Training a Better Loss Function for Image Restoration

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

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

Abstract

Central to the application of neural networks in image restoration problems, such as single image super resolution, is the choice of a loss function that encourages natural and perceptually pleasing results. A popular choice for a loss function is a pre-trained network, such as VGG and LPIPS, 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. In this work, we explore the question of what makes a good loss function for an image restoration task. First, we observe that a single natural image is sufficient to train a lightweight feature extractor that outperforms state-of-the-art loss functions in single image super resolution, denoising, and JPEG artefact removal. We propose a novel Multi-Scale Discriminative Feature (MDF) loss comprising a series of discriminators, trained to penalize errors introduced by a generator. Second, we show 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

Image restoration is defined as a problem of recovering a pristine image from its distorted counterpart, which could have been corrupted by noise, under-sampling, blur, compression, etc. 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. One of the critical choices effecting the CNNs performance is the loss function. A popular mean-squared error (MSE or $L_2$) loss often results in blurry, splotchy [35] 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 [4]. 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 [4]. Further, GANs are also 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 [26]). Such losses have been successful in training learning-based image restoration models. However, a major drawback of these loss functions is their dependence on a large external network. 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. [34] 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 MSE 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. First, we demonstrate that a feature extractor for a loss function that is trained on a single image can outperform popular feature-wise (perceptual) losses trained on very large datasets. This is possible, because we train our multi-scale loss to penalize the errors introduced by a generator, rather than to learn the distribution of natural images. The former task is much easier and more relevant for a loss function. Second, we demonstrate that a loss function does not need to predict image quality and that there is no clear correlation between the performance as a loss function and the performance as a quality metric. Both observations

*Equal contribution
go against the common notion that a network needs to extract semantic or style information to serve as an effective loss [14], or that it needs to predict perceptual image quality [34, 35].

The main contributions of our paper are as follows:

1. We demonstrate that an effective feature extractor for a loss function can be trained on a single image when it is trained to detect the errors of a generator used for a given image restoration task.

2. We propose feature-wise loss functions for super-resolution, denoising and JPEG artifact removal, which outperform the state-of-the-art losses, do not require hyper-parameter tuning or to be combined with other losses, and are more light-weight than the perceptual loss (VGG).

3. We provide a performance comparison of most commonly used losses ($L_1$, $L_2$, SSIM, MS-SSIM, MS-SSIM + $L_1$ [35], VGG, and LPIPS) in terms of objective and subjective quality, and analyze them from the perspective of perception-distortion trade-off [3].

4. We demonstrate that an effective loss function does not need to predict image quality and a good quality metric does not necessarily make a good loss function.

2. Related work

The choice of loss function greatly influences the quality of images produced by the image restoration algorithms. 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 a deep learning model, and distribution losses, where the loss pushes the solution to the manifold of natural images.

**Hand-crafted losses:** Zhao et. al [35] have studied visual quality of images produced by the image super-resolution, denoising and demosaicing algorithms using $L_2$, $L_1$, SSIM [30] and MS-SSIM [31] 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 test image can be computed in the feature space of the 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 [14] used the $L_2$ norm between the features of the reference and test images extracted from the VGG [26] network as a loss function to train style-transfer and super-resolution algorithms. Here the VGG network was trained on ImageNet dataset [23]. Authors in [34] (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 [27] 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 [4]. 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 [4]. 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.

**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 are 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 [32, 12] authors used $L_1$ norm to regularize the training. Similarly the works of [8, 15] used the feature-wise VGG-based loss to constraint the generator. Some other works combined both hand-crafted and feature-wise losses [24, 29]. Others have introduced regularization based on the feature loss of the discriminator [28, 13]. 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.
3. Multi-Scale Discriminative Feature Loss

Feature-wise (perceptual) loss functions are commonly used 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 [15]. However, a fundamental weakness of such methods is that they owe much of their success to delicate hyper-parameter tuning and large datasets they are trained on. To this end, we introduce our Multi-Scale Discriminative Feature (MDF) loss, which aims to learn a feature-wise loss function from a single image. 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 image restoration task in hand.

