Projected Distribution Loss for Image Enhancement

Mauricio Delbracio, Hossein Talebei and Peyman Milanfar

Abstract—Features obtained from object recognition CNNs have been widely used for measuring perceptual similarities between images. Such differentiable metrics can be used as perceptual learning losses to train image enhancement models. However, the choice of the distance function between input and target features may have a consequential impact on the performance of the trained model. While using the norm of the difference between extracted features leads to limited hallucination of details, measuring the distance between distributions of features may generate more textures; yet also more unrealistic details and artifacts. In this paper, we demonstrate that aggregating 1D-Wasserstein distances between CNN activations is more reliable than the existing approaches, and it can significantly improve the perceptual performance of enhancement models. More explicitly, we show that in imaging applications such as denoising, super-resolution, demosaicing, deblurring and JPEG artifact removal, the proposed learning loss outperforms the current state-of-the-art on reference-based perceptual losses. This means that the proposed learning loss can be plugged into different imaging frameworks and produce perceptually realistic results.

Index Terms—Computational Photography

1 Introduction

Image restoration has seen remarkable progress in recent years mostly coming hand-in-hand with the success of deep neural networks. Greater computational power, stable and accessible training frameworks as well as a large amount of data have enabled deep image processing models that exceed or are on par with those conceived through careful and artisan modeling.

However, how to train deep models in such a way that the restored images capture the realism of natural images remains an open challenge. The most popular approach for training deep image models is through supervised learning in which a loss function measuring the difference with respect to a reference or ground-truth image is minimized [1], [2], [3], [4].

Perhaps the most common approach is to use a loss function that measures the difference directly between pixel values by some standard norm (e.g., L_1 or L_2). The pixel loss suffers from the well-known problem of regression to the mean. Since inverse problems are generally poorly conditioned, there are countless possible explanations for a given observed image. Minimizing a L_2 pixel loss leads to predicting an average image that generally looks blurry and lacks of details (e.g., grain, noise, edge contrast).

Generative adversarial networks [5], [6], and in particular adversarial losses [7], [8], [9], [10] are among recent approaches that lead to more realistic images. However, these networks are generally hard to train, due to a min-max type optimization [6], [11], [12]. Additionally, since GANs are trained to minimize the distance to the manifold of natural images, they generally introduce significant hallucinations: what is perceived as real content for one image may be seen as hallucination in a different image [13].

Johnson et al. [15] showed that networks trained for image recognition tasks produce deep features (i.e., low / mid / high level image representations) that can capture perceptual information well. Based on this powerful observation, the authors proposed to augment a pixel loss (or replace it) by penalizing the L_1/L_2 difference in a deep feature space. The perceptual loss leads to images with more details and improve the missing realism ingredient. This has been recently analyzed in [16], [17] and further exploited in the contextual loss [18], [19] and contextual bilateral loss [20].

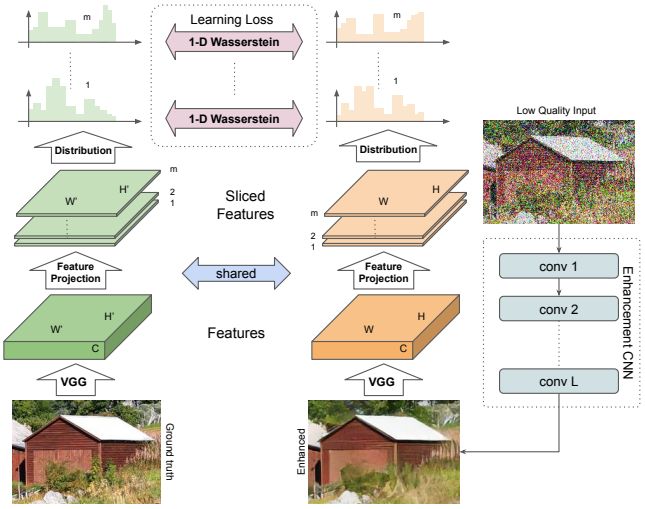


Fig. 1. The proposed image enhancement framework. Our enhancement CNNs are trained with a loss that measures 1-D Wasserstein distance between distributions of the VGG features [14].

In this work, we build on recent findings and introduce a new loss function that measures differences in the feature space by comparing the distributions of an image and its reference counterpart. This allows recovering details (texture/grain) without forcing them to be located in the exact same spatial position as in the reference image. This is a critical flexibility for tackling inverse imaging problems. For instance, one can consider the problem of single image super-resolution where a low resolution

解决空间对齐问题，比较distribution distance

• *The authors are with Google Research, Mountain View, CA 94043 USA. The first two authors contributed equally to this work.*

image can be up-scaled to slightly different high-resolution images with identical feature distributions. This implies that given the low resolution observation, there are multiple high-resolution images that lead to the low-resolution observation. Any pixel loss or L_1 distance on features (Johnson et al. [15]) learns to predict the mean target (i.e., conditional expectation given the low resolution observation), leading to a less realistic (and often blurry) image. On the other hand since we compare an image directly with its reference, the amount of hallucinated content is better controlled than in an adversarial loss.

style transfer

Our proposed loss function has some similarity to style transfer works [21]. In neural style transfer, a convolutional network is used to merge the content of one guide image with the style of another one. A neural network is trained for each different input style by using two different and complementary losses. The content loss is defined as the element-wise distance between feature maps of both images, while style loss penalizes difference in the statistics of the feature maps [21], [22], [23].

Our distribution loss makes explicit use of the fact that the extracted features belong to a metric space and therefore their geometries as well as their distributions are both relevant. This leads us to pose the problem as one of optimal transport [24], in which we seek to minimize the cost of transforming the input feature distribution into the reference one. The optimal transport problem and the Wasserstein distance [24], [25] provide an elegant and formal way to measure distance between distributions by transporting one distribution into another. However, directly minimizing the Wasserstein distance on a high-dimensional distribution is in general intractable [26]. Rabin et al. [27] introduced the sliced-Wasserstein distance that makes explicit use of the closed-form solution in one-dimension as an elegant tractable approximation. The sliced variant is based on the invertibility of the Radon transform, and consists of projecting the data samples and computing an average one-dimensional distance between the marginal distributions at all possible orientations. Our loss function for comparing feature distributions is in effect an implementation of the sliced Wasserstein loss where we aggregate 1D Wasserstein distance on projected distributions.

The proposed framework for training an arbitrary image enhancement model is shown in Figure 1. First, VGG features [14] for the ground truth and the predicted images are obtained. Then, the extracted feature activation maps are projected and for each projected feature map we compute the 1D Wasserstein distance between the predicted and target distributions. Finally, the sum of the 1D Wasserstein distances between the distributions is used in our learning loss. The proposed framework is straightforward, and it can be easily implemented (see Algorithm 1). In comparison with the existing perceptual losses our Projected Distribution Loss (PDL) does not add any significant computational complexity.

We demonstrate effectiveness of the proposed loss on an extensive set of experiments analyzing the effect of adopting different loss functions on five different image restorations tasks: denoising, single-image upscaling (super-resolution), demosaicing, JPEG artifact removal and deblurring. We evaluate the results according to different perceptual and distortion metrics. The proposed distribution loss leads to perceptually superior results without introducing significant distortions.

The remainder of the paper is organized as follows. Section 2 discusses related work and the substantial differences of what we propose. Section 3 explains the mathematical foundations of the distribution loss, and then the details are discussed in Section 4. Section 5 presents extensive results and comparisons to other popular loss functions. Finally we conclude the paper in Section 6.

2 Related work

Image restoration aims to generate a high-quality image from its degraded low-quality measurement (e.g., low-resolution, compressed, noisy, blurry). Recently, deep convolutional neural networks (CNN) have been popular due to their remarkable performance in many image restoration tasks. A detailed analysis of the image restoration literature is beyond the scope of this article. In the following we present the most relevant work.

