Evaluating Robust Perceptual Losses for Image Reconstruction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Nowadays, many deep neural networks (DNNs) for image reconstructing tasks are trained using a combination of pixel-wise loss functions and perceptual image losses like learned perceptual image patch similarity (LPIPS). As these perceptual image losses compare the features of a pre-trained DNN, it is unsurprising that they are vulnerable to adversarial examples. It is known that: (i) DNNs can be robustified against adversarial examples using adversarial training, and (ii) adversarial examples are imperceptible by the human eye. Thus, we hypothesize that perceptual metrics, based on a robustly trained DNN, are more aligned with human perception than those based on non-robust models. Our extensive experiments on an image super resolution task show, however, that this is not the case. We observe that models trained with a robust perceptual loss tend to produce more artifacts in the reconstructed image. Furthermore, we were unable to find reliable image similarity metrics or evaluation methods to quantify these observations (which are known open problems).

1 Introduction

For image reconstruction methods that involve deep neural networks (DNNs), the choice of the loss function is crucial for good reconstruction results. Pixel-wise loss functions, like the mean squared error (MSE) loss, are often a reasonable choice, but often do not completely align with human perception. For example, it is well known that using the pixel-wise MSE loss leads to blurry reconstructions over reconstructions having misplaced edges. Nowadays, modern image reconstruction networks are usually trained to minimize a combination of different loss functions. While pixel-wise losses are still a crucial ingredient (because they create a strong error signal), the reconstruction quality can be greatly improved by simultaneously minimizing multiple losses, e.g., an adversarial loss or a perceptual loss. In this work, we focus on the latter.

1.1 Perceptual Losses

Instead of considering the error in pixel space, a perceptual loss calculates the error between the feature maps of an already pre-trained and fixed classification DNN. This was first done by Johnson et al. [1] who define a perceptual loss between to images $x \in \mathbb{R}^{c \times h \times w}$ and $y \in \mathbb{R}^{c \times h \times w}$ as

$$\mathcal{L}^{\Phi,p}_{\text{perceptual}}(x, y) = \frac{1}{c_p h_p w_p} \left\| \Phi^{(p)}(x) - \Phi^{(p)}(y) \right\|_F^2, \tag{1}$$

where $\Phi^{(p)} : \mathbb{R}^{c \times h \times w} \to \mathbb{R}^{c_p \times (h_p w_p)}$ denotes the flattened feature mapping of the VGG-19 net [2] after the p-th layer and $\|\cdot\|_F$ denotes the Frobenius norm. This is motivated by the fact that the feature maps learned by a classifier specialize on different concepts. Usually, the first layers of

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

a DNN detect low-level features like edges and corners, while the deeper layers detect high-level features that represent more abstract concepts. Thus, it is common practice to combine the perceptual loss of different layers into a single perceptual loss, e.g., the perceptual loss as a sum of perceptual losses $\mathcal{L}^{\Phi}_{\text{perceptual}}(x, y) = \sum_{p \in \mathcal{P}} \lambda_p \mathcal{L}^{\Phi,p}_{\text{perceptual}}(x, y)$, where $\mathcal{P} \subset \mathbb{N}$ is a set of layer indices and positive weights $\lambda_p > 0$ for $p \in \mathcal{P}$.

A different way of defining a perceptual loss that considers local texture information is the texture matching loss which compares the Gram matrices of the feature maps [3]. Sajjadi et al. [4] define this loss for k patches $x^{(1)}, \ldots, x^{(k)}$ and $y^{(1)}, \ldots, y^{(k)}$ of the input images x and y, respectively, as

$$\mathcal{L}^{\Phi,p}_{\text{texture-matching}}(x, y) = \frac{1}{kc_p^2} \sum_{i=1}^{k} \left\| G^{(p)}(x^{(i)}) - G^{(p)}(y^{(i)}) \right\|_F^2, \tag{2}$$

where $G^{(p)}$ calculates the Gram matrix of the feature map $\Phi^{(p)}$. Again, different layers can be added to form the total loss $\mathcal{L}^{\Phi}_{\text{texture-matching}}(x, y) = \sum_{p \in \mathcal{P}} \lambda_p \mathcal{L}^{\Phi,p}_{\text{texture-matching}}(x, y)$ for some $\mathcal{P} \subset \mathbb{N}$ and positive weights $\lambda_p > 0$ for $p \in \mathcal{P}$.