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 generator. 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. The discriminator networks are trained as a single-image GAN [25] that removes a task-specific distortion from a seed image. We denote the seed image by $y$ to differentiate it from the training images, denoted by $\{x_i\}_{i=1}^N$, 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. Training the discriminators

We use the architecture of SinGAN [25] 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 $k$, $y^k$, 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 upsampled output of the lower scale after adding the task specific artefacts to it $\tilde{y}^k = G^k(\tilde{y}^{k-1} + \rho^k + z^k)$. Here $z^k$ is the distortion added to the upsampled output of the lower scale $\tilde{y}^{k-1} + \rho^k$. For the first scale of training, the input to the generator is the original image downsampled by a fac-

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 upsampled output of the previous level $\tilde{y}^{k-1} + \rho^k$ added with the task specific distortions $z^k$. For SISR, no distortions are added ($z^k = 0$). 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 $x_i$ and the restoration output $\tilde{x}_i$. The distance is measured between the two images at every scale $k$ and intermediate layers of the discriminator $||\phi^k_i(x_i) - \phi^k_i(\tilde{x}_i)||_2^2$. 

Phase 1: Training the discriminators
Phase 2: Reconstruction Loss
tor 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 [25], we use application-specific distortion $z^k$ while training. For image denoising, $z^k$ is Gaussian Noise of a magnitude randomly sampled from a uniform distribution of $[0,55]$ on a pixel scale of $(0,255)$. For JPEG artefact 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 ($z^k = 0$) 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 $\hat{y}^k$ and produces a map of $[0,1]$ activations $\hat{u}^k = D^k(\hat{y}^k)$ with the same dimensionality as $\hat{y}^k$. Alternatively, the discriminator can be supplied with with the downscaled seed image $y^k$, resulting in the activation map $u^k$. The discriminator is trained to distinguish patches of the seed image from patches of the generated image and, therefore, the activations of $u^k$ are pushed towards 1 and those of $\hat{u}^k$ 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:

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

(1)

The reconstruction loss $\mathcal{L}_{rec}$ employed is the MSE loss between the generated $\hat{y}^k$ and the ground truth image $y^k$ 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 [25].

It must be noted that addition of the above distortions (Gaussian Noise and JPEG compression artefacts) to the seed image at various scales makes the discriminator sensitive to such artefacts 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. 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 $\phi^k_l(x)$ meaning the output of the $l$-th layer of the discriminator for the $k$-th scale. The output of the whole discriminator is then $\phi^k_L(x)$, where $L$ is the total number of layers.

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 for the MDF loss will not be scaled and it will be $H_y \times W_y$, the size of the $x_i$.

4. Image manifold assumption

The main objective of GANs [11] in image restoration is to learn a discriminator model that differentiates between image manifolds [16, 32, 21, 19, 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 photorealistic images closer to the natural image manifold [15, 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 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 as 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 [23]. 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. 2 shows the plot of the features from each set of images. The visualization 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.

5. 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. Specifically, we perform experiments for Single Image Super-Resolution (SISR), image denoising, and JPEG artefact removal. For all the above cases, we provide a comprehensive comparison with a series of baseline loss functions, listed in Table 1, including the most widely used perceptual loss [10, 14]. Note that for VGG and LPIPS, losses must be combined with the MSE 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 \( \{ \lambda : \lambda = 10^k, k = -3, \ldots, 3 \} \) 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 image 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 MSE regularization while training. It is also worth noting that our MDF loss function is less computationally expensive 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 function for the task of SISR, which aims at estimating a High-Resolution (HR) image from a given Low-Resolution (LR) image. For this application, we use two state-of-the-art SISR architectures, namely Enhanced Deep Super-Resolution (EDSR) [17] and SR-ResNet [15]. The LR image is first generated by downsampling the original HR image by a factor of 4 using bicubic filter. Both models are trained on the DIV2K dataset [1] and the inference is run on Berkeley Segmentation Data (BSD 500) [18]. We used cross-dataset validation as we found the task to be too easy to discriminate between losses when tested on the split of the same dataset. For training, we randomly extract 96 × 96 patches from the dataset and perform data augmentation with 90°, 180° and 270° rotations, and horizontal and vertical flips. The models are trained for a super-resolution factor \( \delta \) of 4. Each model is trained for 500 epochs with an initial learning rate of 0.001 with gradual learning rate scheduling. The best model is selected based on the validation loss. The quantitative results, shown in Table 2, indicate that our loss produces images with higher PSNR and lower (lower) NIQE score. The advantage of our loss is better seen when images are directly compared in Fig. 3.