From deep features to perceptual losses. Briefly after the resurgence of CNNs, Dong et al. [28] presented an end-to-end convolutional neural model that super-resolves an image. The network is trained end-to-end by minimizing the mean squared error (L_2 loss). At that time results were remarkably good comparing to previous approaches that learn very shallow models or assumed a prior on the data (such as sparsity or small TV). In 2016, Zhao et al. [29] executed a thorough experimental comparison of different pixel losses (L_1, L_2, SSIM, MS-SSIM) in three applications (super-resolution, JPEG artifacts removal, and joint demosaicing & denoising). Their study led to proposing a loss that combines L_1 and MS-SSIM that obtains superior restoration results particularly in terms of image artifacts.

Johnson et al. [15] introduced the idea of using deep image features extracted from an auxiliary network to capture perceptual information. Their used deep features are the activation maps of a deep CNN trained for object recognition. Their perceptual loss directly penalizes element-wise differences in the deep feature space leading to superior perceptual results. The perceptual loss has become a de facto standard for deep supervised learning in image restoration applications. However, being also an element-wise distance also suffers from the regression to the mean phenomenon. Zhang et al. [16] introduced an experimental study where they showed that intermediate deep features trained for computer vision tasks capture the low-level perceptual similarity surprisingly well. An image metric is introduce to compare image pairs in a perceptual way (LPIPS). Based on similar observations, Talebi and Milanfar [30], [31] proposed a trainable model to predict the distribution of human opinion scores using deep features. A systematic analysis carried out in [17] shows that deep features are indeed correlated with basic human perception characteristics, such as contrast sensitivity and orientation selectivity. Their findings suggest that a perceptual loss function can potentially select a subset of features that are more correlated with human perception leading to a better perception-distortion trade-off [32].

Motivated by maintaining the statistics of natural images, the contextual loss (CTX) was introduced in [18]. The contextual loss compares the deep features of the generated image with the most similar ones of the reference image (similar context, does not need data to be aligned). The authors adopted this loss in the context of non-aligned image transformation and in particular style transfer. In a follow up the authors show that this loss produces high-quality results on other image processing applications such us super-resolution [19]. The authors also show that the contextual loss can be seen as an approximation of the Kullback-Leibler divergence between the input and target features. In [20] the contextual loss is modified into a contextual bilateral loss (CoBi) that prioritizes local features in the global search of similar features.

Adversarial losses. Adversarial losses [7], [8], [9], [10] based on generative adversarial training [5], [6] have become the golden standard in terms of perceptual quality. The overall idea is to train in an alternate fashion a generator, that learns to generate realistic images and a discriminator that learns to distinguish real from fake images. The trained discriminator can be plug-in on any loss as an extra term that penalizes images that do not look realistic. The price to pay is that in general generated images do not tightly follow the low-resolution observation leading to a superior distortion than perceptual or pixel losses [13].

Our Projected Distribution Loss (PDL). We propose a distribution loss that penalizes difference of generated and target deep image feature distributions based on Optimal Transport theory [24] and in particular with the Wasserstein metric [24], [25]. By comparing the distribution of features in addition to the pixel values, we are able to produce images with a higher level of realism. Furthermore since we compare the generated image directly with its reference, the amount of hallucinated content is better controlled than in a (non-referenced) adversarial loss. Our loss can be seen as an intermediate alternative between the L1-perceptual loss and the adversarial losses. Comparing high-dimensional distributions is a challenging problem. In [27] authors introduce the sliced-Wasserstein distance as a way to circumvent the high-dimensional drawbacks, but also keep the desirable mathematical properties of the Wasserstein distance. The sliced-Wasserstein distance consists of projecting the data distribution into all possible directions and then computing the average value. The Wasserstein and sliced-Wasserstein have recently received attention in a number of different computer vision applications [25], [33], [34], [35]. We use the sliced Wasserstein distance to compare the feature distributions by adding the comparison of 1D marginal distributions of projected features. The projected distribution loss works as a complement to a pixel fidelity loss term.

3 Comparing Feature Distributions

There are several ways to define a distance (or a quasi-distance) between distributions [36]. Among the most popular ones are the Kullback-Leibler (KLD) and Jensen-Shannon (JSD) divergences, the total variation, and the Wasserstein distance. A relevant characteristic of the Wasserstein distance is that it uses the geometry of the domain to explicitly penalize the differences between the distributions. In Figure 2 (a) we present a toy example where a base discrete distribution (histogram) is shifted a variable amount. As the distribution shifts, the KLD and the JSD divergences remain constant while the Wasserstein distance increases. Examples of distances between distributions of VGG16 features [14] are shown in Figure 2 (b)-(e). The first row shows some distributions from a low resolution input, and the second row shows the high-resolution counterparts. These results show how the Wasserstein distance (measured as the Earth Mover's Distance) is more sensitive to the shape of the distributions.

Since our goal is to compare distributions of features, and given that distributions by nature belong to a metric space, we adopt the Wasserstein distance. In the following we present a short summary explaining the Wasserstein distance and the efficient variant that leads to our projected distribution loss.

3.1 Optimal Transport and the Wasserstein distance

Our proposed loss can be formulated as a problem of optimal transport in a metric space. Let I_u and I_v be two probability density functions (pdf) defined on $\mathbb{R}^d$. This two pdf represent the two distributions that we want to compare. The goal of optimal transport is to find a coupling π (also known as a transport plan) that transforms I_u into I_v with minimal cost. The solution to this problem leads to the p-Wasserstein distance between I_u and I_v,

$$W_p^p(I_u, I_v) = \inf_{\pi \in \Pi(I_u, I_v)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^p \pi(x, y) dx dy, \quad (1)$$

where $\Pi(I_u, I_v)$ denotes the space of all joint distributions π having marginals I_u and I_v. Unfortunately, the optimal transport problem in $\mathbb{R}^d$ does not have a closed form solution and an optimization scheme is needed [25].

3.2 The 1D Wasserstein distance

In the particular case where the densities are defined on the real line, the one-dimensional p-Wasserstein distance has a closed form. In this case, it can be shown that,

$$W_p^p(I_u, I_v) = \int_0^1 \left| F_u^{-1}(s) - F_v^{-1}(s) \right|^p ds, \quad (2)$$

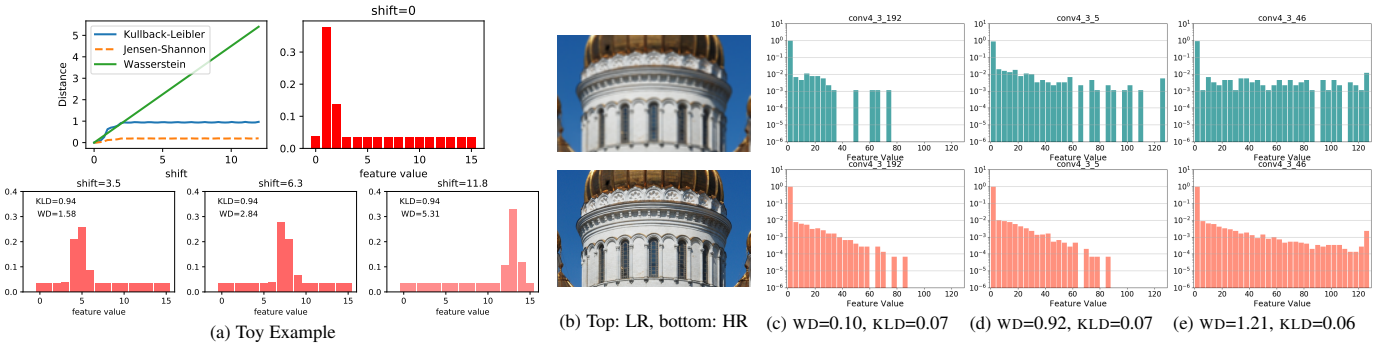
where $F_u(s), F_v(s)$ are the cumulative distribution functions (CDF) of I_u and I_v, respectively [25].