1.2 Adversarial Examples and Adversarial Training

It is common knowledge that DNNs can easily be fooled by adversarial examples, i.e., manipulated images that change the network's prediction [5]. These adversarial examples are usually unrecognizable by the human eye. Recent work has investigated how DNNs can be robustified against these adversarial examples. A widely-used approach is adversarial training which robustifies DNNs by including adversarial images (with the correct label) during training [6]. Usually, adversarial training relies on the PGD-K attack [6; 7] which crafts an adversarial image (that the model wrongly assigns the class label y_{target}) by iterating

$$x_{k+1} = P_\epsilon(x_k - \alpha \nabla_x \mathcal{L}^{\theta}_{\text{cross-entropy}}(x, y_{\text{target}})), \tag{3}$$

where $x_0 = x$ is the original image x , $\alpha > 0$ is the step-size, and P_ϵ is a function that projects the iterate back onto an norm-ball $\mathcal{B}_\epsilon(x)$ around x. After K iterations we obtain the adversarial example $\bar{x} = x_K$. The change made to the original image $x - \bar{x}$ is usually referred to as a perturbation.

These adversarial images combined with the true label are then included in the training dataset. The reasoning is that the DNN classifier should learn to also classify theses adversarial images correctly. In general, the resulting robust classifiers tend to have a slightly worse accuracy, but they are, at the same time, less susceptible to adversarial examples. To further strengthen a model's robustness adversarial logit pairing (ALP) [8] can be used. The key idea of ALP is that the logits of the classifier $F_\theta(x)$ should be invariant under adversarial perturbations. This is done by including an additional regularization term to the loss function used to train the model that encourages logits of adversarial examples $F_\theta(\bar{x})$ to be similar to those of clean examples $F_\theta(x)$. This regularization term is given as

$$\mathcal{L}^{\theta}_{\text{ALP}}(x, \bar{x}) = \frac{1}{d} \left\| F_\theta(x) - F_\theta(\bar{x}) \right\|_2^2, \tag{4}$$

where d is the number of classes. In this paper, we use these two ideas to train the robust classifier used for the perceptual losses.

Since DNNs can be affected by adversarial examples, it is unsurprising that perceptual losses are also susceptible to them shown by Kettunen et al. [9]. For many attacks, the human eye is not able to differentiate between the adversarial examples and the original images. This motivates our hypothesis which we analyze in this paper:

Hypothesis: *Feature maps produced by robust DNNs are more aligned with human perception and should therefore be more suitable to be used in perceptual losses in image reconstruction problems.*

Our hypothesis is loosely inspired by Rojas-Gomez et al. [10] who have shown that feature-inversion for robust networks is easier than for non-robust networks. Moreover, Ross and Doshi-Velez [11] argue that robust DNNs are more human interpretable.

Table 1: Accuracies of the robust and non-robust classifiers on the ImageNet dataset. Adversarial examples for evaluation were calculated using a (targeted) PGD-50 attack with a step-size $\alpha = 0.155$. The Euclidean norm of the adversarial perturbations was restricted to $\epsilon = 3.88$.

Model	Robust	Accuracy	Adversarial accuracy
VGG-19	✗	74.72%	0.00%
VGG-19	✓	64.85%	59.22%
ResNet-50	✗	76.63%	0.00%
ResNet-50	✓	66.60%	61.49%

2 Experimental Setup

In the following, we want to evaluate the effectiveness of robust perceptual losses for $4\times$-super-resolution of images, where we use an EnhanceNet [4] as a base model that learns to upscale the images.