<table>
<thead>
<tr>
<th>Metric</th>
<th>L2</th>
<th>L1</th>
<th>SSIM</th>
<th>MS-SSIM</th>
<th>VGG</th>
<th>LPIPS</th>
<th>MS-SSIM + L1</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>25.28</td>
<td>25.06</td>
<td>25.12</td>
<td>25.15</td>
<td>24.44</td>
<td>25.35</td>
<td>25.08</td>
<td>25.37</td>
</tr>
<tr>
<td>NIQE</td>
<td>0.695</td>
<td>0.704</td>
<td>0.719</td>
<td>0.716</td>
<td>0.654</td>
<td>0.651</td>
<td>0.685</td>
<td>0.699</td>
</tr>
<tr>
<td>MS-SSIM</td>
<td>5.964</td>
<td>5.761</td>
<td>5.691</td>
<td>5.785</td>
<td>6.174</td>
<td>5.812</td>
<td>4.325</td>
<td>3.635</td>
</tr>
</tbody>
</table>

### Image Denoising

<table>
<thead>
<tr>
<th>Metric</th>
<th>L2</th>
<th>L1</th>
<th>SSIM</th>
<th>MS-SSIM</th>
<th>VGG</th>
<th>LPIPS</th>
<th>MS-SSIM + L1</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>24.56</td>
<td>24.68</td>
<td>24.07</td>
<td>24.18</td>
<td>23.19</td>
<td>23.42</td>
<td>24.48</td>
<td>25.07</td>
</tr>
<tr>
<td>NIQE</td>
<td>0.692</td>
<td>0.695</td>
<td>0.695</td>
<td>0.689</td>
<td>0.608</td>
<td>0.609</td>
<td>0.670</td>
<td>0.602</td>
</tr>
<tr>
<td>MS-SSIM</td>
<td>5.904</td>
<td>5.761</td>
<td>5.676</td>
<td>5.496</td>
<td>4.915</td>
<td>4.725</td>
<td>4.215</td>
<td>4.104</td>
</tr>
</tbody>
</table>

### JPEG Artefact Removal

<table>
<thead>
<tr>
<th>Metric</th>
<th>L2</th>
<th>L1</th>
<th>SSIM</th>
<th>MS-SSIM</th>
<th>VGG</th>
<th>LPIPS</th>
<th>MS-SSIM + L1</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>25.64</td>
<td>25.71</td>
<td>25.43</td>
<td>25.49</td>
<td>25.54</td>
<td>25.39</td>
<td>25.32</td>
<td>25.75</td>
</tr>
<tr>
<td>NIQE</td>
<td>0.705</td>
<td>0.716</td>
<td>0.725</td>
<td>0.737</td>
<td>0.711</td>
<td>0.708</td>
<td>0.710</td>
<td>0.714</td>
</tr>
<tr>
<td>MS-SSIM</td>
<td>5.901</td>
<td>5.734</td>
<td>5.956</td>
<td>5.715</td>
<td>5.998</td>
<td>5.345</td>
<td>5.665</td>
<td>4.910</td>
</tr>
</tbody>
</table>

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

Table 2: Comparison of our proposed Multi-Scale Discriminative Feature (MDF) loss function with other losses on four tested applications.