Numerical Implementation. In our setting, we want to compare the empirical distributions of the projected features from two images. Let $\{a_i\}_{i=1}^n$ and $\{b_i\}_{i=1}^n$ represent the set of one-dimensional (projected) features of the two images to compare. In this case, the cumulative distribution functions are directly replaced by the empirical distributions, that is, $F_{\mathbf{a}}(s) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\{a_i \leq s\}}$ and $F_{\mathbf{b}}(s) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\{b_i \leq s\}}$, where $\mathbf{1}_{\mathcal{X}}$ is the indicator function of set $\mathcal{X} \subset \mathbb{R}$. Let (a_{i_p}) be the set $\{a_i\}$ sorted in ascending order, i.e., $a_{i_p} \leq a_{i_{p+1}}$ (and similarly (b_{i_p})). The empirical distribution $F_{\mathbf{a}}(s)$ is a non-negative non-decreasing step-wise function having n steps at $s_i = a_{i_p}$ and values $F(s_i) = i/n$. Then (2) can be directly computed by measuring the distance between the sorted sequences,

$$W_p^p(\mathbf{a}, \mathbf{b}) = \int_0^1 \left| F_{\mathbf{a}}^{-1}(s) - F_{\mathbf{b}}^{-1}(s) \right|^p ds = \sum_{p=1}^n |a_{i_p} - a_{i_p}|^p. \quad (3)$$

This implies that for the one-dimensional case, we can compute the Wasserstein distance between the cumulative distributions by sorting the (projected) features. This leads to a straightforward implementation of the 1D-Wasserstein distance in Tensorflow with just a few lines of code. Note that sorting is not a fully-differentiable operation. In fact, $\text{sort}(\mathbf{u}) = \mathbf{P}\mathbf{u}$, where $\mathbf{P}$ is a permutation matrix that depends on $\mathbf{u}$. During the forward pass the right permutation matrix $\mathbf{P}$ is computed. During the backward pass, $\mathbf{P}$ is kept constant, leading to an approximation of the derivative[1].

1. This is straightforward to implement using the Tensorflow function `tf.nn.top_k(.., sorted=True)`.



(a) Toy Example (b) Top: LR, bottom: HR (c) WD=0.10, KLD=0.07 (d) WD=0.92, KLD=0.07 (e) WD=1.21, KLD=0.06

Fig. 2. Distance between distributions. (a) toy example showing how the Wasserstein distance is sensitive to the geometry of the space. A base distribution is translated, while the KLD and JSD divergences remain constant, the Wasserstein distance increases with the shift amount. Since distant features are in fact less similar the Wasserstein distance is a natural way to compare feature distributions. (b-e) Examples of distances between distribution of VGG16 features. The first row shows some histograms from a low resolution input, and the second row shows the histogram counterparts computed at $4\times$ higher resolution. These results show how the Wasserstein distance (WD) is more sensitive to the shape of the feature distributions.

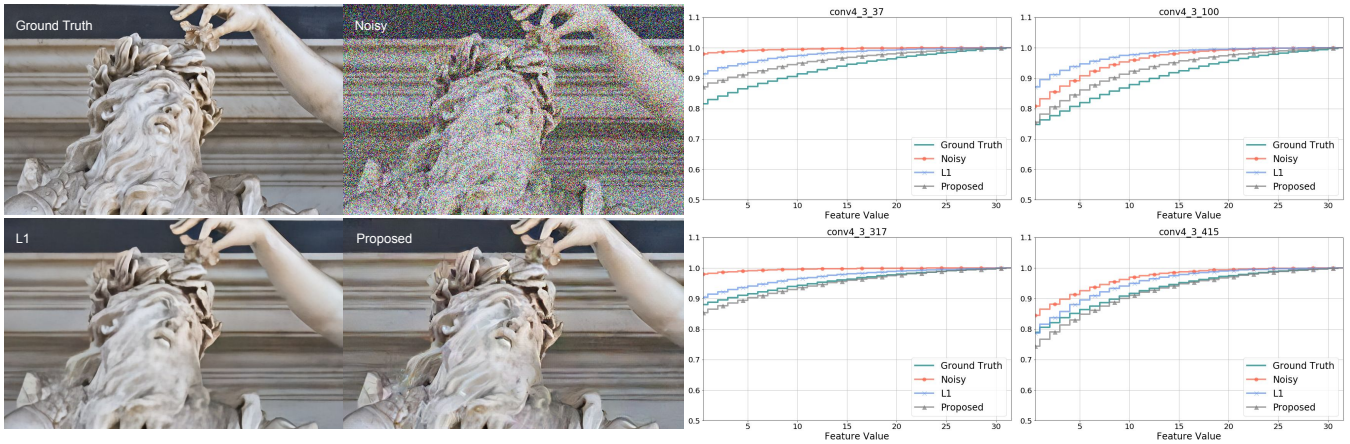


Fig. 3. The cumulative distribution of VGG features. The enhancement CNNs are trained with two different losses: 1) L1 norm of the VGG feature differences (labeled as L1), 2) the sliced-Wasserstein distance of the extracted VGG features (labeled as proposed). As can be seen in the plots, compared to the L1 case, the cumulative distribution of the features obtained from our proposed method is closer to the ground truth. This leads to a perceptually superior result with more details.

Algorithm 1: The proposed PDL training loss.

Input : Predicted image $\mathbf{u} \in \mathbb{R}^n$, ground truth $\mathbf{v} \in \mathbb{R}^n$, projection matrix $\mathbf{W} \in \mathbb{R}^{m' \times m}$, the Wasserstein weight λ

Output: Learning loss $L_{\text{PDL}}(\mathbf{u}, \mathbf{v})$

Compute the VGG features: $\Phi(\mathbf{u})$ and $\Phi(\mathbf{v}) \in \mathbb{R}^{n \times m}$

Projection: $\Phi'(\mathbf{u}) = \Phi(\mathbf{u})\mathbf{W}^T$ and $\Phi'(\mathbf{v}) = \Phi(\mathbf{v})\mathbf{W}^T$

Pixel fidelity term: $L_{\text{PDL}}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_q$

for $j = 1$ *to* m' **do**

    (1D Wasserstein)

    Sort $\phi'_j(\mathbf{u})$ and $\phi'_j(\mathbf{v})$

    $L_{\text{PDL}}(\mathbf{u}, \mathbf{v}) += \lambda \|\phi'_j(\mathbf{u}) - \phi'_j(\mathbf{v})\|_p$

3.3 Sliced Wasserstein distance and generalizations

Bonneel et al. [37] proposed to exploit the goodness of the one-dimensional case and introduced the sliced-Wasserstein distance. By making use of the Radon transform, they show that one can integrate one-dimensional Wasserstein distances and define a distance between multi-dimensional distributions. Let $\theta \in \mathbb{S}^{d-1}$, we define the θ-projector operator as $p_\theta(x) = \theta \cdot x$. This leads to the marginal distribution $p^*_\theta I_u(s) = \int_{\mathbb{R}^d} I_u(x)\delta(s - p_\theta(x))dx$. Then, given I_u, I_v defined in $\mathbb{R}^d$, the sliced-Wasserstein distance is defined as

$$\text{SW}^p_p(I_u, I_v) = \int_{\mathbb{S}^{d-1}} W_p\left(p^*_\theta I_u, p^*_\theta I_v\right) d\theta. \tag{4}$$

In practice, the integral in (4) is approximated by Monte Carlo sampling and averaging the 1D Wasserstein distance on a set of random projections.

