Exploring Properties of the Deep Image Prior

Andreas Kattamis
University of Cambridge
ak960@alumni.cam.ac.uk

Tameem Adel
University of Cambridge
tah47@cam.ac.uk

Adrian Weller
University of Cambridge
aw665@cam.ac.uk

Abstract

The Deep Image Prior (DIP, Ulyanov et al., 2017) is a fascinating recent approach for recovering images which appear natural, yet is not fully understood. This work aims at shedding some further light on this approach by investigating the properties of the early outputs of the DIP. First, we show that these early iterations demonstrate invariance to adversarial perturbations by classifying progressive DIP outputs and using a novel saliency map approach. Next we explore using DIP as a defence against adversaries, showing good potential. Finally, we examine the adversarial invariancy of the early DIP outputs, and hypothesize that these outputs may remove non-robust image features. By comparing classification confidence values we show some evidence confirming this hypothesis.

1 Introduction

Ulyanov et al. (2017) surprisingly showed that just the structure of a convolutional neural network is capable of capturing a good portion of images' statistics. They demonstrated that starting with an untrained network, and then training to guide the output towards a specific target image for image restoration tasks such as denoising, super-resolution, and in-painting achieved performance which is comparable to state-of-the-art approaches. Their approach, termed the Deep Image Prior (DIP), shows that the architecture of a network can act as a powerful prior.

The same network has been found to have excellent performance as a natural image prior (Ulyanov et al., 2017). The ability to detect natural images poses great significance in recent years, especially with the increasing security concerns raised over natural-looking images that are not correctly classified, called adversarial examples (Szegedy et al., 2013). These adversarial examples can be thought of as incremental, non-natural perturbations. As such, using the Deep Image Prior as a recovery method can indicate its ability to work as a natural denoiser, a hypothesis that will initially be tested. Furthermore, we use the Deep Image Prior to develop an adversarial defence, thereby investigating its potential.

Then, we investigate the early iterations of the network by producing saliency maps of the Deep Image Prior outputs (DIP outputs). Saliency maps show the pixels which are most salient (relevant) in reaching a target classification. We hope to show that the salient pixels gather to display more clear, distinct, and robust features of the images.

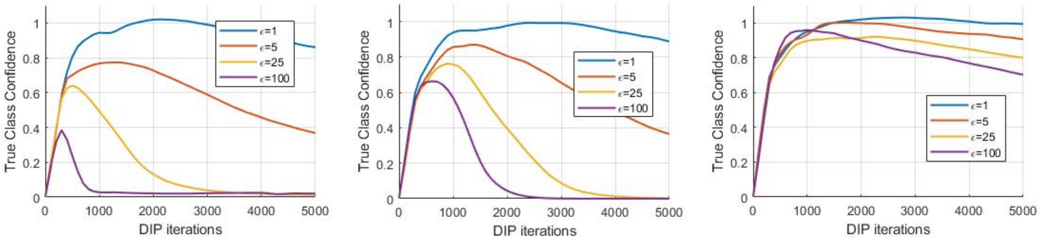
Recently, Ilyas et al. (2019) showed that adversarial examples are a result of the existence of non-robust features in the images, which are highly predictive, yet incomprehensible to humans. The successful performance of the Deep Image Prior in recovering the original classes from adversarial examples (Ilyas et al., 2017), raises the argument that the Deep Image Prior produces images that have 'dropped' their non-robust features and are left with the robust image features. To test this theory we directly use the dataset from Ilyas et al. (2019) consisting of robust and non-robust image features, and passing these through the Deep Image Prior. By comparing the DIP outputs of the robust and non-robust image datasets, we hope to see evidence towards the ability of the Deep Image Prior to select robust images.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2 Recovering from adversaries

As a prerequisite we first show the ability of the Deep Image Prior to recover from adversarial perturbations. For this investigation, three methods for generating adversarial examples will be considered: the Fast Gradient-Sign Method (FGSM) (Goodfellow et al., 2015), the Basic Iterative method (BI), and the Least-Likely Class Iterative method (LLCI) (Kurakin et al., 2016).