Training: Similar to the original work, we train the EnhanceNet [4] on the MS COCO dataset [12] using various loss functions: a pixel-wise MSE loss (E), a perceptual loss (P), or a texture matching loss (T). These losses can also be combined, e.g., EP indicates an EnhanceNet trained with both the pixel-wise MSE (E) and the perceptual (P) losses. Both the perceptual and the texture matching loss (P and T) rely on a VGG-19 net, which is trained on the ImageNet dataset [13]. For the P loss we use feature maps of the second ($\lambda_2 = 0.2$) and the fifth ($\lambda_5 = 0.02$) convolutional layer (after applying the ReLU activation function) and for the T loss we use the feature maps of the first ($\lambda_1 = 3\times10^{-7}$), the third ($\lambda_3 = 1\times10^{-6}$), and the fifth ($\lambda_5 = 1\times10^{-6}$) convolutional layer. The texture matching loss is calculated using 16-by-16-patches.

We aim to test our hypothesis by comparing the performance of an EnhanceNet trained with a perceptual loss based on a robust VGG-19 net instead of the commonly used non-robust VGG-19 net. We do the same for the texture matching loss. When we train the EnhanceNet with the P and the T loss we use the same VGG-19 net (either robust or non-robust) for both losses. To train a robust VGG-19 net, we use adversarial training [6] based on the PGD-5 attack with a Euclidean norm constraint of 5.82. We combined this with ALP [8] with an ALP weight of 0.5 (see, Section 1.2). We carefully scaled the weights and biases of both VGG-19 nets such that the feature maps are normalized. Note, this does not change the output because of the used ReLU activation functions.

Evaluation: The EnhanceNets trained using different robust and non-robust perceptual loss functions are evaluated on common super-resolution benchmark datasets: BSD100 [14], Manga109 [15], Set14 [16], Set5 [17], and Urban100 [18]. We extensively evaluate the performance using the following metrics: PSNR, SSIM [19], gradient magnitude similarity deviation (GMSD) [20], information fidelity criterion (IFC) [21], and universal image quality index (UQI) [22].

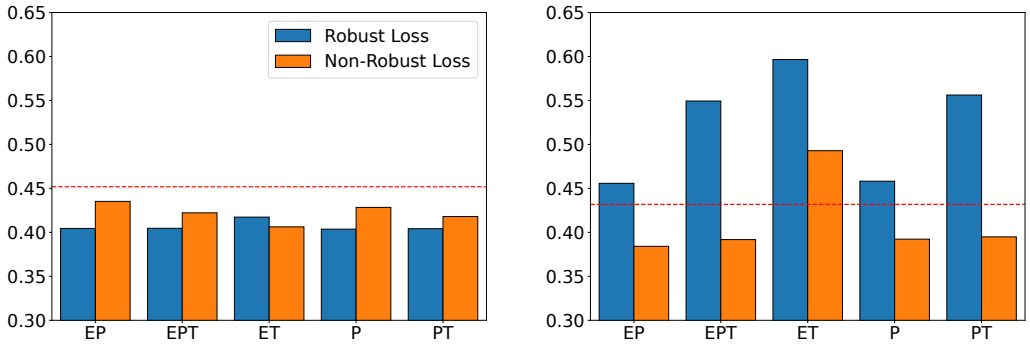
Furthermore, we use the learned perceptual image patch similarity (LPIPS) [23] that is based on a VGG-16 network [2] with additional linear layers to output a single value (the LPIPS score) for a pair of input images. The dataset used during training is collected using a two-alternative forced choice method and consists of images each having two transformed variants. Each of these triplets (original image and two variants) is accompanied by a human-made decision about which of the two variants is perceived to be more similar to the original image. The network is trained to mimic this human-made decision based on the difference of the LPIPS scores.

