

# Training a Task-Specific Image Reconstruction Loss

Aamir Mustafa      Aliaksei Mikhailiuk      Dan Andrei Iliescu  
Varun Babbar      Rafał K. Mantiuk  
University of Cambridge, UK

Project website: <https://www.cl.cam.ac.uk/research/rainbow/projects/mdf/>

## Abstract

*The choice of a loss function is an important factor when training neural networks for image restoration problems, such as single image super resolution. The loss function should encourage natural and perceptually pleasing results. A popular choice for a loss is a pre-trained network, such as VGG, which is used as a feature extractor for computing the difference between restored and reference images. However, such an approach has multiple drawbacks: it is computationally expensive, requires regularization and hyperparameter tuning, and involves a large network trained on an unrelated task. Furthermore, it has been observed that there is no single loss function that works best across all applications and across different datasets. In this work, we instead propose to train a set of loss functions that are application specific in nature. Our loss function comprises a series of discriminators that are trained to detect and penalize the presence of application-specific artifacts. We show that a single natural image and corresponding distortions are sufficient to train our feature extractor that outperforms state-of-the-art loss functions in applications like single image super resolution, denoising, and JPEG artifact removal. Finally, we conclude that an effective loss function does not have to be a good predictor of perceived image quality, but instead needs to be specialized in identifying the distortions for a given restoration method.*

## 1. Introduction

The success of deep learning over the past several years has led to extensive use of Convolutional Neural Networks (CNNs) on a wide range of image restoration tasks, such as single-image super resolution or denoising. One of the critical choices effecting CNNs performance is the loss function. A popular mean-squared error (MSE or  $L_2$ ) loss often results in blurry, splotchy [38] or unnatural looking images as the reconstructed image tends to be an average of potential solutions, which may not lie on the natural image manifold [3]. Generative Adversarial Networks (GANs) [11] can ensure that resulting images lie on such a manifold, but

when used alone, may result in images that are substantially different from the input [3]. Furthermore, GANs are challenging to train due to the instability of their optimization problem.

A new category of loss functions, which has recently gained noticeable popularity, employs neural networks as feature extractors. Most commonly, the loss is computed as the  $L_2$  distance between the activations of the hidden layers of a trained image classification network (e.g. a VGG network [28]). Such losses have been successful in training learning-based image restoration models. However, a major drawback of these loss functions is that they use large image classification networks as feature extractors. This not only makes the training process memory intensive, but also focuses on image regions which are more salient for the task of image classification. Recently, Zhang *et al.* [37] tried to overcome this shortcoming by introducing a Learned Perceptual Image Patch Similarity (LPIPS) metric. They calibrated existing pre-trained classification networks on a new dataset of human perceptual similarity judgments. However, this approach still requires an extensive dataset for training the feature extractor. Furthermore, LPIPS/VGG features need to be complemented with an  $L_2$  loss term to offer acceptable performance, which involves the need to carefully tune weights of both loss terms.

In this work, we explore the question of what makes a good loss function for an image restoration task, such as single image super resolution, denoising and JPEG artifact removal. It has been observed that there is no single loss function that works best across different applications [4, 6, 14]. This motivates the need to a novel set of loss functions that do not aim to be universal, but instead are task-specific. In this work, we propose our task-specific Multi-Scale Discriminative Feature (MDF) loss function, which is trained on a single natural image. Despite very lightweight training, our loss outperforms popular feature-wise (perceptual) losses, which have been trained on very large datasets. This is possible, because our loss does not learn the distribution of natural images but instead is trained to penalize the task specific distortions at different image scales. The latter task

is more relevant for a loss function and much easier, thereby can be learned with as little as a single training image. Furthermore, we show that our loss function performs better as a regularization term in an adversarial setting than the VGG loss. An extensive comparison in terms of objective metrics and subjective image quality study shows that our loss function outperforms the state-of-the-art losses for varied image restoration tasks across different datasets.

## 2. Related work

In recent years, the search of an optimal perceptual loss function has gained much attention. Below, we differentiate between hand-crafted losses, which rely on existing metrics, feature-wise losses, where image statistics are extracted using deep learning models, and distribution losses, where the loss pushes the solution to the manifold of natural images.

**Hand-crafted losses:** Zhao *et. al* [38] have studied visual quality of images produced by the image super-resolution, denoising and demosaicing algorithms using  $L_2$ ,  $L_1$ , SSIM [33] and MS-SSIM [34] as loss functions. Images, produced by the algorithms trained with the combination of  $L_1$  and MS-SSIM losses attained the best quality as measured by objective quality metrics. That result was closely followed by the  $L_1$  loss used on its own. Ding *et al.* [6] compared a number of image quality metrics used as a loss function in image reconstruction methods. They found that many of the popular quality metrics do not have properties that could warrant good reconstruction results.

**Feature-wise losses:** Similarity between the reference and the generated image can be computed in the feature space of deep CNNs. This class of losses are often called *perceptual losses* as they are meant to optimize the perceptual quality rather than the pixel differences. However, since these loss functions do not explicitly model perceptual processing, we use a more descriptive name of feature-wise losses.

Authors in [15] used the  $L_2$  norm between the features of the reference and test images extracted from the VGG [28] network as a loss function to train style-transfer and super-resolution algorithms. Here the VGG network was trained on ImageNet dataset [25]. Authors in [37] (LPIPS) have noted that features learned while training the network for image quality assessment task might better capture perceptual similarity between the target and generated image. The work used the features of several networks (untrained VGG, VGG trained on the ImageNet dataset, and on image quality dataset) to predict image quality. The authors observed that hidden representations of all tested deep models encode features important for perceptual similarity. However, deep features at various levels vary in their capacity to model perceived quality. The work of [29] proposed a methodology for selecting deep features of pre-trained CNNs that have the strongest relationship with the perceptual similarity.

However, training image restoration algorithms reliant

only on the features extracted from the deep network as a loss is unstable [3]. Due to pooling in the hidden layers, the network implementing the function is often not bijective, meaning that different inputs to the function may result in identical latent representations [3]. Therefore, feature-wise losses are often used in conjunction with a regularization term, such as  $L_2$  or  $L_1$  norms, and require careful tuning of the weights of each loss component. Delbracio *et. al* [4] proposed a modification to penalize the VGG features of the reference and the predicted image based on the 1D-Wasserstein distance [31, 24]. However, this method again relies on  $L_1$  normalization for training to achieve acceptable results.

**Distribution loss (GAN):** Many image restoration algorithms are inherently ill-posed. For example images produced by super-resolution or denoising algorithms can have acceptable perceptual quality while not precisely matching the ground-truth. These algorithms can be optimized to produce images that lie on the natural image manifold, constrained by the similarity to the ground truth distribution. To ensure that the first requirement is met, many works have relied on GANs [11]. In such a setting, the image-generation algorithm has two loss terms: the discriminator, trained to differentiate between the generated and natural images, and a term constraining the generator network to produce images close to the ground truth. In [35, 13] authors used  $L_1$  norm to regularize the training. Similarly the works of [8, 16] used the feature-wise VGG-based loss to constraint the generator. Some other works combined both hand-crafted and feature-wise losses [26, 32]. Others have introduced regularization based on the feature loss of the discriminator [30, 14]. To avoid regularization in training for SISR the work of [2] proposed to use the consistency enforcing module. The module can wrap any SISR architecture, making it satisfy the consistency constraint – a down-sampled version of the image reconstructed with the network must be close to the low-resolution input.

Inability of the losses to generalize over different image restoration applications and over varied datasets raises the need of a task-specific loss function. In the following section, we introduce our proposed loss function which is trained to identify and thereby successfully remove the distortions for a given image restoration task in hand.