Adversarial examples were generated for 20 images using the three methods for various adversarial strengths. The DIP output was collected and classified every 100 iterations and the values of the true class confidence were obtained. The classifier used was ResNet18 (He et al., 2015a). The results are shown in Figure 1. More accurate classifiers were also briefly considered, such as Inception-v3 (Szegedy et al., 2015), but no qualitative differences were found.



(a) FGSM adversaries (b) BI adversaries (c) LLCI adversaries

Figure 1: Different adversaries passed through the Deep Image Prior and classified every 100 iterations for three adversaries and various adversarial strengths, $\epsilon \in \{1, 5, 25, 100\}$

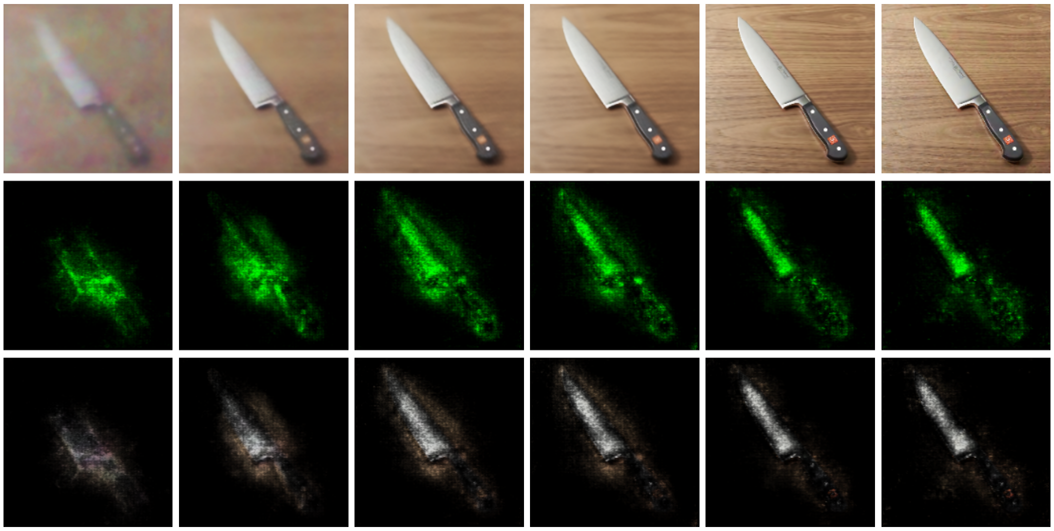
It is apparent that the DIP outputs in the earlier iterations allow the classifier to recover the true class, as evident from the peaks in the confidence values. Then, the network output converges back to the adversary observed through the decrease in confidence values. The number of iterations at which peak confidence occurs appears to be different among adversaries and also among adversarial strengths. Nevertheless the theme is similar; exhibiting peaks at the earlier DIP iterations, showing the ability of the network to recover from adversarial perturbations.

3 Interpreting DIP outputs

We have shown that the DIP can be an interesting and effective tool for recovering from adversarial perturbations. However, details about the iterative process and the transformation of the input into the various DIP outputs are still unknown. To test the nature of these outputs we introduce a novel saliency map approach, termed MIG-SG. This method is a variation of the integrated gradients approach (Sundararajan et al., 2017) while using SmoothGrad (Smilkov et al., 2017). More information about this method can be found in the Appendix.

Figure 2 shows, on a step-by-step basis, how the salient features of the image change for progressive DIP outputs. While the image is very blurry after 300-400 iterations, the saliency map already shows that all the key features of the knife have already been identified. This is confirmed by the confidence of the DIP output, which increased to > 0:5 after just 200 iterations. On the contrary, observing the outputs at 2000 and 4000 iterations shows that salient features have become more focused on the blade of the knife. Previously, the salient features focused on the handle and the butt of the blade as observed from the bottom row of images in Figure 2. Furthermore, it is no longer clear what the salient features represent, a fact also illustrated in the decreasing value of the true class confidence. Overall, the salient maps "lost" various salient features as the DIP output was converging towards the adversarial example.

