revealed structural changes in feature maps that were induced by adversarial noise. While both our experiment and the experiment outlined in [21] show that feature denoising is effective in improving adversarial robustness, this result suggests that denoising might not address deeper effects of adversarial perturbation. Adversarial noise makes unexpected changes to feature map structure, beyond just noise.

8 FUTURE WORK

The results of this paper present many questions which could be addressed in future research projects.

In this experiment, Structural Similarity Index (SSIM) was effective in revealing structural change to the feature maps throughout the baseline and denoising Resnet models. To explore more about structural changes, it would be beneficial to explore different image similarity metrics in addition to the four evaluated in this paper. Other similarity metrics that could be explored include the Information theoretic-based Statistic Similarity Measure (ISSM)[2], Universal image quality index (UIQ)[20], and Feature-based similarity index (FSIM)[22]. Qualitative analyses of the induced feature maps may also provide insight to induced structures from adversarial perturbation.

Further work investigating the changing structure of an input's feature map specifically induced by adversarial noise could be joined with exploration of varying model structure. Experiments that explore the effects of adversarial perturbation against different types of filtering methods beyond denoising, such as noising or combinations of both, could yield models that are more robust to adversarial attacks.

REFERENCES

[1] Competition on adversarial attacks and defenses, 2018.

[2] Mohammed Aljanabi, Zahir Hussain, and Noor Shnain. Design of a hybrid measure for image similarity: a statistical, algebraic, and information-theoretic approach. *European Journal of Remote Sensing*, 52:1–14, 07 2019.

[3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65 vol. 2, 2005.

[4] Nicholas Carlini and David A. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *CoRR*, abs/1801.01944, 2018.

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[6] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] P. Kegelmeyer, T. Shead, J. Crussell, K. Rodhouse, D. Robinson, C. Johnson, D. Zage, W. Davis, J. Wendt, J. Doak, T. Cayton, R. Colbaugh, K. Glass, B. Jones, and J. Shelbur. Counter adversarial data analytics, 2015.

[9] Charis Lanaras, José Bioucas-Dias, Silvano Galliani, Emmanuel Baltsavias, and Konrad Schindler. Super-resolution of sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146:305–319, 2018.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[12] M. U. Müller, N. Ekhtiari, R. M. Almeida, and C. Rieke. Super-resolution of multispectral satellite images using convolutional neural networks. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-1-2020:33–40, Aug 2020.

[13] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim, 2020.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[15] Austin Short, Trevor La Pay, and Apurva Gandhi. Defending against adversarial examples. 2019.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[17] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[18] Liwei Wang, Yan Zhang, and Jufu Feng. On the euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1334–1339, 2005.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[20] Zhou Wang and A.C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[21] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *CoRR*, abs/1812.03411, 2018.

[22] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.