## 3. Multi-Scale Discriminative Feature Loss

Feature-wise (perceptual) loss functions, such as a pre-trained VGG-Net is commonly used as a feature extractor when training image restoration models. Additionally, adversarial loss is often used as a regularizer to push the solution to the natural image manifold using a discriminator network that is trained to differentiate between distorted and the natural images [16]. However, a fundamental weakness of these methods is that they aim at learning the distribution

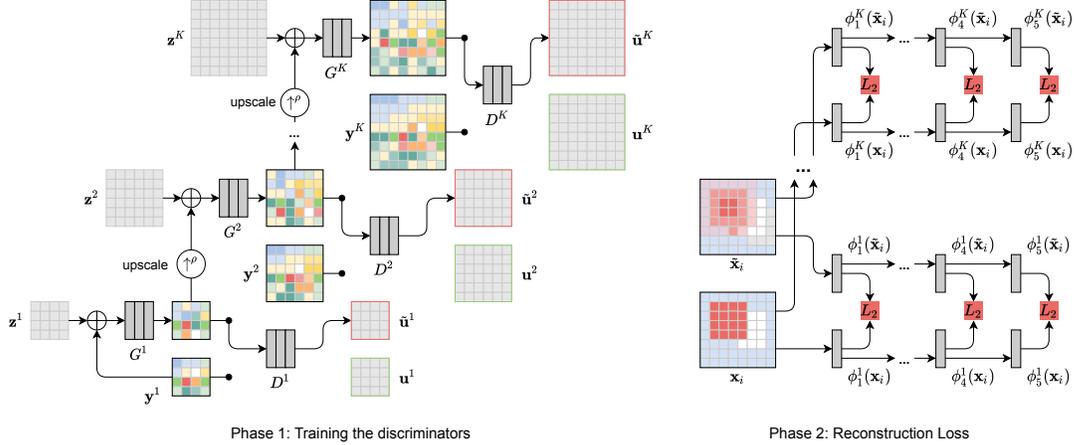


Figure 1: Graphical illustration of the two phases of our loss. **Phase 1** denotes the adversarial training of the discriminators. The generated image is produced by the scale-specific generator  $G^k$ , which takes as input the upscaled output of the previous level  $\tilde{\mathbf{y}}^{k-1} \uparrow^\rho$  added with the task specific distortions  $\mathbf{z}^k$ . For SISR, no distortions are added ( $\mathbf{z}^k = 0$ ). The levels are sequentially trained from the coarsest scale to the finest. In **Phase 2**, the discriminators are frozen and used as feature extractors over whose outputs an  $L_2$  distance is measured between the ground-truth training image  $\mathbf{x}_i$  and the restoration output  $\hat{\mathbf{x}}_i$ . The distance is measured between the two images at every scale  $k$  and intermediate layers of the discriminator  $\|\phi_l^k(\mathbf{x}_i) - \phi_l^k(\hat{\mathbf{x}}_i)\|_2^2$ .

of natural images using large training datasets, which is less relevant for a loss function. In this paper, we introduce our **Multi-Scale Discriminative Feature (MDF)** loss, which, instead is trained to penalize the task specific distortions that are introduced iteratively to the generator at various stages of training, making the trained discriminator specialized in identifying the distortions for a given restoration method. Unlike VGG and LPIPS networks, which were trained for image classification and the prediction of image quality, respectively, our feature extractor networks are trained for the task that is directly relevant to restoration task in hand. Furthermore, this task is much easier than learning the entire distribution of natural images and thereby can be achieved with as little as a single training image.

The foundations of our loss function are based on the following propositions:

**Proposition 1:** *Networks employed as feature extractors for the loss should be trained to be sensitive to the restoration error of the input image. This makes the feature space more suitable for penalizing the distortions during training for that specific task.*

**Proposition 2:** *Learning natural-image manifold, which is the task often attributed to discriminators, is a much harder task and is less relevant for the feature-wise loss function. The loss function should be able to detect relevant distortions regardless of image content, i.e. be content invariant.*

To validate both propositions, we design a new feature-wise loss. The feature-space comprises the intermediate activations of the set of discriminator networks trained as a single-image GAN [27] specialized in removing task-specific distortions from a *seed image*. We denote the seed

image by  $\mathbf{y}$  to differentiate it from the *training images*, denoted by  $\{\mathbf{x}_i\}_{i=1}^N$ , which are used for learning the restoration task. The proposed loss function is trained in a multi-scale manner so that it is sensitive to the relevant distortions at multiple scales. The seed image can have a different size from the training images, can depict a different type of a scene, or can be a synthetic image. Below we revisit the training procedure for our multi-scale discriminators and operation of MDF loss on image restoration tasks.

### 3.1. Phase 1: Training the discriminators

We use the architecture of SinGAN [27] to train the multi-scale discriminators on a single seed image in a task specific manner. A set of generators  $\{G^k\}_{k=1}^K$  and a set of discriminators  $\{D^k\}_{k=1}^K$  are instantiated for a pre-defined set of  $K$  scales. Conventionally, scale 1 is the coarsest level and scale  $K$  is the finest (the original image). In our experiments, we chose  $K = 8$ , resulting in 8 sets of discriminators in the MDF loss. The seed image at scale  $k$ ,  $\mathbf{y}^k$ , is obtained by downsampling (using Lanczos filter) the original image by a factor of  $\rho^{(K-k)}$ , where  $\rho = 2$ .

For each scale of training, the generator takes as input the upscaled output of the lower scale after adding the task specific artifacts to it:  $\tilde{\mathbf{y}}^k = G^k(\tilde{\mathbf{y}}^{k-1} \uparrow^\rho + \mathbf{z}^k)$ . Here  $\mathbf{z}^k$  is the distortion added to the upscaled output of the lower scale  $\tilde{\mathbf{y}}^{k-1} \uparrow^\rho$ . For the first scale of training, the input to the generator is the original image downsampled by a factor of  $\rho^{(K-1)}$  and then distorted by the task-specific error. We have experimented with the generator that was taken directly from SISR network but we did not observe a substantial improvement in performance. Phase 1 of Fig. 1

provides a graphical illustration of the training scheme.

In contrast to the protocol used for training a single image GAN [27], we use application-specific distortion  $\mathbf{z}^k$  while training. For image denoising,  $\mathbf{z}^k$  is Gaussian Noise of a magnitude randomly sampled from a uniform distribution of [0,55] on a pixel scale of (0,255). For JPEG artifact removal, the upscaled output from the previous scale is compressed with a JPEG quality chosen randomly between 7 and 10, before being fed to the finer scale. However, for the task of Single-Image Super Resolution (SISR), no distortion is added ( $\mathbf{z}^k = 0$ ) and the upscaled output from the previous scale is directly fed to the next level.

The corresponding discriminator at scale  $k$  takes as input the generated image  $\tilde{\mathbf{y}}^k$  and produces a map of [0, 1] activations  $\tilde{\mathbf{u}}^k = D^k(\tilde{\mathbf{y}}^k)$  with the same dimensionality as  $\tilde{\mathbf{y}}^k$ . Alternatively, the discriminator can be supplied with the downscaled seed image  $\mathbf{y}^k$ , resulting in the activation map  $\mathbf{u}^k$ . The discriminator is trained to distinguish patches of the seed image from patches of the generated image and, therefore, the activations of  $\mathbf{u}^k$  are pushed towards 1 and those of  $\tilde{\mathbf{u}}^k$  are pushed towards 0. The number of such activations in the map depends on the number of convolutional layers in the discriminator and their kernel size. In our case, each activation corresponds to an  $11 \times 11$  patch in the input. Training is done sequentially across scales. The coarsest scale is trained for 3000 iterations, then the weights are frozen and the next scale is trained, and so on. The training loss for the  $k$ -th GAN is comprised of an adversarial term and a reconstruction term:

$$\max_{G^k} \min_{D^k} \mathcal{L}_{adv}(G^k, D^k) + \alpha \mathcal{L}_{rec}(G^k) \quad (1)$$

The reconstruction loss  $\mathcal{L}_{rec}$  employed is the MSE loss between the generated  $\tilde{\mathbf{y}}^k$  and the ground truth image  $\mathbf{y}^k$  to ensure faithful generation of the output image. The reconstruction loss weight  $\alpha$  is set at 100. Selection of the loss function and the hyper-parameters are based on [27].

It must be noted that addition of the above distortions (Gaussian Noise and JPEG compression artifacts) to the *seed image* at various scales makes the discriminator sensitive to such artifacts but agnostic to the image content. The main benefit of our discriminative loss function is that it does not require thousands of images to be trained on, instead a single natural image and knowledge of the distortions are sufficient to provide state-of-the-art results.

### 3.2. Phase 2: Training for image restoration

In this phase, the trained discriminators are used as the loss function for an image restoration task. For all restoration tasks we use latent embeddings after every ReLU layer of the trained discriminators as features. We denote the embeddings by  $\phi_l^k(\mathbf{x})$  meaning the output of the  $l$ -th layer of the discriminator for the  $k$ -th scale. The output of the whole

Table 1: Comparison of the properties of our proposed loss against other competing losses.