Overall, with the clearest saliency maps observed at low iteration numbers (300-400), we observe evidence that the early DIP outputs are invariant towards adversarial effects.



(a) 100 iterations Conf.: 0.004 (b) 200 iterations Conf.: 0.52 (c) 300 iterations Conf.: 0.72 (d) 400 iterations Conf.: 0.72 (e) 2000 iterations Conf.: 0.68 (f) 4000 iterations Conf.: 0.55

Figure 2: DIP outputs and saliency maps for selected iterations using the MIG-SG method. Top row shows the DIP output, middle row shows the saliency mask, and bottom row shows the saliency mask overlaid on the original image.

4 Deep Image Prior as an adversarial defence

To mount a defence using the Deep Image Prior we aim to transform the input of the classifier to a state where the adversarial perturbations are not detectable. The classifier used for this investigation was ResNet18 (He et al., 2015a). By using a simple classifier to make this defence, we are able to evaluate the potential of the Deep Image Prior to recover the original class from adversarial perturbations individually. Our results are compared against the defence from Xie et al. (2017) that uses randomisation to defend against adversarial perturbations, and which also uses a similar evaluation method.

Understandably, using the Deep Image Prior decreases the accuracy of the network on clean images. From Table 1, there is a noticeable decrease in top-1 accuracy, especially when using fewer DIP iterations. As the number of iterations is increased, the top-1 accuracy increases with it, at a loss of computational speed. Since the computational costs are already very high, the defence was not tested for larger iteration numbers, as that would make it slow and impractical.

Table 1: Top-1 classification accuracy for the clean dataset.

	Without DIP	DIP output 500 iterations	DIP output 750 iterations	DIP output 1000 iterations
Clean image dataset	100%	67.5%	75.3%	79.6%

The results of using the Deep Image Prior on adversarial examples are shown in Table 2 and display a very competitive performance with the reference defence method, having a higher accuracy across all three adversaries used in that comparison. The average accuracy is highest after 1000 iterations, however, this is not best for all the adversaries as observed from the FGSM adversary with $\epsilon = 10$.

Overall, we see a decreased ability to correctly classify images, combined with an increased ability to defend against adversaries. This result, is similar to the one described by Ilyas et al. (2019), where the classifier trained on the robust image dataset, highlighting the ability of the Deep Image Prior to select these robust features.

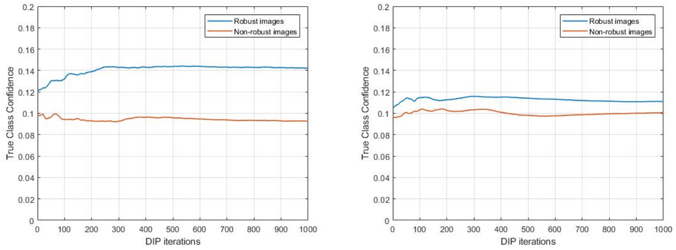
Table 2: Top-1 classification accuracy for different adversarial attacks; best defences shown in bold.

Adversary	Adv. Str. ϵ	No Defence	Using randomisation operations (Xie et al., 2017)	(ours) DIP using 500 it.	(ours) DIP using 750 it.	(ours) DIP using 1000 it.
FGSM	2	21.7%	65.1%	66.2%	73.1%	**75.1**%
FGSM	5	3.5%	54.5%	61.2%	69.0%	**70.1**%
FGSM	10	2.6%	52.4%	58.4%	**62.9**%	59.9%
BI	10	0.2%	-	61.8%	66.2%	**69.2**%
LLCI	10	1.1%	-	64.2%	73.5%	**78.5**%
Average	N/A	5.8%	57.3%	62.3%	68.9%	**70.6**%

5 Using the robust image dataset

For this test, We used the pre-generated robust and non-robust image datasets from Ilyas et al. (2019). The architecture used for the Deep Image Prior had to be altered since CIFAR-10 images were used. Details can be found in the Appendix. The outputs were evaluated through the classification confidence of the original class of the image.