\( ^2 \)This is additional training on top of a pre-trained network using 1.3M ImageNet dataset

\( ^3 \)Only training of discriminators is performed in a multi-scale fashion.
Figure 3: SISR results for EDSR [17] trained using different loss functions. The results for our loss are sharper images and have fewer artifacts. The average PSNR values are reported for the entire test set. Best viewed when zoomed.

Figure 4: Results for denoising using DnCNN model [33] trained using different losses. Our loss improves noise reduction, especially in the uniform areas of an image. Best viewed when zoomed.

**Image denoising** Next, we evaluate the efficacy of our MDF loss on image denoising. We train the DnCNN architecture proposed by Zhang et al. [33] on the BSD dataset [18]. 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 Nestrov 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. The best model is selected based on the validation loss, as done for SISR. Quantitative results in Table 2 show that our loss produces better results in terms of PSNR and NIQE. The qualitative results in Fig. 4 show that training with our loss results in the best noise suppression, especially in the uniform areas. It also does not result in blurring or spurious details, which is often produced when DnCNN is trained on other losses.

**JPEG artefact removal** For this application, the same DnCNN [33] as for the denoising task is employed. We train the network on the DIV2K dataset [1] and test on BSD500. We used cross-dataset validation as we found the task to be too easy to discriminate between losses when tested on the split of the same dataset. 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° image rotation, vertical and horizontal flips. The model is trained with Adam optimizer and the learning rate set to $10^{-4}$. The test images are compressed with a quality factor of 10 and a more challenging factor of 7. Quantitative results, for the compression quality factor of 7, in Table 2 show that our loss produces better results in terms of PSNR and NIQE. We provide qualitative results for JPEG artefact removal in the appendix.
Comparison with Adversarial Loss  We further explored the performance of a model trained using our MDF function with that trained using the adversarial loss. For this experiment, we chose the task of SISR and used state-of-the-art GAN based method — ESRGAN [29]. Both models were trained on the DIV2K dataset and tested on the BSD dataset. The model trained using our MDF loss achieved a PSNR of 25.37 dB as compared to ESRGAN’s 25.06 dB. 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 (BSD) dataset with each pairwise comparison performed four times. MDF has an advantage over ESRGAN and was selected in 58% of the comparisons.

5.1. Subjective experiment

Objective metrics such as PSNR or NIQE, 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 0.09 dB is going to be appreciated by an average observer. For that reason, we run a perceptual experiments on the Amazon Mechanical Turk crowd-sourcing platform.

For best sensitivity of the test, we used full-design pairwise-comparison protocol [22]. In each trial, participants were presented with 3 side-by-side images: one reference and two generated by the image reconstruction methods, each with different loss function. Participants were asked to select the image that appeared closer to the reference. For fair comparison, we randomly selected 50 images from the corresponding test sets. 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 [22]. The results of scaling expresses 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 Figure 5, show consistent improvement of our method over other loss functions. MS-SSIM + L1 performed the second best for SISR on the EDSR, with MDF having small advantage of 0.05 JND.

To gain further insights, in Fig. 6 we visualize the results as the perception-distortion trade-off [4], 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 L2 – 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.

6. Ablation analysis

The ablation studies test the importance of task-specific MDF, the number of images used to train the discriminator, the choice of seed image and the number of discriminator scales. The latter two studies are described in the appendix.
Task-specific distortions  Here we test whether a generator trained on one task can serve as a feature-wise loss for another task. We train DnCNN for JPEG artifact removal using MDF with introduced either noise or JPEG distortions (different vectors $z^k$). The task-specific discriminator gained moderate performance increase in terms of PSNR (25.75 dB for MDF JPEG and 25.61 dB for MDF noise) but resulted in images of much better visual quality (see Fig. 7).

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. 8 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 in negligible when adding more seed images, we used a single image for training in our results.

7. 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 [20]. The dataset is one of the most accurate (due to large number of comparisons), is scaled in JND units, and contains sufficiently 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.