Loss function	Training overhead	Memory overhead	Multi-scale	Inference GPU (ms)	Backpropagation time (ms)	Regularization
$L_2$	None	None	No	1.2	1.0	-
$L_1$	None	None	No	1.2	1.0	-
SSIM [33]	None	None	No	12	1.6	-
MS-SSIM [34]	None	None	Yes	24	6.5	-
VGG [15]	1.3M images	58.9MB	No	27	21.8	Required
LPIPS [37]	161k images <sup>1</sup>	9.1MB	No	31	17.5	Required
MS-SSIM + $L_1$ [38]	None	None	Yes	25	8.2	-
Ours	One image	4.2MB	No <sup>2</sup>	11	4.0	-

discriminator is then  $\phi_L^k(\mathbf{x})$ , where  $L$  is the total number of layers. If  $\mathbf{x}_i$  is the ground-truth for the  $i$ -th training image and  $\tilde{\mathbf{x}}$  is its reconstruction, then our MDF loss is:

$$\mathcal{L} = \sum_{k=1}^K \sum_{l=1}^L \|\phi_l^k(\mathbf{x}) - \phi_l^k(\tilde{\mathbf{x}})\|_2^2 \quad (2)$$

A subtle but crucial aspect of our loss is that *the discriminators are not applied to the scales on which they were trained*. If the seed image has dimensions  $H_y \times W_y$ , the training input (both seed and synthetic) to the discriminator  $D^k$  will have dimensions  $H_y/\rho^{(K-k)} \times W_y/\rho^{(K-k)}$ . However, the input to the discriminator during phase 2 of training will not be scaled and it will be  $H_i \times W_i$ , the size of the  $\mathbf{x}_i$ .

## 4. Comparison of loss functions

In this section, we evaluate the efficacy of our MDF loss on a variety of image restoration tasks that rely on CNN architectures and also as a regularization term in an adversarial training (Sec. 4.4). We compare our loss with the most widely used loss functions, listed in Table 1, including the perceptual loss [10, 15]. In all cases, we train the models on the training portion of the DIV2K dataset [1] and use for testing DIV2K (the validation set), Berkeley Segmentation Data (BSD 500) [19] and real world mobile phone captured images from the DPED dataset [12]. The best model is selected based on the validation loss.

Note that both VGG and LPIPS losses must be combined with the  $L_2$  loss to produce acceptable results. For fair comparison, we conducted a hyper-parameter search over the scalar  $\lambda$  controlling the weight of the feature-wise loss function. We searched over the values in  $\{\lambda : \lambda = 10^k, k = -3, \dots, 3\}$  for super-resolution and the values of 0.01 and 1 for other applications, due to computational cost. The results of these experiments can be found in the appendix. In our experiments across all restoration applications, we found the best results are produced when  $\lambda = 1$  for VGG and  $\lambda = 0.1$  for LPIPS loss. Note that unlike VGG and LPIPS, our MDF loss function does not require addition of  $L_2$  regularization while training. It is also worth noting that our MDF loss function is less computationally expensive

<sup>1</sup>This is training on top of a pre-trained network using 1.3M images

<sup>2</sup>Only training of discriminators is performed in a multi-scale fashion.

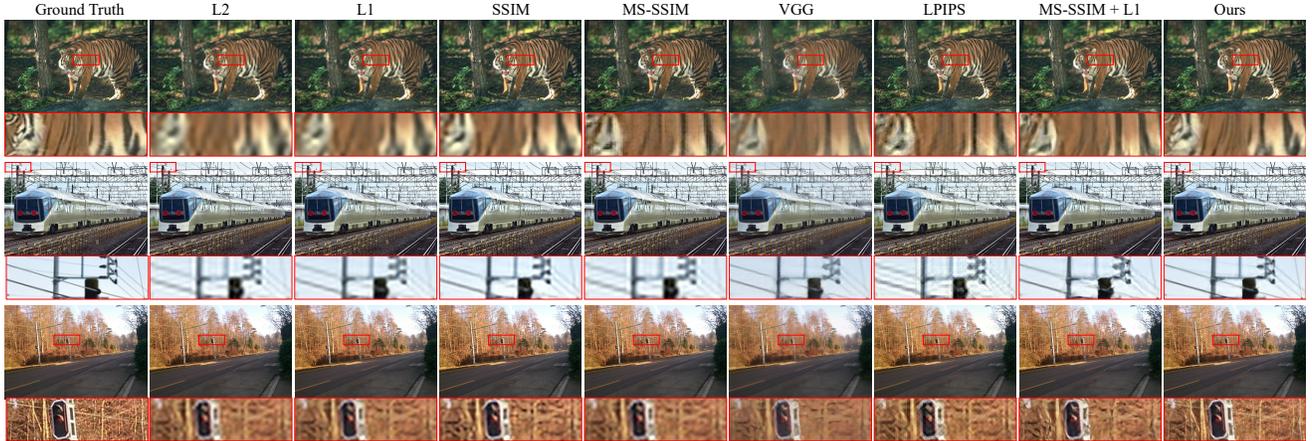


Figure 2: SISR results for EDSR [18] trained using different loss functions. Top row shows a sample image from BSD [19], second row from the DIV2K validation [1] and the bottom row from DPED dataset [12]. The results for our loss are sharper and have fewer artifacts across all datasets. Best viewed when zoomed. Additional results are provided in the SM.



Figure 3: Results for denoising using DnCNN model [36] trained using different losses. Top row shows a sample image from BSD [19] and second row from the DIV2K validation [1]. Our loss improves noise reduction, especially in the uniform areas of an image. Best viewed when zoomed. Additional results are provided in the SM.

and has a much lower memory overhead compared to VGG and LPIPS (refer to Table 1).

**Single-image super resolution** Here, we evaluate our proposed loss for the task of SISR, which aims at estimating a High-Resolution (HR) image from a given Low-Resolution (LR) image. For SISR, we use two state-of-the-art architectures, namely Enhanced Deep Super-Resolution (EDSR) [18] and SR-ResNet [16]. The LR image is generated by downsampling the original HR image by a factor of 4 using bicubic filter. For training, we randomly extract  $96 \times 96$  patches from the dataset and perform data augmentation with  $90^\circ$ ,  $180^\circ$  and  $270^\circ$  rotations, and horizontal and vertical flips. Each model is trained for 500 epochs with an initial learning rate of 0.001 with gradual rate scheduling.

**Image denoising** We train the DnCNN architecture proposed by Zhang *et al.* [36]. The training set is generated by adding Gaussian noise with the standard deviation randomly selected from the uniform distribution of  $[0, 55]$ . We

use SGD with a weight decay of 0.0001 with Nesterov momentum optimizer for training. Each model is trained for 50 epochs with an exponential learning rate scheduling from 0.1 to  $10^{-4}$  with the momentum parameter set to 0.9.

**JPEG artifact removal** For this application, we use the same DnCNN [36] as for the denoising. During training we feed in images compressed with the JPEG codec with the quality factor of 10 as in [7, 9]. We perform data augmentation with  $90^\circ$  image rotation, vertical and horizontal flips. The model is trained with Adam optimizer and the learning rate set to  $1e-4$ . The test images are compressed with a quality factor of 10 and a more challenging factor of 7.

#### 4.1. Qualitative results

In Figs. 2 and 3 we provide qualitative results for SISR and image denoising respectively. The examples for other applications can be found in the appendix. Furthermore, we include an extensive set of results at the original resolution in a separate HTML report. The visual results consistently