As an additional evaluation metric, we propose a robust LPIPS (R-LPIPS) metric which is trained in a similar manner but where the original image and both of its variants are perturbed using a PGD-10 attack constrained to a Euclidean norm of 1.33.

Besides these image similarity metrics we also compare the classification performance of a ResNet-50 [24] on sub-sampled images from the ImageNet dataset, upscaled using the differently trained EnhanceNets. On the one hand, we consider the classification performance of a robust ResNet-50, trained in a similar manner as the VGG-19 network. On the other hand, we use a non-robust ResNet-50.

Table 2: The evaluation results of the EnhanceNets trained with different robust and non-robust perceptual losses. Averaged across three runs and the five benchmark datasets.

Loss	Robust	PSNR (↑)	SSIM (↑)	UQI (↑)	IFC (↑)	GMSD (↓)	LPIPS (↓)	R-LPIPS (↓)
E	-	**26.9408**	**0.7475**	**0.9723**	0.6557	0.1698	0.2558	0.2699
EP	✗	26.3043	0.6542	0.9566	0.6488	0.1660	**0.1658**	0.2983
EP	✓	25.3076	0.6709	0.9624	0.6978	0.1674	0.1721	**0.2447**
EPT	✗	26.1829	0.6419	0.9545	0.7141	0.1649	0.1662	0.2783
EPT	✓	24.4920	0.6471	0.9562	0.7141	0.1824	0.2169	0.2936
ET	✗	24.7270	0.6790	0.9605	**0.7481**	0.1798	0.1794	0.3189
ET	✓	23.5006	0.6117	0.9506	0.7073	0.2049	0.2544	0.3913
P	✗	26.2542	0.5342	0.9351	0.6512	0.1656	0.1850	0.3096
P	✓	25.3015	0.6669	0.9619	0.6977	0.1675	0.1736	0.2452
PT	✗	26.1562	0.5209	0.9313	0.7148	**0.1646**	0.1909	0.2961
PT	✓	24.4914	0.6462	0.9561	0.7152	0.1828	0.2210	0.2924



(a) Classification Error (Robust Classifier)

(b) Classification Error (Non-Robust Classifier)

Figure 1: The classification errors on sub-sampled images from the ImageNet dataset, upscaled using the EnhanceNets trained with different robust and non-robust perceptual losses. The red dashed line indicates the classification performance of the EnhanceNet trained using the E loss. We use an robust classifier (a) and non-robust classifier (b). The choice of the classifier inverts the results.

In Table 1, we compare the accuracies on the ImageNet dataset (with and without adversarial perturbations) of the robust and non-robust VGG-19 networks used for the perceptual and texture-matching loss, as well as, the ResNet-50 used for evaluation. As expected, the robust models have a slightly lower accuracy than the non-robust ones, but they perform drastically better on the adversarial images.

3 Results

In Table 2, we quantitatively evaluate the differently trained EnhanceNets using the previously introduced metrics. For each set of losses that involved a perceptual loss, we consider the robust and the non-robust variant (indicated in the second column). We observe that the EnhanceNet trained only with the E-loss performs the best in terms of the PSNR and SSIM metrics. This is not surprising as the pixel-wise MSE loss (E) and the PSNR metric are strongly related. Comparing the EnhanceNets trained with robust and non-robust perceptual losses, we are unable to conclude whether the robust perceptual loss improves the performance, e.g., for the SSIM metric, the robust perceptual losses are, in many cases, better than their non-robust counterpart, but for the GMSD metric, they are always worse. Moreover, we observe that the EnhanceNet trained with the robust EP loss performs the best on the R-LPIPS metric.