Both figures in 3 show the difference between the classification of robust and non-robust datasets to an external classifier, where robust images hold more information about the true class compared to the non-robust images. Regarding the response at the earlier iteration numbers, it is very subtle, yet we see some evidence to support our hypothesis. The non-robust image datasets show a trough before converging to their final classification confidence, while the robust image datasets shows a peak in confidence, indicating that the the convergence of the network towards the robust images was faster than the convergence on the non-robust ones.



(a) Architecture-1

(b) Architecture-2

Figure 3: True class confidence against DIP iterations, for the two architectures

6 Discussion on architecture & Conclusions

Ulyanov et al. (2017) showed that the DIP achieved remarkable results, apparently due to the prior encoded in the network architecture. To test this, we evaluated the sensitivity of DIP to changes in network architecture. Surprisingly, we found that while some sensitivity exists, it is not high, with various architecture changes showing little to no changes in performance. However, some changes showed great influence on the performance of the network. In particular, the network was found to fail when no skip connections were used, or when a very shallow network was used. Nevertheless, no evidence was found that can describe this sensitivity as a “resonance", as stated in the original paper. See Appendix for details.

We observed the network’s ability to successfully recover from adversarial perturbations, caused by the resistance of the early DIP outputs to adversarial perturbations. This was further observed from looking at appropriate saliency maps, where we introduced a new method. Consequently, the network was found to create a promising adversarial defence. Lastly, we provided evidence for the ability of the Deep Image Prior to select robust image features over non-robust features in its early iterations, as defined by Ilyas et al. (2019).

Acknowledgements

AW acknowledges support from the David MacKay Newton research fellowship at Darwin College, The Alan Turing Institute under EPSRC grant EP/N510129/1 & TU/B/000074, and the Leverhulme Trust via the CFI.

References

Goodfellow, I., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015a). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

He, K., Zhang, X., Ren, S., and Sun, J. (2015b). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852.

Ilyas, A., Jalal, A., Asteri, E., Daskalakis, C., and Dimakis, A. G. (2017). The robust manifold defense: Adversarial training using generative models. *CoRR*, abs/1712.09196.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2016). Adversarial examples in the physical world. *CoRR*, abs/1607.02533.

Smilkov, D., Thorat, N., Kim, B., Viégas, F. B., and Wattenberg, M. (2017). Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2013). Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Deep image prior. *arXiv preprint arXiv:1711.10925*.

Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. L. (2017). Mitigating adversarial effects through randomization. *CoRR*, abs/1711.01991.

A Appendix: Supplementary information for Exploring Properties of the DIP

A.1 Modified Integrated Gradients with SmoothGrad

As the name suggests, this method performs integration between a baseline and our image, numerically, by calculating the forward derivative of the network. SmoothGrad calculates this forward derivative, by performing the differentiation on a noisy version of the input, and averaging the derivative over multiple samples (Smilkov et al., 2017). As a result, combining the two methods appears to yield significantly improved saliency maps.

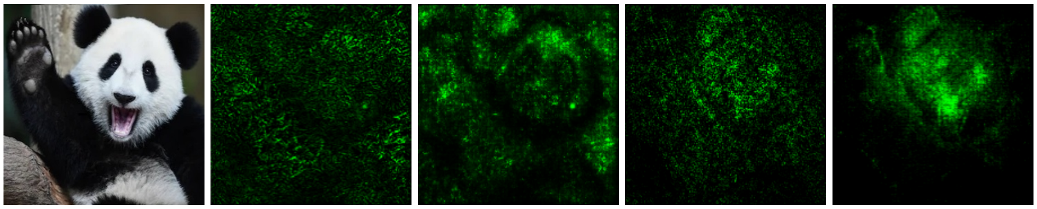
Since we are performing integration, solely taking the absolute value of the result of the grad function, failed to produce results. However, a small modification was made to the algorithm in an attempt to stop the method from failing. By also taking the absolute value of the final result, the method produced very promising results. Using the absolute values of the gradients for coloured images, enables negative gradients to also be contribute to the salient features of the image.