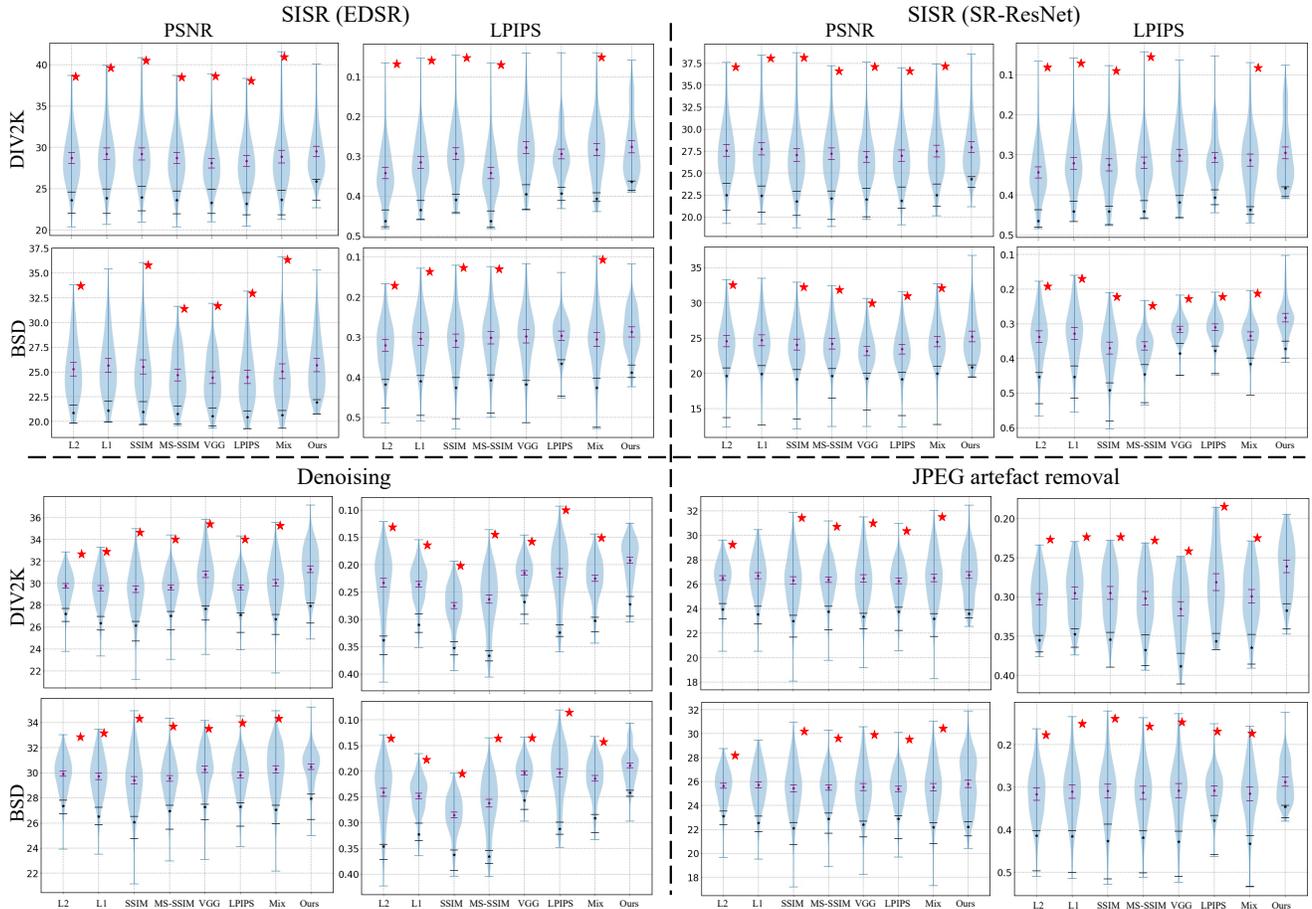


Figure 4: Violin plots illustrating the distribution of the PSNR [dB]  $\uparrow$  and LPIPS  $\downarrow$  values across different losses across all applications for two datasets. Note that the y-axis is reversed for LPIPS so that the quality improves towards the top of each plot. The error bars show the 95% confidence intervals for the mean (magenta) and the 5<sup>th</sup> percentile (black). The latter CIs were computed by bootstrapping. The red asterisks indicate that one-tailed t-test on the means gives statistically significant difference at  $\alpha = 0.05$ . It is worth noting that our loss produced fewer images with low quality values.

indicate that our task-specific loss can produce sharper, less noisy images with fewer artifacts. The differences are the most noticeable in the flat areas of the images.

## 4.2. Quantitative results

The quantitative results for all four applications are shown as distributions in Fig. 4 for two test datasets: DIV2K and BSD. We report the quantitative results in tabular form and provide additional results for DPED dataset in the appendix. The differences in means (magenta dots in Fig. 4) are small but statistically significant for most comparisons (one-tailed t-test with  $H_1$  show that the quality score is higher for our method, red \* symbols are shown if the difference is significant at  $\alpha = 0.05$ ). The means, however, are not the best indicator of performance of different losses. This is because the differences in loss functions are mostly visible in smooth or flat parts of the images, which occupy only small percentage of all pixels but

have a substantial impact on the perceived image quality (as demonstrated in Sec. 4.3). The advantage of our loss is better visible for the worst-case results, shown in Fig. 4 as the lower 5<sup>th</sup> percentile of values (black asterisks). In majority of the comparison, MDF loss produces fewer images with low quality values, especially in terms of LPIPS.

## 4.3. Subjective quality assessment

Objective metrics such as PSNR or LPIPS, can be unreliable in predicting the perceptual quality of images. They also do not capture the practical significance of the perceptual difference; we do not know whether the improvement of 0.5 dB is going to be appreciated by an average observer. For that reason, we ran perceptual experiments on the Amazon Mechanical Turk crowd-sourcing platform.

For best sensitivity of the test, we used full-design pairwise-comparison protocol [23]. In each trial, participants were presented with 3 side-by-side images: one refer-

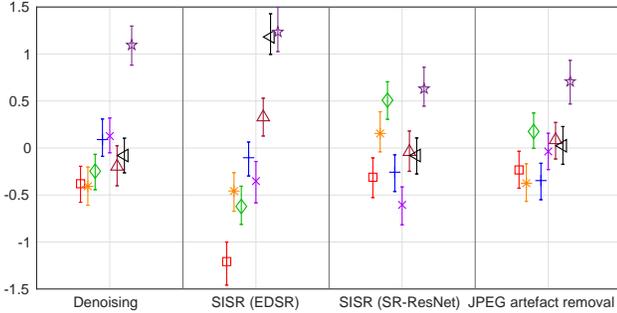


Figure 5: Subjective experiment in JND units (the higher, the better). Error bars denote 95% confidence intervals. The legend is same as Fig. 6.

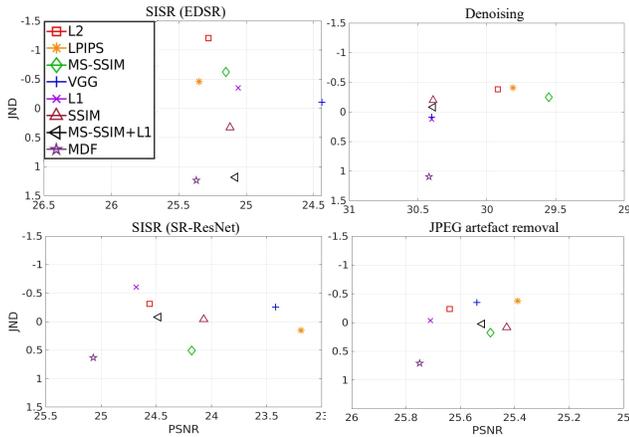


Figure 6: Perception-distortion trade-off for the tested losses. The axes have been reversed so that the lowest distortion is shown on left and the highest perceptual quality at the bottom as in [3].

ence and two generated by the image reconstruction methods, each with different loss function from the BSD dataset. Participants were asked to select the image that appeared closer to the reference. For fair comparison, we randomly selected 50 images from testset. Thus, every loss function was compared to every other loss 50 times. Overall, we collected 1400 comparisons for each restoration method.

In each Human Intelligence Task (HIT) we included nine pairwise comparison trials and one (for denoising and JPEG artefact removal) or two (EDSR and SR-ResNet) additional pairwise comparisons with an obvious outcome to screen the results against the participants who misunderstood the task. If a participant made a mistake in those comparisons, we excluded that HIT. Overall we discarded 4.2% and 14.2% comparisons for SISR with EDSR and SR-ResNet respectively, 7.1% comparisons for denoising and 9.2% comparisons for JPEG artefact removal.

For each application we aggregated collected comparisons and performed Just Noticeable Difference (JND) (Thurstonian) scaling on the results using the method from [23]. The results express the quality difference in JND units. One JND unit means that 75% of the population will select

one method over another (from a pair). The results of the scaling, plotted in Fig. 5, show consistent improvement of our method over other losses. MS-SSIM +  $L_1$  performed the second best for SISR on the EDSR, with MDF having an advantage of 0.05 JND. For other applications, MDF shows a substantial improvement over all competing losses.

To gain further insights, in Fig. 6 we visualize the results as the perception-distortion trade-off [3], which shows the distortion (PSNR) on the x-axis and the JND quality values on the y-axis (reversed scale). The results across all applications clearly show that the proposed MDF loss results in both the lowest distortion and the highest perceived quality. The results for EDSR show drastic difference in the performance as measured by PSNR and subjective experiment. MDF and  $L_2$  – the best and the worst performing losses, differ only by 0.09 PSNR, but have 2.4 JND difference in the perceptual quality, corresponding to 94.7% of the population selecting the results produced by MDF.

#### 4.4. Comparison with adversarial loss

Our MDF loss can be also used as a reconstruction term when training a GAN architecture for image restoration. For this experiment, we chose the task of SISR and used state-of-the-art GAN based method — ESRGAN [32]. The model trained with the MDF loss function alongside the adversarial loss achieves a PSNR of 25.37 dB as compared to the weighted combination of VGG and MSE loss function’s 25.06 dB. Both the models are trained using the DIV2K dataset and inference is run on the BSD dataset. Since, models trained with adversarial loss are known to produce lower PSNR values, we further conducted a subjective study to predict the perceptual quality of the images. We ran a pairwise comparison study on 50 randomly selected images from the testing dataset with each pairwise comparison performed four times. MDF has an advantage over VGG loss function and was selected in 58% of the comparisons.