The sliced distance has recently received significant attention [34] since it inherits all the desirable properties of the multi-dimensional Wasserstein distance but it presents an alternative and more efficient calculation. The main drawback is that the number of necessary samples to accurately approximate the integral grows exponentially with respect to the dimension. In particular most directions will not present relevant information for discriminating the two distributions [38]. Several works propose to mitigate this problem by finding discriminative directions. In [39] authors propose to use a discriminator, similar as used in GANs, to provide good discriminant projections. [40] propose an alternative to the sliced Wasserstein distance that replace the average over the set of random projections with the maximum value. [38] introduce a

generalization of the sliced Wasserstein distance using non-linear projections.

Our proposed Projected Distribution Loss (PDL) is based on the 1D Wasserstein distance for comparing the VGG feature distributions of the generated and target image on a set of one-dimensional projections of the extracted feature maps.

4 Projected Distribution Loss

The proposed learning loss is summarized in Algorithm 1. Starting with the predicted image $\mathbf{u}$, the extracted features (e.g., VGG16-convN activation map [14]) are represented by $\Phi(\mathbf{u}) \in \mathbb{R}^{n \times m}$, where $n = h \times w$ represent the spatial dimensions, and m is the number of extracted features. Similarly, the features associated with the target image $\mathbf{v}$ are denoted by $\Phi(\mathbf{v})$.

Feature projection. We are aiming to compare the feature distributions, thus, we assume that features of image $\mathbf{u}$ are presented as set of n vectors $\boldsymbol{\phi}_i(\mathbf{u}) \in \mathbb{R}^m$ with $i = 1, \ldots, n$. To compute the distance between multidimensional distributions we can aggregate one-dimensional distances computed by projecting the features into different directions. The sliced-Wasserstein distance is computed by projecting the features into random directions on the sphere. Let $\mathbf{w}_j \in \mathbb{R}^m$ such that $\|\mathbf{w}_j\| = 1$ with $j = 1, \ldots, m'$ directions in $\mathbb{R}^m$. Then, the features are projected to generate $\phi'_{i,j}(\mathbf{u}) = \mathbf{w}_j^T \boldsymbol{\phi}_i(\mathbf{u})$. An alternative naive procedure is to compute the distances in the distributions taken from each individual feature value. This implies computing the distance between the marginal distributions. This makes sense if the features are independent.

In this work we follow the naive approach of computing and averaging the distance between the marginal distributions of the features. Nonetheless, in the experimental section we present an analysis comparing different (random) projections schemes.

4.1 Projected Distribution Training Loss

Given an image $\mathbf{u}$, its target counterpart $\mathbf{v}$, and a set of m' projections $\{\mathbf{w}_j\}$, our training loss is defined as

$$L_{\text{PDL}}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_q + \lambda \sum_{j=1}^{m'} W_p \left(\boldsymbol{\phi}'_j(\mathbf{u}), \boldsymbol{\phi}'_j(\mathbf{v}) \right), \quad (5)$$

where we have compactly denoted by $\boldsymbol{\phi}'_j = \{\phi'_{1,j}, \ldots, \phi'_{n,j}\}$ to the set of projected features at direction $\mathbf{w}_j$ for the image $\mathbf{u}$ and $\mathbf{v}$, respectively. The distance W_p is computed using (3) and the non-negative constant λ controls the effect of the distribution loss. In all the presented results we used $q = 1$ and $p = 1$. Note that this loss is fully differentiable and straightforward to implement. The impact of the regularization parameter in (5) is discussed in the next section. The proposed distribution loss combines a pixel fidelity loss term with the distribution mismatch penalization that allows to transfer details (texture, grain) without forcing them to be located in the exact same spatial position as in the target reference. This flexibility allows mitigating the *regression to the mean* problem, where, for example, slightly different high-quality images with identical feature distributions lead to very similar low resolution images. In this particular case, any point loss (pixel or L_1 on features) as the perceptual loss will learn to predict the average leading to a less realistic (blurry) image. Figure 3 shows examples comparing the proposed loss with the original perceptual loss L_{percep}. Penalizing the distance between distributions results in better preservation of fine details and overall sharpness; whereas using the L_1 norm on the features leads to over-smoothed images.

Perceptual Loss. The (original) perceptual loss [15] is computed by replacing the distribution term in 5 by the L_p distance directly computed on the extracted features,

$$L_{\text{percep}}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_q + \lambda \|\Phi(u) - \Phi(v)\|_p. \quad (6)$$

As we will show in the experimental part, measuring the distance directly between the features produces artifacts in particular in problems that are seriously ill-conditioned (such as denoising under strong noise).

Contextual Loss Our proposed framework can be closely compared with the contextual loss [18]. This loss is based on the idea that for calculating similarity between two images one should find corresponding features with minimal distances from each other. To this end, the authors define the contextual similarity based on the maximum dot product (CTXDP), or alternatively the minimum L_2 distance (CTXL2) between features. Also, it has been shown that the contextual loss is an approximation of the KL divergence [19]. Since our framework is based on the Wasserstein distance between image features, in the following section we carry out detailed comparisons with the contextual loss. We show that in most image enhancement scenarios, the proposed Wasserstein loss shows a consistent perceptual advantage over the contextual loss.

5 Experiments

In what follows we present several experiments comparing the proposed PDL distribution loss against other perceptual losses. As a baseline we train a model per application without any perceptual loss and just an L_1 fit to the pixel values ($\lambda = 0$ in (5)). We also compare to the L_1-perceptual loss given by (6) and the recently introduced contextual loss [18]. In all experiments, we tried to find the best balance between the data fitting term and the perceptual term that produces the best possible results, which is a complex and expensive task.

Since we focus on comparing different training loses, all models are trained using the same model architecture and optimization parameters. All perceptual loses are computed on `VGG16conv4` features. All models, unless otherwise stated, are trained for $n_{\text{iter}} = 10^6$ iterations, using ADAM optimizer with default parameters and a mini-batch of size 8.

We extend our analysis to five image enhancement tasks: image denoising, single image super-resolution, deblurring, JPEG artifacts removal and demosaicing. All the experiments are carried out using the standard Div2k dataset [41]. In all applications we simulated the respective degradation operator to generate training and validation data. The evaluation is done in the DIV2K validation dataset. For each application we generated $200,000$ random 256×256 crops using the div2k training images. This implies that the feature distributions are locally compared to the given image crop.

5.1 CNN Models

We focus on five image enhancement applications, and use three CNN models to showcase performance of the proposed learning loss. For denoising, JPEG artifact removal and demosaicing we use the SRN [3] model, which consists of typical convolutional and residual blocks [42], [43] at multiple spatial scales (i.e., encoder-decoder network [3], [44], [45]).

We chose this model since it proves to work well for high noise levels. For the super-resolution application we use the EDSR [2], which applies depth-to-space operation to increase the spatial resolution of the convolutional feature maps. Also, for deblurring the DsDeblur [46] model is employed in our experimentation. This model is a 3-stage encoder-decoder architecture with a selective parameter sharing scheme. In our experimentation we opt to use the default architectures proposed by respective authors.

(R3,R4) No-reference measures (NIQE, FID). We have added FID and NIQE scoresto the main tables in the paper to provide the reader with a better sense of the perceptual quality of the reconstructed images. NIQE is a no-reference quality metric, and FID is a no-reference distributional comparison metric popular in generative modeling. The updated results are given in

5.2 Quantitative and Qualitative Evaluation

To evaluate our results we report three full-reference metrics, that is, PSNR, MS-SSIM [47], and LPIPS [16], as well as the no-reference image quality scores NIQE, and the Fréchet Inception Distance (FID)[2], a no-reference distributional comparison metric. Note that in contrast to PSNR and MS-SSIM, lower LPIPS means higher similarity to the ground truth. Similarly, a lower NIQE and FID scores implies a better restoration. We should mention that this particular way of weighting the different metrics may seem somewhat arbitrary, however, we found that it sufficiently serves the purpose of finding the best model configuration.