Mathematically our saliency method can be expressed as:

$$S^i_{MIG-SG}(\boldsymbol{x}, t) = \left| \frac{(x^i - x^i_0)}{mN} \sum_{k=0}^{m} \sum_{n=1}^{N} \left| \nabla^i_{\boldsymbol{x}} F_t\big(\boldsymbol{x}_0 + \frac{k}{m}(\boldsymbol{x} - \boldsymbol{x}_0) + \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})\big) \right| \right| \tag{1}$$

where $\boldsymbol{x}$ is the input image, $\boldsymbol{x_0}$ is the baseline image and is only used for comparative purposes (Sundararajan et al., 2017). Additionally, m is the number of integration steps and N is the number of samples used in the computation of SmoothGrad (Smilkov et al., 2017). Lastly, $F_t(\boldsymbol{x})$ returns the classification result of class t before the cost function is calculated.

Common saliency maps have been generated for a panda image, shown in Figure 4. The MIG-SG saliency map is observed in Figure 4e and while it can definitely appear as a scary image, it provides very interesting information about the panda. This saliency map, instead of picking up all the panda in the image, has instead focused on its characteristic features, the eyes and the mouth. This makes it a very useful tool to visually interpret images.



(a) Original (b) Gradients (c) SmoothGrad (d) IG (e) MIG-SG

Figure 4: Saliency maps for an image of a panda using all the methods considered in this report.

A.2 Adversarial defence model

For the Deep Image Prior, the original architecture was used (Ulyanov et al., 2017). The number of iterations was left at a low value, as the results of this work suggested that the DIP output is less sensitive to adversarial perturbations at earlier iterations. Three iteration numbers were tested: 500, 750 and 1000. The tests were conducted on a dataset of images from 200 randomly selected classes from the ImageNet database. From this dataset, 500 images correctly classified using the ResNet18 classifier were then chosen to test the performance of our defence.

The diagram of the defence is shown in Figure 5.

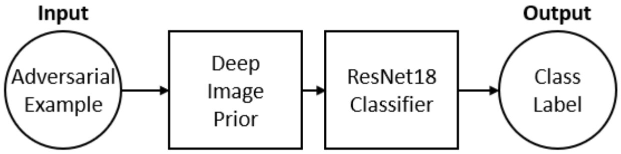


Figure 5: Diagram of the adversarial defence algorithm.

A.3 Architectures used for investigation into robust and non-robust datasets

Two architectures were considered, the first is the one used in the original paper of the Deep Image Prior (Ulyanov et al., 2017) but with the encoder depth changed from 5 to 3, to allow for the decreased dimensionality of the CIFAR-10 images. The second architecture, uses only 16 feature maps in each layer, compared with the original number which was 128, while also the encoder depth was kept at 3.

Architecture-1 and architecture-2 can be found in Tables 3 and 4 respectively.

Table 3: Architecture-1

CNN components	
Layers of encoder	3
Layers of decoder	3
Features maps per layer	128
Skip connections per layer between encoder-decoder	4
Convolutional kernel size	3-by-3
Total number of parameters	2217831
Downsampler	Convolution stage with stride = 2
Upsampler	Bilinear
Activation function	Leaky ReLU (He et al., 2015b)
Batch Normalisation? (Ioffe and Szegedy, 2015)	Yes
Input	
Input type	Uniform noise [0, 0.1]
Input Dimensionality (Input Depth)	32 × Image Size
Iteration noise standard deviation	1/30
Optimisation over parameters	
Optimizer	Adam (Kingma and Ba, 2015)
Learning rate	0.01

Table 4: Architecture-2

CNN components	
Layers of encoder	3
Layers of decoder	3
Features maps per layer	16
Skip connections per layer between encoder-decoder	4
Convolutional kernel size	3-by-3
Total number of parameters	2217831
Downsampler	Convolution stage with stride = 2
Upsampler	Bilinear
Activation function	Leaky ReLU (He et al., 2015b)
Batch Normalisation? (Ioffe and Szegedy, 2015)	Yes
Input	
Input type	Uniform noise [0, 0.1]
Input Dimensionality (Input Depth)	32 × Image Size
Iteration noise standard deviation	1/30
Optimisation over parameters	
Optimizer	Adam (Kingma and Ba, 2015)
Learning rate	0.01