As previously observed, it is challenging to quantitatively conclude which EnhanceNet produces the best images. Thus, following the reasoning of Sajjadi et al. [4], we argue that the classification error

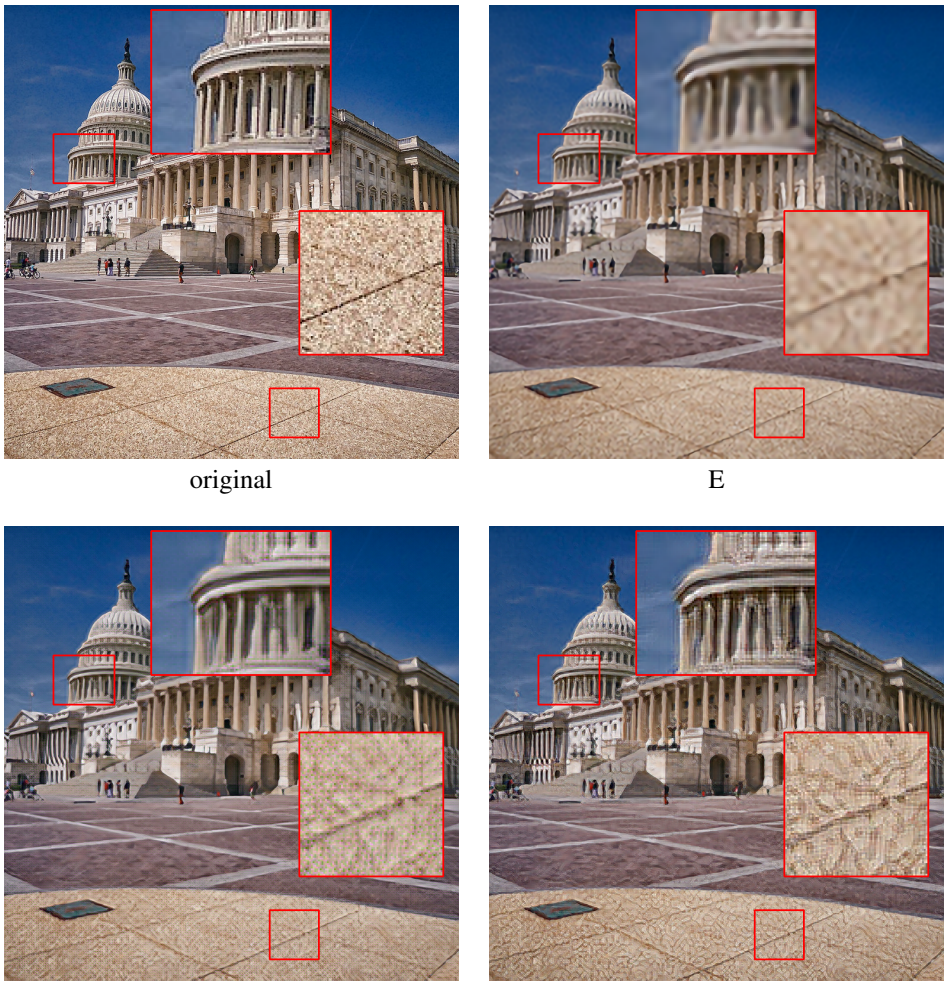


Figure 2: Reconstructions produced by the EnhanceNet for different loss functions (E: MSE, P: Perceptual loss, T: Texture matching loss). The robust perceptual and texture matching loss introduces many artifacts (see highlighted areas). Best viewed digitally.

of a DNN classifier correlates with the image quality perceived by humans. To this end, we plot the classification errors of the robust and non-robust ResNet-50 classifiers in Figure 1. The classification error of the restored images using the EnhanceNet-E is depicted by the red dashed line. We see in Figure 1a that all EnchanceNets trained with a perceptual loss have a lower error than only using the pixel-wise MSE loss. Moreover, the EnhanceNets trained with robust perceptual losses almost always have a smaller error when compared with the non-robust perceptual losses. This trend is inverted when we use a non-robust ResNet-50, as depicted in Figure 1b. In this case, the EnhanceNets trained with a robust perceptual loss always perform worse than the non-robust counterpart, and even worse than the EnhanceNet-E. Therefore, we conclude that the choice of the classifier highly influences the outcome when evaluating the quality of images based on the classification error of a pre-trained DNN.