User study. In addition to the quantitative metrics mentioned above, we also run perceptual studies with human subjects. We used the forced-choice pairwise comparison framework through Amazon Mechanical Turk, and assigned our tasks to 25 human raters with a minimum of 70% approval rating. These raters were paid 2 cents per question, and were asked to select the image with the best quality from side-by-side image crops of size 800×800. We also asked raters to only use displays with resolution 1920×1080 or higher for our experiment. Each rater answered 20 questions to compare results from two models. To avoid potential rater bias in ratings, images in each pair were randomly permuted and displayed. The reported results in the paper represent the average raters preference computed from 500 comparisons (20 raters, and 25 pairs). Since the outcome of each pairwise comparison can be interpreted as a Bernoulli random variable with probability p, the standard deviation can be expressed as $\sqrt{p(1-p)}$. This means that the standard error for each reported average is $\sqrt{p(1-p)/n}$ with $n = 500$ representing the total number of trials. Note that in the worst case scenario where $p = 0.5$, the standard error is about 0.022.

5.3 Computational cost of the PDL loss

VGG16 is a heavy network that introduces an additional cost compared to the classical pixel loss. In principle, the additional cost of our PDL loss over the perceptual loss can be attributed to the sorting operation ($O(N \log N)$). Table 1 shows the relative training speeds in global (gradient descent) steps per second for each of the tested losses. Interestingly, in practice PDL and the classical perceptual L_1 loss have very similar training times which indicates that the cost of sorting is marginal. Additionally, all the VGG-based perceptual losses run approximately 50% slower than the baseline. It is worth noting that all this extra cost is only applied at training time, and the inference complexity is the same across all loss options explored for each application.

	No-perceptual	L1	L2	CTXDP	CTXL2	PDL
steps/sec	1.00	0.57	0.54	0.54	0.53	0.57

TABLE 1
Relative training speed in global_step/sec.

5.4 Denoising under strong noise

To analyze the proposed loss we focus on the fundamental problem of image denoising. We address the particular and challenging case where the input image has been severely damaged by additive Gaussian noise (noise std. $\sigma = 100$). This allows us to get a clearer idea of the differences of the compared methods.

Projected Distribution Loss vs L_1-perceptual loss [15]. Our first and base experiment is to compare the proposed PDL distribution loss against the classical perceptual loss [15]. This is a key experiment showing the importance of comparing feature distributions over comparing extracted (aligned) features directly. Comparing training losses with multiple terms is challenging as it requires choosing a relative weight between each loss term. To accomplish this we train several models using different weights for each loss (eqs. (5) and (6)). In Table 2 we present a summary of the average quantitative performance on the validation set. As can be seen, the proposed distribution loss produces significantly better LPIPS values with similar PSNR values. The weight parameter in each loss plays a major role between the pixel distortion and the perceived quality measured by LPIPS. In Figure 4 we show some results for the different trained models. Our proposed PDL produces images with more fine texture and overall sharpness. Increasing the contribution of the L1 perceptual term in (6) does not improve the overall results but leads to images with unrealistic artifacts.

Projected Distribution Loss vs other (perceptual) losses. Table 2 shows quantitative results of training models using different perceptual losses (L_1, L_2, Contextual Loss - CTXDP and CTXL2) and no-perceptual loss at all. In Figure 5 we show image crops. The first observation is that the results with any perceptual loss are better looking than the one from the model trained without any perceptual loss. This is expected since only penalizing the difference in pixel values leads to blurry images despite having a high PSNR (as shown in Table 2). Directly penalizing the extracted features using the L_1 or L_2 distance lead to very similar quantitative and qualitative results. The same applies to the two different implementations of the contextual loss (CTXDP and CTXL2). In general, the contextual is the closest comparison to the proposed PDL distribution loss. The PDL results are in general sharper and present better edge and detail structure as shown in the building windows in Figure 5 (middle panel). In terms of LPIPS the proposed PDL loss and the contextual loss produce similar results for a similar PSNR value. Given that the contextual loss can be seen as an approximation to a distance between distributions, this confirms the superiority of comparing distributions compared to directly comparing the values of the features. We also carried out a perceptual user study and the average pairwise preferences is presented in Table 3. Overall, PDL outperforms the other models. Our method shows the smallest/largest difference with the no-perceptual/CTXL2 models. Note that performance of the two contextual models are very close to each other.

2. FID is computed using pytorch-fid, and NIQE using Matlab-niqe.

Fig. 4. Denoising under strong noise. Comparison of PDL and L1-perceptual loss. For each shown result we indicate the weight value for the L_1-perceptual/PDL loss term.

Perceptual Loss		PSNR	MS-SSIM	LPIPS	NIQE	FID
reference		-	-	-	3.166	-
input		10.49	0.416	1.289	23.171	260.61
no-perceptual		**27.14**	**0.906**	0.311	3.766	84.33
L1	0.001	27.07	0.904	0.279	3.539	61.11
	0.005	26.92	0.900	0.264	4.414	59.18
	0.010	26.81	0.896	0.268	4.456	59.89
	0.100	25.90	0.877	0.261	4.570	60.98
	0.150	25.50	0.871	0.270	4.799	62.57
L2	0.001	27.02	0.905	0.283	3.639	69.38
	0.010	26.72	0.895	0.296	3.964	64.74
	0.050	26.30	0.884	0.308	5.381	72.55
CTXDP	0.010	26.91	0.902	0.252	3.402	59.28
	0.100	26.57	0.896	0.241	3.490	58.82
	0.500	26.34	0.894	0.249	3.480	60.98
CTXL2	0.010	26.85	0.901	0.246	3.510	61.60
	0.100	26.44	0.896	0.239	3.740	57.92
	0.500	25.98	0.891	0.244	3.404	56.32
PDL	0.001	27.10	0.906	0.250	**3.226**	57.47
	0.005	26.74	0.899	0.243	3.291	55.26
	0.010	26.62	0.898	**0.233**	3.398	**54.51**
	0.015	26.48	0.895	0.238	3.275	54.95
	0.020	26.36	0.893	0.239	3.483	58.12

TABLE 2
Average performance metrics for Denoising at high noise levels $\sigma = 100$. All models were trained using the same model configuration and optimization parameters. Results with different weight for the respective perceptual loss term are presented. All perceptual looses are computed on `VGG16conv4` features. The best results are highlighted in bold.

	no-perceptual	L1	CTXDP	CTXL2	PDL
no-perceptual	-	0.43	0.31	0.29	0.23
L1	0.57	-	0.40	0.38	0.26
CTXDP	0.69	0.60	-	0.47	0.39
CTXL2	0.71	0.62	0.53	-	0.43
PDL	**0.77**	**0.74**	**0.61**	**0.57**	-

TABLE 3
Average pairwise human preference for denoising at high noise levels $\sigma = 100$. Each value represents the fraction of times the Amazon Mechanical Turk raters chose the row over the column. The average number of raters is 25. All models were trained using the same model configuration and optimization parameters. Results with the best loss weight are shown here. The best results are highlighted in bold.

5.5 Single-Image Super-resolution

We evaluated our proposed loss on single-image $4\times$ super-resolution. In addition to comparing to the different losses we already mentioned we include a comparison with SRGAN [7]. Table 4 summarizes the quantitative results of the different evaluated losses on $4\times$ super-resolution using the div2k training and validation dataset (bicubic downscaling). A selection of results is presented in Figure 6. Our proposed PDL loss leads to super-resolved images with more defined structure. SRGAN produces more fine grain details but also tends to hallucinate more content (for instance the structure on the roof). This is also reflected in the low PSNR score implying a significant pixel distortion. Results from our subjective study are shown in Table 5. We observe that on average PDL and SRGAN perform better than the other compared methods.