#### 5. Ablation analysis

The ablation studies test the importance of task-specific MDF, the choice of seed image, the number of images used to train the discriminator and the number of discriminator scales. The latter two studies are described in the appendix.

**Task-specific distortions** Here we test whether the loss trained on one task can serve as a feature extractor for another. We train DnCNN for JPEG artefact removal using MDF with introduced either noise or JPEG distortions (different vectors  $\mathbf{z}^k$ ). The task-specific discriminator gained moderate performance increase in terms of PSNR (25.75 dB for MDF JPEG and 25.61 dB for MFD noise) but resulted in images of much better visual quality, as shown in Fig. 8.

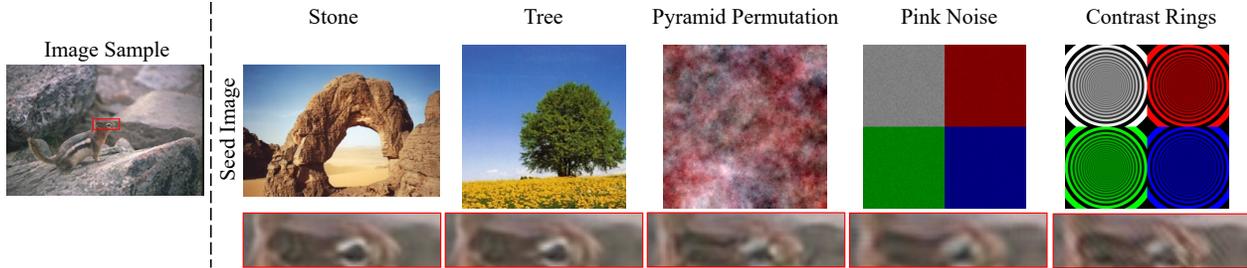


Figure 7: Ablation study on changing the image used for training our MDF loss function. It can be seen that natural images provide visually better results as compared to synthetic images. Best viewed when zoomed.



Figure 8: Example results for JPEG artifact removal when trained on task-specific MDF JPEG and MDF noise. Task-specific MDF results in improved visual quality.

**Seed image** We study the effect of using different natural and synthetic images for training our MDF loss function. Fig. 7 shows 5 seed images including 2 natural and 3 synthetic ones that were used to train the discriminators. *Pyramid Permutation* image has been created by a random permutation of pixel order on each level of the Laplacian pyramid. Such permutation distorts image second-order statistics, but preserves the composition of the spatial spectrum. *Pink Noise* image contains  $1/f^2$  noise that is typical for natural images. *Contrast Rings* image contains concentric rings whose contrast is reduced towards the centre to cover the range of edges of all orientations and contrast magnitudes. The results of SISR (EDSR), shown in the bottom part of Fig. 7, indicate that the visual quality of the super-resolved images is best for natural images and is degraded as the statistics of the training image is distorted. However, from the results for all the applications, the visual quality of the restored images is more dependent on the nature of distortions added ( $z^k$ ) than the choice of the seed image.

## 6. Image quality metrics and loss functions

Provided with the results of our subjective quality experiment from the previous section, we further test whether a good loss function is also a good image quality metric. Here, we used each loss function as a quality predictor for the improved version of the TID2013 dataset [21]. The dataset is one of the most accurate (due to large number of comparisons), is scaled in JND units, and contains suffi-

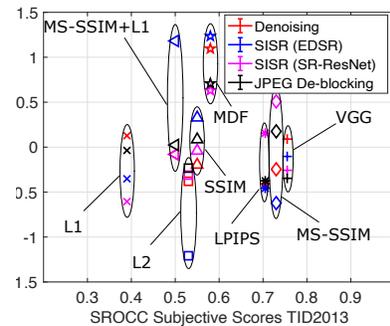


Figure 9: Performance of loss functions on the task of image quality prediction versus performance as objective functions. Results do not show strong correlation. Markers are consistent with Fig. 5

ciently large number of conditions (over 4000 images). In Fig. 9 we plot the Spearman Rank Order Correlation Coefficient (SROCC) with the subjective scores from the improved TID2013 against the JND values from our subjective experiment. High SROCC value indicate that the loss is a good predictor of image quality. The scatter plot shows little correlation; the best quality predictors are not necessarily the best loss functions. This is an important finding because it puts in question whether loss functions should be optimized for prediction of image quality. Additional experiments to investigate the performance of various loss functions as quality predictors are provided in the appendix.

## 7. Conclusions

In this paper, we have shown several observations that go against the common assumptions of what make a good loss. We demonstrated that a small multi-scale discriminator network, trained to detect application-specific distortions, can serve as a better feature-wise loss than large networks, such as VGG, which have been trained on large datasets. This shows that learning a natural image manifold, semantic, or style features may not be essential for an effective loss function. Instead, the loss needs to penalize errors specific for restoration task in hand. Our subjective assessments reveal that the restored images generated using models trained with a task specific loss function are consistently chosen by human observers to be closer to the reference images.

## Appendix

This appendix includes additional details that could not be included in the main paper due to the lack of space. This comprises: *a)* manifold assumption validation and visual comparison with SR-GAN discriminator as compared to our multi-scale discriminators; *b)* quantitative results in terms of average PSNR and LPIPS for all application across 3 benchmark datasets *c)* qualitative results for the JPEG artefact removal application; *d)* ablation study on the number of seed images and the number of discriminators of the Multi-Scale Discriminative Feature (MDF) loss; *e)* hyperparameter tuning for the VGG and LPIPS feature-wise loss functions; and *f)* performance of loss functions as quality predictors.

### 1. Image manifold assumption

The main objective of GANs [11] in image restoration is to learn a discriminator model that differentiates between image manifolds [17, 35, 22, 20, 5]. This is based on the hypothesis that input samples (e.g. noisy images) and their corresponding ground truth samples lie on two different manifolds. The generator model thereby learns a mapping function from one manifold to another, resulting in photo-realistic images closer to the natural image manifold [16, 5].

However, in this paper, we propose that learning the natural image manifold, which is often the task attributed to the discriminator, is less important than being able to detect errors introduced by the generator. Moreover, learning the natural image manifold requires the GAN to be trained with thousands of natural and fake images, making the training process computationally intensive. Here, we show that our task-specific discriminators, trained on a single image, can be used as feature extractors for the loss function because they learn the generator errors rather than the natural image manifold.

To validate this claim, a multi-scale discriminator trained on *a single image* for the task of JPEG artefact removal is employed as feature extractor. We randomly sample 100 natural images from the ILSVRC validation dataset [25]. From these images we generate *a)* JPEG compressed images using a compression quality between 7 and 10, *b)* blurry image samples by downsampling and upsampling the images by a factor of 4 using bi-linear filter and *c)* scrambled images by randomly permuting the pixels on each level of the Laplacian pyramid. Such permutations distort the second-order statistic, but preserve the composition of the spatial spectrum. The JPEG trained discriminator is used to extract the latent feature space of each set of images. The feature space for each image is the average across the channels and the resulting feature vector is reduced to a dimensionality of 3 using t-SNE for visualization. Fig. 10 shows the plot of the features from each set of images. The visual-

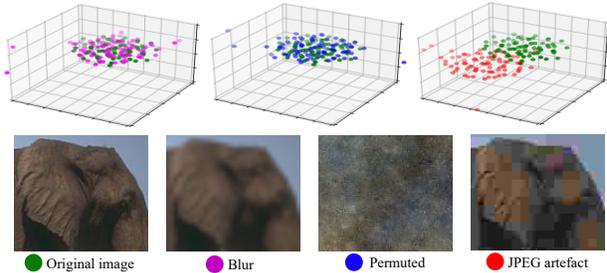


Figure 10: Manifold assumption validation: The figure shows the 3D t-SNE plots of the latent feature vectors extracted from diverse sets of images using multi-scale discriminators trained for the JPEG artefact removal task. Our JPEG-tuned discriminator cannot differentiate between the original and permuted images (middle plot), yet is a very effective feature-extractor for a loss function for JPEG task.

ization shows that the discriminator does not learn the natural image manifold and cannot discriminate between natural and randomly permuted images. It also cannot discriminate between blurred and original images, but performs well in detecting JPEG artifacts regardless of image content.

#### 1.1. Image manifold comparison

In this section, we repeat the experiment conducted above, instead this time for a fully trained SR-GAN [16] discriminator. This further bolsters our claim that the task-specific discriminators of our MDF loss function learn to detect the generator distortions instead of the entire natural image manifold. This thereby allows our MDF loss function, trained on a single image, to be used to effective feature extractors between the generated and the reference image.