In the top left corner of Figure 2, we show a crop of the original HD image from the Urban100 dataset [18]. The other images show the reconstructions produced by the EnhanceNet trained with different loss functions. In the top right corner, the reconstruction produced by the EnhanceNet-E (which is trained with the pixel-wise MSE loss) is quite blurry. This is a known problem of the MSE loss [25]. In contrast to that, the reconstruction produced by the EnhanceNet-EPT (bottom left) shows sharper edges and finer details, e.g., the quality of the texture on the ground is superior. In the bottom right, we show the reconstruction produced by the EnhanceNet-EPT trained with the robust

perceptual and texture matching losses. Comparing this reconstruction to that of the EnhanceNet-EPT, we see there are many more artifacts and "echoes" in the reconstructed image. The reconstructions of other EnhanceNet trained with robust perceptual and/or texture losses also introduce these artifacts. More example reconstructions using all considered combinations of loss functions (as seen in Table 2) can be found in the appendix.

4 Discussion

We hypothesized that DNNs robust to adversarial examples could provide perceptual losses that are more aligned with human perception. We expected that these robust perceptual losses, when used to train image reconstruction models, would improve the quality of the reconstructed images.

In this work, we evaluated this hypothesis on the task of image super-resolution. We used an EnhanceNet [4] trained with combinations of different loss functions and compare the effectiveness of robust and non-robust perceptual losses on multiple datasets.

Using a wide range of evaluation metrics (including the novel R-LPIPS metric) we observe that robust perceptual losses do not perform better. Furthermore, we observe that these robust perceptual losses lead to unnatural artifacts in the reconstructed images. Thus, our initial hypothesis was falsified by our extensive experiments. To explain this phenomenon, we now hold the opinion that these robust perceptual losses are not only invariant to changes in the image invisible to humans (such as adversarial perturbations), but are also invariant to more substantial changes recognizable to the human eye. This leads to visible artifacts when training image reconstruction models with these robust perceptual loss functions.

References

[1] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[3] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[4] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4491–4500. IEEE, 2017.

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

[6] 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.

[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations (ICLR) Workshop*, 2017.

[8] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[9] Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. E-LPIPS: robust perceptual image similarity via random transformation ensembles. *arXiv preprint arXiv:1906.03973*, 2019.

[10] Renan A Rojas-Gomez, Raymond A Yeh, Minh N Do, and Anh Nguyen. Inverting adversarially robust networks for image synthesis. *arXiv preprint arXiv:2106.06927*, 2021.

[11] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[14] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 416–423. IEEE, 2001.

[15] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017.

[16] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, pages 711–730. Springer, 2010.

[17] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012.

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206. IEEE, 2015.

[19] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

[20] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Transactions on Image Processing*, 23(2):684–695, 2013.

[21] Hamid R Sheikh, Alan C Bovik, and Gustavo De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on Image Processing*, 14(12):2117–2128, 2005.

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

[23] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595. IEEE, 2018.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[25] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544. IEEE, 2016.

A Appendix: Additional Reconstruction Examples

Target

E

EP

EP (robust)

EPT

EPT (robust)

ET

ET (robust)

P

P (robust)

PT

PT (robust)

Target

E

EP

EP (robust)

EPT

EPT (robust)

ET

ET (robust)

P

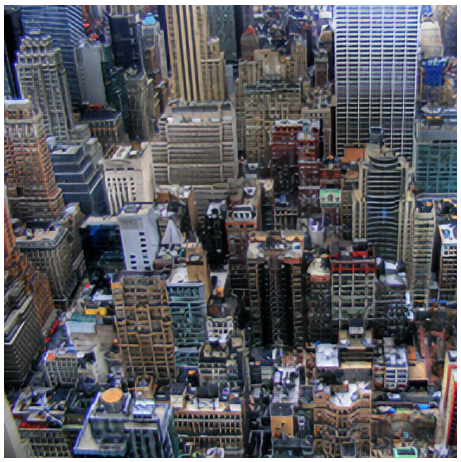
P (robust)