5.6 Deblurring

In order to evaluate the performance of the proposed loss on a deblurring task we simulated motion blur due to camera shake. Following [48], we simulated random blur kernels mimicking

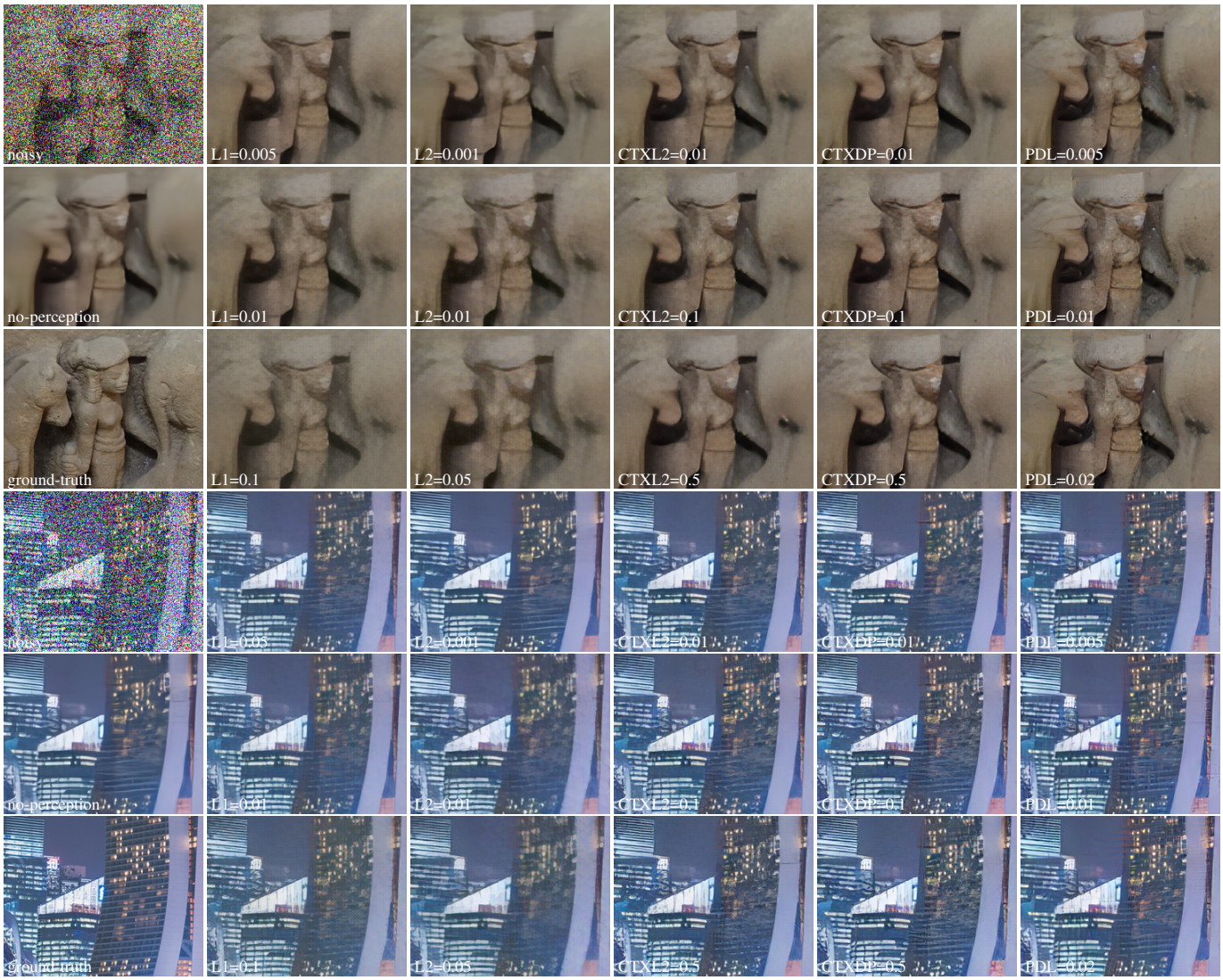


Fig. 5. Denoising at high noise levels $\sigma = 100$ example results. For each shown result we indicate the weight value for the perceptual (L_1, CTXDP, CTXL2, PDL) loss term.

Perceptual Loss	PSNR	MS-SSIM	LPIPS	NIQE	FID
reference	-	-	-	3.166	-
no-perceptual	**28.76**	**0.966**	0.274	4.762	25.30
L1	28.23	0.961	0.161	4.914	**16.37**
CTXDP	28.53	0.964	0.188	4.340	18.40
CTXL2	28.40	0.963	0.173	4.169	17.92
SRGAN	26.21	0.941	0.150	**2.975**	18.05
PDL ($\lambda = 0.003$)	28.21	0.961	0.169	3.892	18.15
PDL ($\lambda = 0.01$)	27.81	0.957	**0.145**	3.989	16.66

TABLE 4
Single image $4\times$ super-resolution. All models were trained using the same model configuration and optimization parameters. The best results are highlighted in bold.

	no-perc	L1	CTXDP	CTXL2	SRGAN	PDL
no-perceptual	-	0.43	0.31	0.30	0.28	0.24
L1	0.57	-	0.41	0.42	0.37	0.34
CTXDP	0.69	0.59	-	0.47	0.41	0.38
CTXL2	0.70	0.58	0.53	-	0.42	0.39
SRGAN	0.72	0.63	0.59	0.58	-	**0.50**
PDL	**0.76**	**0.66**	**0.62**	**0.61**	**0.50**	-

TABLE 5
Average pairwise human preference for single image 4x super-resolution. Each value represents the fraction of times the Amazon Mechanical Turk raters chose the row over the column. The average number of raters is 25. Results with the best loss weight are shown here. The best results are highlighted in bold.

camera shake blur of varying intensity (31×31 maximal support), and also added random additive noise. The added noise is also of random noise level ($\sigma \in [0, 15]$). We trained the DsDeblur [46] model using our proposed PDL loss and all the previously discussed losses. Table 6 summarizes the average quantiative performance of all tested models. As can be seen, our proposed PDL loss produces high-quality results both in term of PSNR and LPIPS. In Figure 7 we show several example results.

5.7 Other applications

We also evaluate the performance of the proposed loss in JPEG artifact removal and demosaicing under mild noise. Table 7 summarizes the quantitative results of the different evaluated losses on JPEG artifact removal (quality factor $q = 20$). Figure 8 presents some selected results on JPEG artifact removal. The PDL loss successfully manages to restore the blockiness artifacts due