We chose the same sample of 100 natural images from the ILSVRC validation dataset [25]. From these images we generated *a)* JPEG compressed images using a compression quality between 7 and 10, *b)* blurry image samples by downsampling and upsampling the images by a factor of 4 using bi-linear filter and *c)* scrambled images by randomly permuting the pixels on each level of the Laplacian pyramid. Such permutations distort the second-order statistic, but preserve the composition of the spatial spectrum. A trained SR-GAN discriminator is used to extract the latent feature space of each set of images. The feature space for each image is chosen after the Global Average Pooling (GAP) layer of the network. We used t-SNE to reduce the dimensionality of the feature vector to 3 for visualization. Fig. 11 shows the plot of the features from each set of images. The visualization shows that the discriminator of SR-GAN learns the natural image manifold (unlike our multi-scale discriminator) and can discriminate between natural and randomly permuted images. However, it cannot dis-

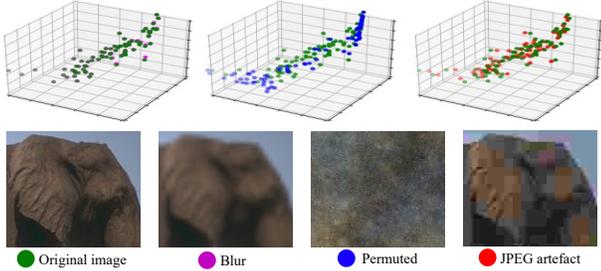


Figure 11: Manifold assumption validation: The figure shows the 3D t-SNE plots of the latent feature vectors extracted from diverse sets of images using an SR-GAN discriminator trained on DIV2K dataset [1]. The SR-GAN discriminator cannot differentiate between the original and jpeg images (right plot), thereby cannot be used as an effective feature extractor to detect and remove distortions.

criminate between the JPEG compressed and original images, making it an inferior feature extractor to detect and remove distortions.

## 2. Quantitative results

The quantitative results for all four applications are shown as distributions in Fig. 12 for real world mobile phone captured images from DPED dataset [12]. The differences in means (magenta dots in Fig. 12) are small but statistically significant for most comparisons (one-tailed t-test with  $H_1$  show that the quality score is higher for our method, red \* symbols are shown if the difference is significant at  $\alpha = 0.05$ ). The means, however, are not the best indicator of performance of different losses. This is because the differences in loss functions are mostly visible in smooth or flat parts of the images, which occupy only small percentage of all pixels but have a substantial impact on the perceived image quality (as demonstrated in Sec. 4.3 of the main paper). The advantage of our loss is better visible for the worst-case results, shown in Fig. 12 as the lower 5<sup>th</sup> percentile of values (black asterisks). In majority of the comparison, MDF loss produces fewer images with low quality values, especially in terms of LPIPS. We also report the quantitative results in terms of average PSNR and LPIPS in Table. 3.

## 3. JPEG artefact removal results

In this section, we provide qualitative results showing comparison between three sample reconstructed images from the BSD Test Set using our (MDF) loss with various other loss functions for the task of JPEG artefact removal application. The test images are compressed with a quality factor of 10 and a more challenging factor of 7. Fig. 14 shows the results for the compression quality factor 7. The

Table 2: Ablation study on training the SISR model (EDSR) using different scales of our loss. The scale number represents the number of scales included in the MDF loss. The inference results are reported for the BSD dataset.

Scales	1	2	3	5	7	8
PSNR $\uparrow$	22.55	23.89	24.43	24.89	25.27	<b>25.70</b>
LPIPS $\downarrow$	0.392	0.357	0.354	0.311	0.305	<b>0.286</b>

performance of the various loss functions seems to be comparable for the quality factor of 10, however, our model substantially provides artefact removal, especially in the uniform areas of the image for a much challenging codec quality of 7. The same was also observed in the subjective experiment conducted (see Sec. 4.3 of the main paper).

## 4. Ablation study

### 4.1. Scales of Discriminators

Since our MDF loss function comprises a series of discriminators trained on a single image at various scales, we need to select the optimal number of scales (the hyper-parameter  $K$  in Equation 2 of the main paper) to achieve the best performance. We perform an ablation study on training the EDSR model [18] using only the coarsest scale discriminator and subsequently adding finer scales. We observe a significant increase in quality of the images generated with the increase in the number of discriminators. As shown in Table 2, our loss performs the best when all 8 scales are employed.

**Number of seed images** Next we investigate the impact of increasing the number of seed images while training the MDF loss function. The plot in Fig. 13 shows that the performance of EDSR increases by only 0.03 dB when trained on 4 images and then it saturates. We did not observe any improvement in visual quality. Because the increase in performance is negligible when adding more seed images, we used a single image for training in our results.

## 5. Hyper-parameter tuning for VGG and LPIPS

In Fig. 15 we show the qualitative results for the trade-off between the MSE and LPIPS/VGG network components in the joint loss function. For fair comparison, we conducted a hyper-parameter search over the scalar  $\lambda$  controlling the weight of the feature-wise loss function. We searched over the values in  $\{\lambda : \lambda = 10^k, k = -3, \dots, 3\}$ . The greater  $\lambda$  parameter is, the more LPIPS/VGG components contribution is. In our experiments across all image restoration applications, we found the best results are produced when  $\lambda = 1$  for VGG and  $\lambda = 0.1$  for LPIPS loss.

Table 3: Comparison of our proposed Multi-Scale Discriminative Feature (MDF) loss function with other losses on 3 public benchmark datasets for four tested applications. Results show PSNR [dB]  $\uparrow$  / LPIPS  $\downarrow$ . The numbers in red indicate the best performance and the ones in blue the second best.

Dataset	L <sub>2</sub>	L <sub>1</sub>	SSIM	MS-SSIM	VGG	LPIPS	MS-SSIM + L <sub>1</sub>	Ours
Single Image Super-Resolution (EDSR [18])								
DIV2K	28.70 / 0.342	<b>29.22</b> / 0.315	29.21 / 0.293	28.70 / 0.342	28.10 / <b>0.278</b>	28.34 / 0.283	28.87 / 0.283	<b>29.51 / 0.276</b>
DPED	26.99 / 0.415	<b>27.26</b> / 0.394	27.22 / 0.369	27.00 / 0.367	26.54 / <b>0.361</b>	26.76 / 0.366	26.88 / 0.368	<b>27.48 / 0.351</b>
BSD	25.28 / 0.320	<b>25.66</b> / 0.304	25.52 / 0.309	24.70 / 0.301	24.44 / 0.298	24.49 / <b>0.296</b>	25.08 / 0.306	<b>25.70 / 0.286</b>
Single Image Super-Resolution (SR-ResNet [16])								
DIV2K	27.57 / 0.343	<b>27.76</b> / 0.321	27.05 / 0.325	27.20 / 0.320	26.83 / <b>0.301</b>	27.00 / 0.307	27.49 / 0.313	<b>27.95 / 0.295</b>
DPED	27.03 / 0.428	<b>27.41</b> / 0.403	26.54 / 0.381	26.89 / 0.380	26.34 / <b>0.372</b>	26.45 / <b>0.372</b>	27.32 / 0.385	<b>27.50 / 0.367</b>
BSD	24.56 / 0.337	<b>24.68</b> / 0.328	24.07 / 0.370	24.18 / 0.364	23.19 / 0.315	23.42 / <b>0.310</b>	24.48 / 0.336	<b>25.07 / 0.293</b>
Image Denoising [36]								
DIV2K	29.75 / 0.233	29.55 / 0.236	29.47 / 0.275	29.62 / 0.263	<b>30.80 / 0.215</b>	29.61 / <b>0.215</b>	30.05 / 0.225	<b>31.25 / 0.192</b>
DPED	30.24 / 0.218	29.87 / 0.230	29.48 / 0.261	29.60 / 0.255	<b>31.23</b> / 0.195	30.09 / <b>0.191</b>	31.15 / 0.203	<b>31.36 / 0.181</b>
BSD	29.92 / 0.240	29.71 / 0.248	29.39 / 0.285	29.55 / 0.262	<b>30.40 / 0.203</b>	29.81 / <b>0.203</b>	30.39 / 0.214	<b>30.42 / 0.192</b>
JPEG Artefact Removal [36]								
DIV2K	26.50 / 0.303	<b>26.71</b> / 0.295	26.32 / 0.295	26.37 / 0.301	26.48 / 0.315	26.27 / <b>0.281</b>	26.50 / 0.299	<b>26.77 / 0.261</b>
DPED	26.20 / 0.305	<b>26.15</b> / 0.301	25.95 / 0.298	26.05 / 0.305	26.01 / 0.307	25.87 / <b>0.296</b>	26.12 / 0.305	<b>26.53 / 0.276</b>
BSD	25.64 / 0.316	<b>25.71</b> / 0.310	25.43 / 0.309	25.49 / 0.313	25.54 / <b>0.308</b>	25.39 / <b>0.308</b>	25.52 / 0.312	<b>25.75 / 0.293</b>