8. Conclusions

In this paper, we have shown several observations that go against the common assumptions of what make a good loss function. 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 generator errors, but it involves more challenging and less stable optimization problem. The second important observation is that an effective loss function does not need to predict perceived image quality and a good quality metric may not make a good loss function. Both tasks may be unrelated.

It has been argued that image restoration tasks are often ill-defined as an input image, for example a lower resolution image, may be upscaled to multiple, equally plausible high resolution images [2]. GAN architectures are intended to address this problem by steering the generator to produce images that belong to the natural image manifold. But the discriminator, whose task is to discriminate between reference (sharp) and generated (blurry) images, may be more likely to learn to detect image sharpness rather than to learn a complex manifold of natural images. It is yet to be determined to what extend training a discriminator network on very large dataset results in learning a complex natural image manifold and to what extend it results in learning to detect blur and other artifacts.
Appendix

The appendix includes additional details that could not be included in the main paper due to the lack of space. This comprises: a) manifold visualization for SR-GAN discriminator as compared to our multi-scale discriminators; b) qualitative results for the JPEG artefact removal application; c) ablation study on the choice of seed images and the number of discriminators of the Multi-Scale Discriminative Feature (MDF) loss; d) hyper-parameter tuning for the VGG and LPIPS feature-wise loss functions; and e) performance of loss functions as quality predictors.

1. Image manifold comparison

In this section, we repeat the experiment conducted in Sec. 4, instead this time for a fully trained SR-GAN [15] 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 [23]. 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. 10 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 discriminate between the JPEG compressed and original images, making it an inferior feature extractor to detect and remove distortions.

2. 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. 11 shows the results for the compression quality factor 7. The 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. 5.1 of the main paper). Additional qualitative results are provided on the Project website.

3. Ablation study

3.1. 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) to achieve the best performance. We perform an ablation study on training the EDSR model [17] 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 3, our loss performs the best when all 8 scales are employed.

### Table 3: Ablation study on training the SISR model (EDSR) using different scales of our loss.

<table>
<thead>
<tr>
<th>Scales</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR↑</td>
<td>22.55</td>
<td>23.89</td>
<td>24.43</td>
<td>24.89</td>
<td>25.27</td>
<td>25.37</td>
</tr>
<tr>
<td>SSIM↑</td>
<td>0.51</td>
<td>0.59</td>
<td>0.62</td>
<td>0.66</td>
<td>0.68</td>
<td>0.70</td>
</tr>
<tr>
<td>NIQE↓</td>
<td>6.109</td>
<td>5.358</td>
<td>4.953</td>
<td>4.124</td>
<td>3.979</td>
<td>3.635</td>
</tr>
</tbody>
</table>

shows the results for the compression quality factor 7. The 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. 5.1 of the main paper). Additional qualitative results are provided on the Project website.

3.2. Seed image

Next, we study the effect of using different natural and synthetic images for training our MDF loss function. Fig. 12 shows five seed images including two natural and
Figure 11: Results for JPEG artefact removal (compression quality = 7) using DnCNN model [33] trained using different losses. Our loss improves artefact reduction, especially in the uniform areas of an image. Qualitative results in terms of PSNR, SSIM and NIQE are reported in Table 2. Best viewed when zoomed.

three 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. 12, indicate that the visual quality of the super-resolved images is the best for natural images and is degraded as the statistics of the training images is distorted. However, from the results for all the applications, the visual quality of the restored images is more dependent on the nature of the distortions added ($z^k$) than the choice of the seed image.

5. 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. 14 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. 15, 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 are much more noticeable than those due to contrast and brightness change (refer to Fig. 14). 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 No 725253–EyeCode).
Figure 12: 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 13: 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 + λ VGG/LPIPS. The average PSNR is reported for the entire test set.

References


Figure 14: 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 15: 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. 14). The MDF loss accurately predicts the perceived magnitude of task specific distortions for which it is trained.