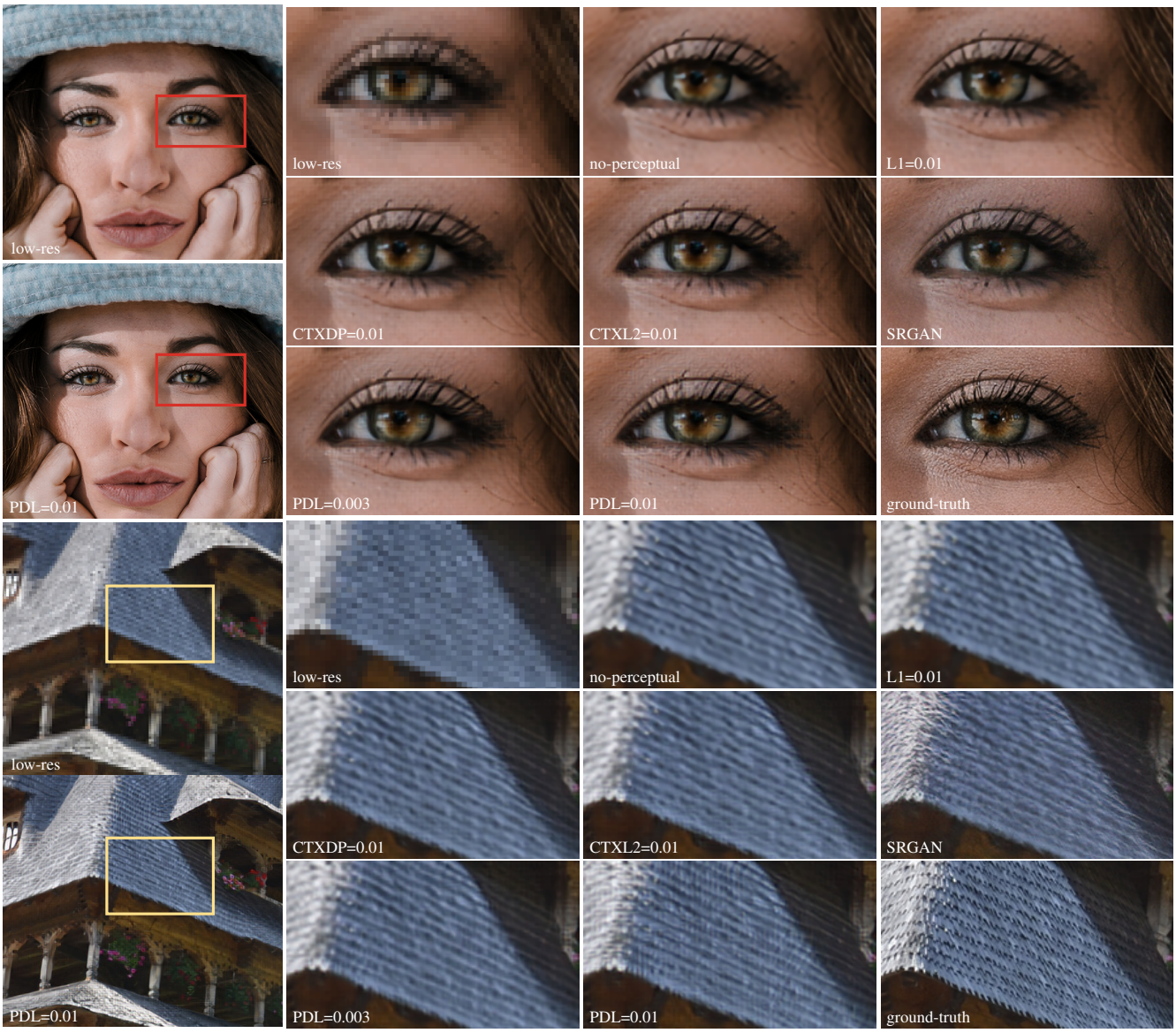


Fig. 6. Single image $\times 4$ super-resolution example results. For each shown result we indicate the weight value for the perceptual (L_1, CTXDP, CTXL2, PDL) loss term.

Perceptual Loss	PSNR	MS-SSIM	LPIPS	NIQE	FID
reference	-	-	-	3.166	-
input	27.07	0.899	0.328	7.482	45.96
no-perceptual	32.57	0.969	0.161	4.009	22.99
L1	**33.07**	**0.973**	0.126	3.768	14.29
CTXDP	32.11	0.965	0.117	3.666	15.60
CTXL2	32.79	0.971	0.095	3.635	12.71
PDL ($\lambda = 0.001$)	33.01	0.974	0.112	3.546	13.76
PDL ($\lambda = 0.005$)	32.53	0.969	**0.092**	**3.476**	**11.76**
PDL ($\lambda = 0.010$)	31.97	0.964	0.093	3.572	13.36

TABLE 6
Camera shake deblurring. Best results are highlighted in bold.

Perceptual Loss	PSNR	MS-SSIM	LPIPS	NIQE	FID
reference	-	-	-	3.166	-
input	29.54	0.953	0.208	4.267	28.24
no-perceptual	**31.77**	**0.974**	0.176	3.892	24.05
L1 ($\lambda = 0.01$)	31.70	0.973	0.144	**3.622**	17.94
L2 ($\lambda = 0.01$)	31.50	0.972	0.138	3.633	17.76
CTXDP ($\lambda = 0.01$)	31.48	0.972	0.113	3.812	16.20
CTXL2 ($\lambda = 0.01$)	31.39	0.971	0.105	4.006	16.08
PDL ($\lambda = 0.001$)	31.63	0.973	0.135	3.687	16.40
PDL ($\lambda = 0.005$)	31.24	0.970	0.106	3.848	**14.55**
PDL ($\lambda = 0.01$)	31.07	0.969	**0.103**	3.991	15.03

TABLE 7
JPEG artifact removal ($q = 20$). All models were trained using the same model configuration and optimization parameters. All perceptual looses are computed on VGG16-conv4 features.

to the sever JPEG compression, and recovers sharpness of the uncompressed image.

In the case of demosaicing, the model is trained to predict an RGB image from the Bayer noisy mosaic. In Figure 9 we present a selection of results. Although the differences are subtle, the proposed PDL lead to images with better defined structures.

5.8 Evaluation of the impact of the Feature Projection

The proposed projected distribution loss requires to define a set of projection directions. In this section we present some results for adopting different projection schemes. We tested different alterna-

Fig. 7. Deblurring example results. For each shown result we indicate the weight value for the perceptual (L_1, CTXDP, CTXL2, PDL) loss term.

Fig. 8. JPEG artifact removal (quality factor $q = 20$) example results. For each shown result we indicate the weight value for the perceptual (L_1, L_2, CTXDP, CTXL2, PDL) loss term.

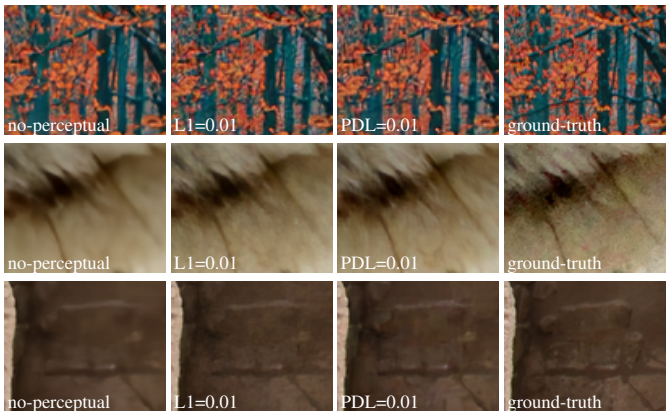


Fig. 9. Demosaicing under mild noise. For each shown result we indicate the weight value for the perceptual (L_1, PDL) loss term.

tives for computing the projected distributions. Projection on each input feature which amounts to comparing the features marginal distributions (Id), a random combination of two randomly picked activation maps (R2P), a small random perturbation of the Identity matrix (RPP), and finally a random sampling on the sphere (RSP) – which corresponds to a numerical approximation of the sliced

Projection Type	PSNR	MS=SIM	LPIPS
Id ($\lambda = 0.001$)	**27.10**	**0.906**	0.250
Id ($\lambda = 0.005$)	26.74	0.899	0.243
Id ($\lambda = 0.01$)	26.62	0.898	**0.233**
Id ($\lambda = 0.01$)	26.36	0.893	0.239
R2P-8× ($\lambda = 0.001$)	26.99	0.903	**0.255**
R2P-4× ($\lambda = 0.005$)	26.67	0.898	0.243
R2P-8× ($\lambda = 0.01$)	26.45	0.895	0.239
R2P-8× ($\lambda = 0.10$)	25.89	0.885	0.245
RPP-2× ($\lambda = 0.001$)	26.80	0.901	0.248
RPP-2× ($\lambda = 0.002$)	26.80	0.900	0.248
RPP-2× ($\lambda = 0.01$)	26.41	0.894	0.245
RSP-1× ($\lambda = 0.001$)	26.50	0.892	0.243
RSP-4× ($\lambda = 0.001$)	26.98	0.904	0.249
RSP-4× ($\lambda = 0.005$)	26.47	0.894	0.247
RSP-1× ($\lambda = 0.01$)	25.75	0.880	0.250

TABLE 8
Evaluation of the impact of the feature projection. The different projection alternatives evaluated are: comparing the original features marginal distributions independently (Id), a random combination of two randomly picked features (R2P), a small random perturbation of the original feature (RPP), and finally random projections on any possible direction on the sphere (RSP) – which corresponds to a numerical approximation of the sliced Wasserstein distance. Additionally, we evaluated the impact of the number of projections, as a factor of the original number of features (-$n\times$, with $n = 1, 2, 4, 8$). The best results are highlighted in bold.