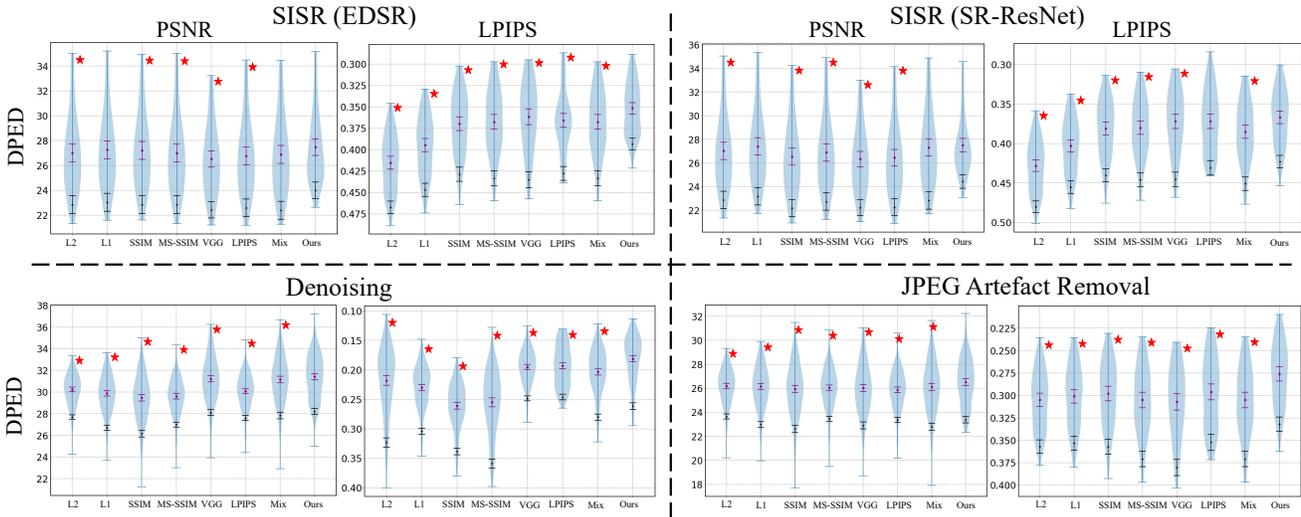


Figure 12: Additional violin plots illustrating the distribution of the PSNR [dB]  $\uparrow$  and LPIPS  $\downarrow$  values for DPED dataset [12] for all applications. Note that the y-axis is reversed for LPIPS so that the quality improves towards the top of each plot. The error bars show the 95% confidence intervals for the mean (magenta) and the 5<sup>th</sup> percentile (black). The latter CIs were computed by bootstrapping. The red asterisks indicate that one-tailed t-test on the means gives statistically significant difference at  $\alpha = 0.05$ . It is worth noting that our loss produced fewer images with low quality values.

Additional qualitative results are provided in the HTML report.

## 6. Image quality metrics and loss functions

To further investigate the performance of loss functions as quality predictors, we generated a set of images that were distorted by blur, noise, added sinusoidal grating, contrast and brightness changes. The distortions were generated so that they degraded the image in equal steps of PSNR.

Fig. 16 presents an example of images with introduced distortions at three PSNR levels. The experiment shows a failure case of PSNR, predicting the same quality even though the distortions due to contrast and brightness are much less objectionable than the others to a human observer.

In Fig. 17, we show the loss values computed for the increasing amount of distortions of different types for different loss functions. Despite the same PSNR value, the distortions due to noise, blur and added sinusoidal wave

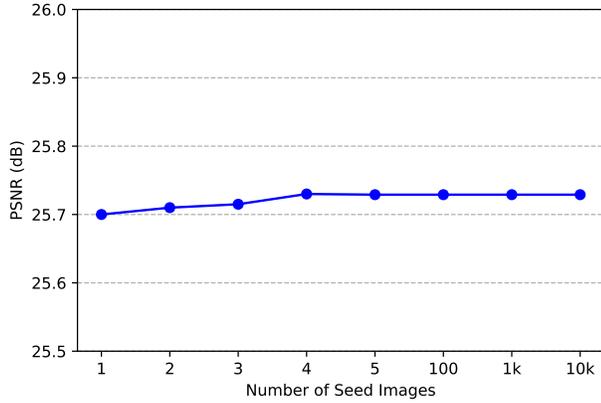


Figure 13: Performance of EDSR model with the increasing the number of seed images used for training the MDF loss function. Note that PSNR increases only by 0.03 dB and saturates for larger number of images. The inference results are reported for the BSD dataset.

are much more noticeable than those due to contrast and brightness change (refer to Fig. 16). The loss functions derived from quality metrics (SSIM, MS-SSIM) and also feature-wise losses (VGG, LPIPS) penalize more the distortions that result in higher degradation of quality. In contrast, MDF losses penalize the most the distortions that are relevant for a given task: blur in case of SISR (MDF SR), blur and noise in case of denoising, and contrast followed by the mixture of all distortions in case of JPEG artifact removal. This is another example demonstrating that an effective loss (MDF) function does not need to predict image quality.

## Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement N° 725253–EyeCode).

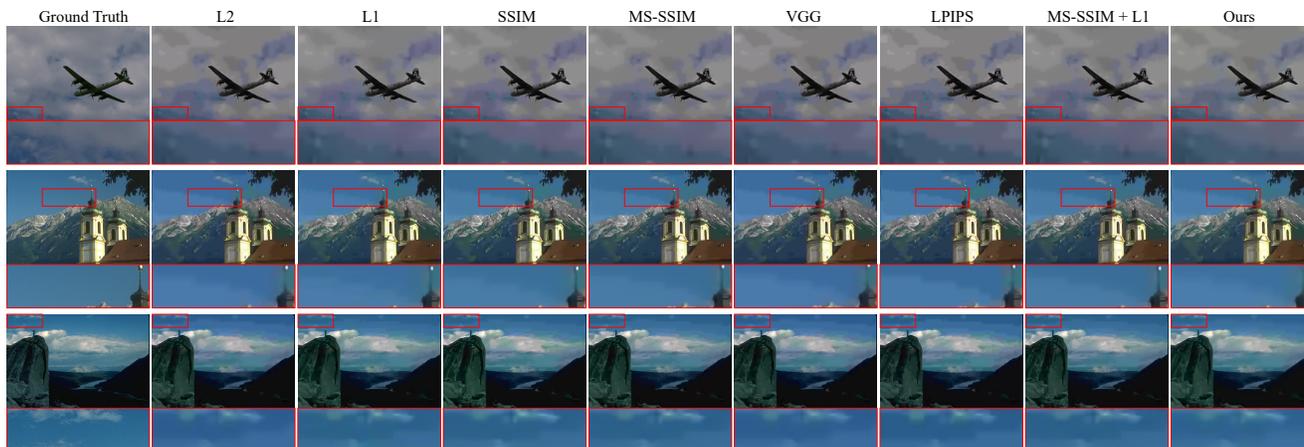


Figure 14: Results for JPEG artefact removal (compression quality = 7) using DnCNN model [36] trained using different losses. Our loss improves artefact reduction, especially in the uniform areas of an image. Qualitative results in terms of PSNR and LPIPS are reported in Table 3. Best viewed when zoomed.

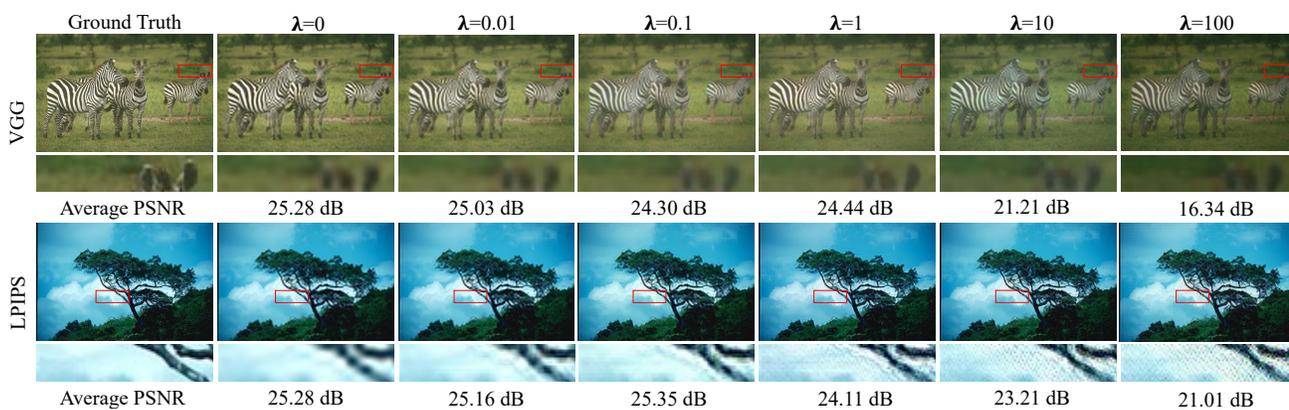


Figure 15: Comparison of the single-image super resolution (SISR) results (EDSR) when trained using a weighted sum of VGG/LPIPS and MSE feature-wise losses:  $MSE + \lambda VGG/LPIPS$ . The average PSNR is reported for the entire test set.