PT

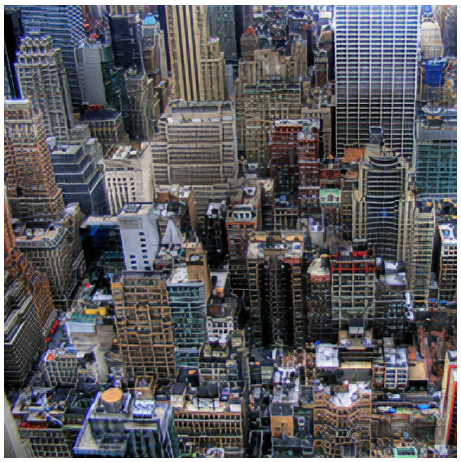
PT (robust)

Target

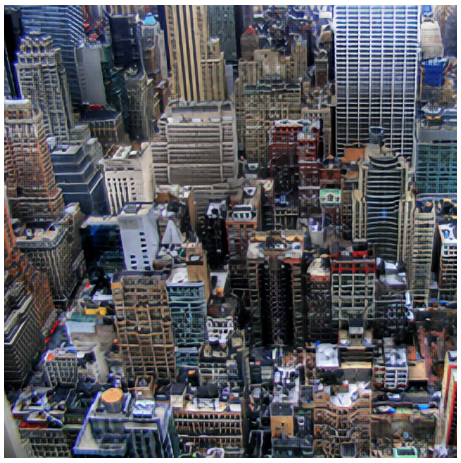
E



EP

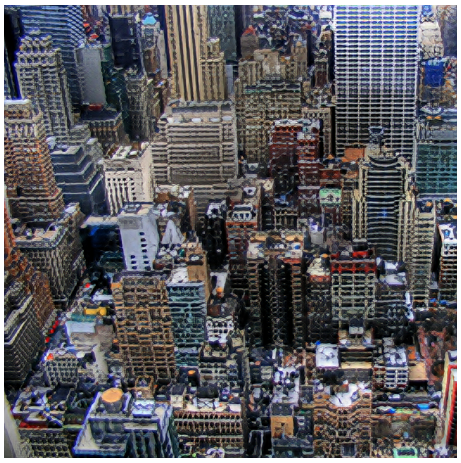


EP (robust)



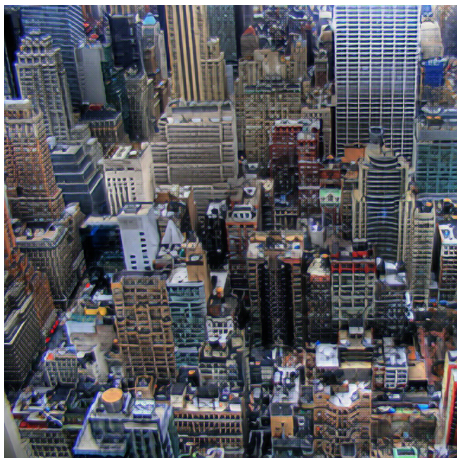
EPT

EPT (robust)

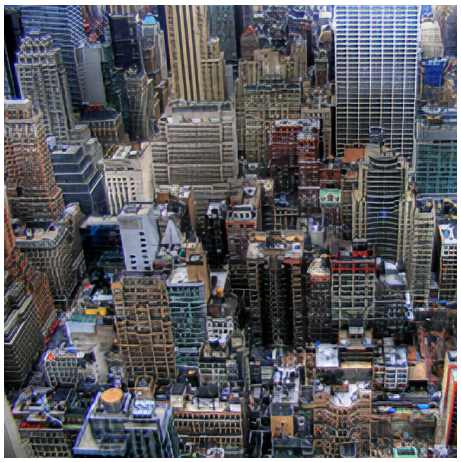


ET

ET (robust)



P



P (robust)

PT

PT (robust)