Wasserstein distance. Additionally we evaluated the impact of the number of projections, as a factor of the original number of features. Table 8 summarizes the results. None of the tested alternatives seem to perform better than directly comparing the extracted features independently (Id). In general, all tested configurations performed similarly. We should note that VGG16 features are not normalized and therefore different features can have very different ranges. Normalizing them can lead to distortions in the geometry of the space and thus leading to artificially changing the relative importance of each feature. Recently [17] showed that not all features are equally important. Finding better features or combinations of features will be a subject of our future work.

6 Conclusion

We have presented an alternative perceptual loss that can be used to train any image restoration model. The proposed projected distribution loss is straightforward to implement being based on comparing 1D marginal distributions. Our loss leads to superior or similar results on all the five tested image restoration applications. In this work we focused on comparing CNN features that have been shown to capture relevant characteristics to visual perception. As a future work we would like to investigate alternative schemes to project the multidimensional feature space into lower dimensions where the proposed distribution loss can be applied.

References

[1] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016. 1

[2] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144. 1, 6

[3] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8174–8182. 1, 5

[4] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3291–3300. 1

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680. 1, 3

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017. 1, 3

[7] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690. 1, 3, 7

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. 1, 3

[9] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *IEEE conference on computer vision and pattern recognition*, 2018, pp. 8183–8192. 1, 3

[10] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *IEEE International Conference on Computer Vision*, 2019, pp. 8878–8887. 1, 3

[11] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *International Conference on Machine Learning*, 2017, pp. 224–232. 1

[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242. 1

[13] J. P. Cohen, M. Luck, and S. Honari, "Distribution matching losses can hallucinate features in medical image translation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2018, pp. 529–536. 1, 3

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 1, 2, 3, 5

[15] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711. 1, 2, 5, 6

[16] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595. 1, 2, 6

[17] T. Tariq, O. T. Tursun, M. Kim, and P. Didyk, "Why are deep representations good perceptual quality features?" in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 445–461. 1, 2, 11

[18] R. Mechrez, I. Talmi, and L. Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 768–783. 1, 2, 5

[19] R. Mechrez, I. Talmi, F. Shama, and L. Zelnik-Manor, "Maintaining natural image statistics with the contextual loss," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 427–443. 1, 2, 5

[20] X. Zhang, Q. Chen, R. Ng, and V. Koltun, "Zoom to learn, learn to zoom," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3762–3770. 1, 2

[21] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423. 2

[22] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," in *26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17. AAAI Press, 2017, p. 2230–2236. 2

[23] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Advances in neural information processing systems*, 2017, pp. 386–396. 2

[24] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338. 2, 3

[25] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019. 2, 3

[26] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in neural information processing systems*, 2013, pp. 2292–2300. 2

[27] J. Rabin, G. Peyré, J. Delon, and M. Bernot, "Wasserstein barycenter and its application to texture mixing," in *International Conference on Scale*

Space and Variational Methods in Computer Vision. Springer, 2011, pp. 435–446. 2, 3

[28] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015. 2

[29] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017. 2

[30] H. Talebi and P. Milanfar, "Learned perceptual image enhancement," in *2018 IEEE international conference on computational photography (ICCP)*. IEEE, 2018, pp. 1–13. 2

[31] ——, "Nima: Neural image assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, 2018. 2

[32] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6228–6237. 2

[33] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio, "Learning with a wasserstein loss," in *Advances in neural information processing systems*, 2015, pp. 2053–2061. 3

[34] J. Wu, Z. Huang, D. Acharya, W. Li, J. Thoma, D. P. Paudel, and L. V. Gool, "Sliced wasserstein generative models," in *IEEE conference on computer vision and pattern recognition*, 2019, pp. 3713–3722. 3, 4

[35] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 203–12 213. 3

[36] M.-M. Deza and E. Deza, *Dictionary of distances*. Elsevier, 2006. 3

[37] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon wasserstein barycenters of measures," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 22–45, 2015. 4

[38] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, "Generalized sliced wasserstein distances," in *Advances in Neural Information Processing Systems*, 2019, pp. 261–272. 4

[39] I. Deshpande, Z. Zhang, and A. G. Schwing, "Generative modeling using the sliced wasserstein distance," in *IEEE conference on computer vision and pattern recognition*, 2018, pp. 3483–3491. 4

[40] I. Deshpande, Y.-T. Hu, R. Sun, A. Pyrros, N. Siddiqui, S. Koyejo, Z. Zhao, D. Forsyth, and A. G. Schwing, "Max-sliced wasserstein distance and its use for gans," in *IEEE conference on computer vision and pattern recognition*, 2019, pp. 10 648–10 656. 4

[41] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135. 5

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 5

[43] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3883–3891. 5

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. 5

[45] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810. 5

[46] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3848–3856. 6, 8

[47] 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*, vol. 2. Ieee, 2003, pp. 1398–1402. 6

[48] M. Delbracio and G. Sapiro, "Burst deblurring: Removing camera shake through fourier burst accumulation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2385–2393. 7

Mauricio Delbracio is a research scientist at Google Research. Before joining Google in 2019, he was an Assistant Professor at the Department of Electrical Engineering, Universidad de la República (UdelaR), Uruguay. From 2013 to 2016 he was a postdoctoral researcher with the ECE Department at Duke University. He received the B.Sc degree in electrical engineering from UdelaR, Montevideo, in 2006, and the M.Sc. and Ph.D. degrees in applied mathematics from École Normale Supérieure de Cachan (ENS-Cachan), France, in 2009 and 2013 respectively. His current research focuses on algorithms, data analysis and applications of machine learning to image and signal processing. In 2016 he was awarded the Early Career Prize from the Society for Industrial and Applied Mathematics (SIAM) Activity Group on Imaging Science in 2016 for his important contributions to image processing.

Hossein Talebi received the B.S. and M.S. degrees in electrical engineering from the Isfahan University of Technology, Iran, and the Ph.D. degree in electrical engineering from the University of California at Santa Cruz, Santa Cruz, CA, USA. Since 2015, he has been with Google Research, Mountain View, CA, where he works on computational imaging, image processing and machine learning problems.

Peyman Milanfar is a Principal Scientist / Director at Google Research, where he leads the Computational Imaging team. Prior to this, he was a Professor of Electrical Engineering at UC Santa Cruz from 1999-2014. He was Associate Dean for Research at the School of Engineering from 2010-12. From 2012-2014 he was on leave at Google-x, where he helped develop the imaging pipeline for Google Glass. Most recently, Peyman's team at Google developed the digital zoom pipeline for the Pixel phones, which includes the multi-frame super-resolution (Super Res Zoom) pipeline (blog, and project website), and the RAISR upscaling algorithm. In addition, the Night Sight mode on Pixel 3 uses our Super Res Zoom technology to merge images (whether you zoom or not) for vivid shots in low light, including astrophotography. Peyman received his undergraduate education in electrical engineering and mathematics from the University of California, Berkeley, and the MS and PhD degrees in electrical engineering from the Massachusetts Institute of Technology. He holds 15 patents, several of which are commercially licensed. He founded MotionDSP, which was acquired by Cubic Inc. (NYSE:CUB). Peyman has been keynote speaker at numerous technical conferences including Picture Coding Symposium (PCS), SIAM Imaging Sciences, SPIE, and the International Conference on Multimedia (ICME). Along with his students, he has won several best paper awards from the IEEE Signal Processing Society. He is a Distinguished Lecturer of the IEEE Signal Processing Society, and a Fellow of the IEEE "for contributions to inverse problems and super-resolution in imaging."