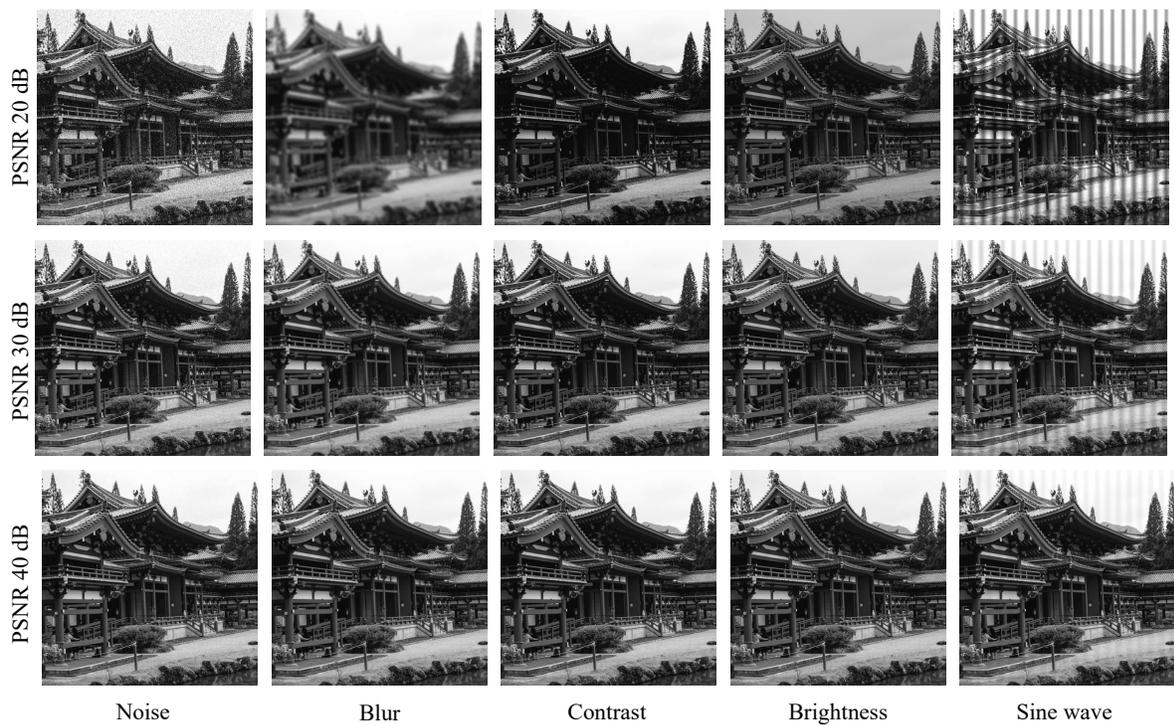


Figure 16: Examples of images used to test the sensitivity of loss functions to different types of distortions. We introduced artifacts so that the each distortion results in the same PSNR level (across each row). Here we provide examples of images at 20 dB, 30 dB and 40 dB. Note that the perceived quality differs between the columns despite the same PSNR level.

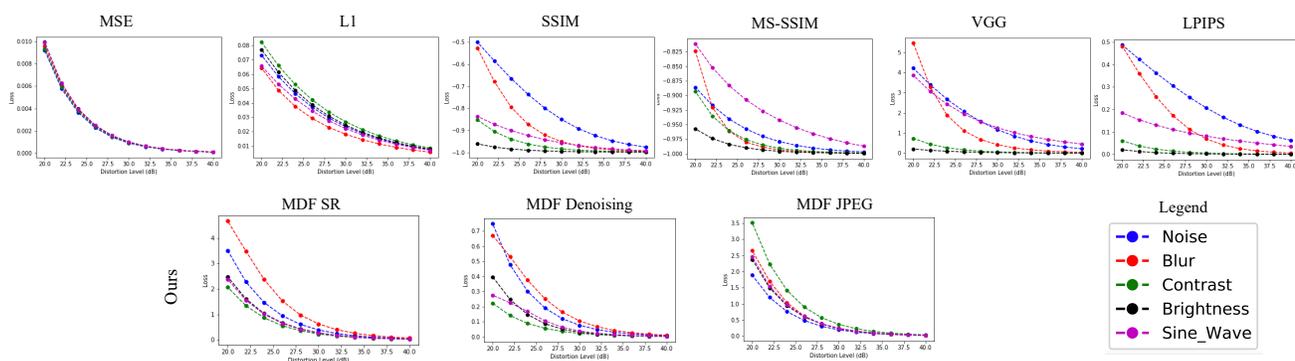


Figure 17: Loss values for the increasing amount of distortions of different types. The distortion levels have been generated to result in equal PSNR values, shown on the x-axis. Despite the same PSNR value, the distortions due to noise, blur and added sinusoidal wave are much more noticeable than those due to contrast and brightness change (refer to Fig. 16). The MDF loss accurately predicts the perceived magnitude of task specific distortions for which it is trained.

## References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 4, 5, 10
- [2] Y. Bahat and T. Michaeli. Explorable super resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2713–2722, 2020. 2
- [3] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6228–6237, 2018. 1, 2, 7
- [4] Mauricio Delbracio, Hossein Talebi, and Peyman Milanfar. Projected distribution loss for image enhancement. *arXiv preprint arXiv:2012.09289*, 2020. 1, 2
- [5] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*, 2015. 9
- [6] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Comparison of full-reference image quality models for optimization of image processing systems. In *International Journal of Computer Vision*, pages 1573–1405. Springer International Publishing, 2021. 1, 2
- [7] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 576–584, 2015. 5
- [8] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, page 658–666, Red Hook, NY, USA, 2016. Curran Associates Inc. 2
- [9] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Deep generative adversarial compression artifact removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4826–4835, 2017. 5
- [10] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*, pages 262–270, 2015. 4
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 1, 2, 9
- [12] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3277–3285, 2017. 4, 5, 10, 11
- [13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2
- [14] Younghyun Jo, Sejong Yang, and Seon Joo Kim. Investigating loss functions for extreme super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 424–425, 2020. 1, 2
- [15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2, 4
- [16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2, 5, 9, 11
- [17] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2479–2486, 2016. 9
- [18] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 5, 10, 11
- [19] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 4, 5
- [20] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 9
- [21] Aliaksei Mikhailiuk, María Pérez-Ortiz, and Rafał K. Mantiuk. Psychometric scaling of TID2013 dataset. In *International Conference on Quality of Multimedia Experience (QoMEX)*, 2018. 8
- [22] Aamir Mustafa and Rafal K Mantiuk. Transformation consistency regularization—a semi-supervised paradigm for image-to-image translation. *arXiv preprint arXiv:2007.07867*, 2020. 9
- [23] Maria Perez-Ortiz and Rafal K. Mantiuk. A practical guide and software for analysing pairwise comparison experiments. *CoRR*, 2017. 6, 7
- [24] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 2
- [25] 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. 2, 9
- [26] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture

- synthesis. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4501–4510, 2017. [2](#)
- [27] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SingGAN: Learning a generative model from a single natural image. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 6228–6237, 2019. [3](#), [4](#)
- [28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. [1](#), [2](#)
- [29] Taimoor Tariq, Okan Tarhan Tursun, Munchurl Kim, and Piotr Didyk. Why are deep representations good perceptual quality features? In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 445–461, Cham, 2020. Springer International Publishing. [2](#)
- [30] Akella Ravi Tej, Shirsendu Sukanta Halder, Arunav Pratap Shandeelya, and Vinod Pankajakshan. Enhancing perceptual loss with adversarial feature matching for super-resolution. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. [2](#)
- [31] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009. [2](#)
- [32] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018. [2](#), [7](#)
- [33] Zhou 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. [2](#), [4](#)
- [34] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003. [2](#), [4](#)
- [35] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with perceptual and contextual losses. *arXiv preprint arXiv:1607.07539*, 2(3), 2016. [2](#), [9](#)
- [36] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. [5](#), [11](#), [13](#)
- [37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep networks as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [1](#), [2](#), [4](#)
- [38] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017. [1](#), [2](#), [4](#)