

Maintaining Natural Image Statistics with the Contextual Loss

Roey Mechrez*, Itamar Talmi*, Firas Shama, Lihi Zelnik-Manor

The Technion - Israel

{roey@campus,titamar@campus,shfiras@campus,lihi@ee}.technion.ac.il

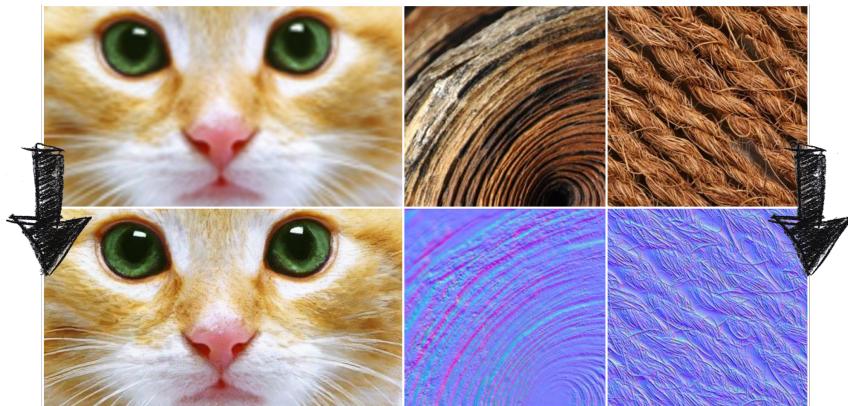


Fig. 1. We demonstrate the advantages of using a statistical loss for training a CNN in: (left) single-image super resolution, and, (right) surface normal estimation. Our approach is easy to train and yields networks that generate images (or surfaces) that exhibit natural internal statistics.

Abstract. Maintaining natural image statistics is a crucial factor in restoration and generation of realistic looking images. When training CNNs, photorealism is usually attempted by adversarial training (GAN), that pushes the output images to lie on the manifold of natural images. GANs are very powerful, but not perfect. They are hard to train and the results still often suffer from artifacts. In this paper we propose a complementary approach, that could be applied with or without GAN, whose goal is to train a feed-forward CNN to maintain natural internal statistics. We look explicitly at the distribution of features in an image and train the network to generate images with natural feature distributions. Our approach reduces by orders of magnitude the number of images required for training and achieves state-of-the-art results on both single-image super-resolution, and high-resolution surface normal estimation.

* indicate authors contributed equally

1 Introduction

“Facts are stubborn things, but statistics are pliable.” — Mark Twain

Maintaining natural image statistics has been known for years as a key factor in the generation of natural looking images [1,2,3,4,5]. With the rise of CNNs, the utilization of explicit image priors was replaced by Generative Adversarial Networks (GAN) [6], where a corpus of images is used to train a network to generate images with natural characteristics, e.g., [7,8,9,10]. Despite the use of GANs within many different pipelines, results still sometimes suffer from artifacts, balanced training is difficult to achieve and depends on the architecture [8,11].

Well before the CNN era, natural image statistics were obtained by utilizing priors on the likelihood of the *patches* of the generated images [2,5,12]. Zoran and Weiss [5] showed that such statistical approaches, that harness priors on patches, lead in general to restoration of more natural looking images. A similar concept is in the heart of sparse-coding where a dictionary of visual code-words is used to constrain the generated patches [13,14]. The dictionary can be thought as a prior on the space of plausible image patches. A related approach is to constrain the generated image patches to the space of patches specific to the image to be restored [15,16,17,18]. In this paper we want to build on these ideas in order to answer the following question: *Can we train a CNN to generate images that exhibit natural statistics?*

The approach we propose is a simple modification to the common practice. As typically done, we as well train CNNs on pairs of source-target images by minimizing an objective that measures the similarity between the generated image and the target. We extend on the common practice by proposing to use an objective that compares the *feature distributions* rather than just comparing the appearance. We show that by doing this the network learns to generate more natural looking images.

A key question is what makes a suitable objective for comparing distributions of features. The common divergence measures, such as Kullback-Leibler (KL), Earth-Movers-Distance and χ^2 , all require estimating the distribution of features, which is typically done via Multivariate Kernel-Density-Estimation (MKDE). Since in our case the features are very high dimensional, and since we want a differentiable loss that can be computed efficiently, MKDE is not an option. Hence, we propose instead an approximation to KL that is both simple to compute and tractable. As it turns out, our approximation coincides with the recently proposed *Contextual loss* [19], that was designed for comparing images that are not spatially aligned. Since the Contextual is actually an approximation to KL, it could be useful as an objective also in applications where the images are aligned, and the goal is to generate natural looking images.

Since all we propose is utilizing the Contextual loss during training, our approach is generic and can be adopted within many architectures and pipelines. In particular, it can be used in concert with GAN training. Hence, we chose super-resolution as a first test-case, where methods based on GANs are the current state-of-the-art. We show empirically, that using our statistical approach

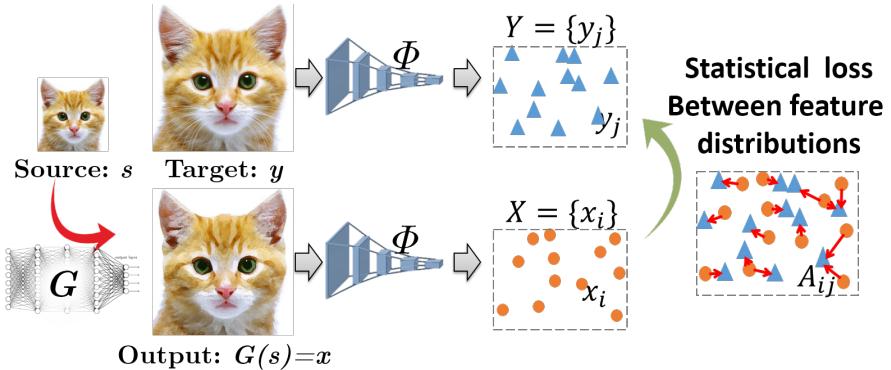


Fig. 2. Training with a statistical loss: To train a generator network G we compare the output image $G(s) \equiv x$ with the corresponding target image y via a statistical loss — the Contextual loss [19] — that compares their feature distributions.

with GAN training outperforms previous methods, yielding images that are *perceptually* more realistic, while reducing the number of required training images by orders of magnitude.

Our second test-case further proves the generality of this approach to data that is not images, and shows the strength of the statistical approach without GAN. Maintaining natural internal statistics has been shown to be an important property also in the estimation of 3D surfaces [15,20,21]. Hence, we present experiments on surface normal estimation, where the network’s input is a high-resolution image and its output is a map of normals. We successfully generate normal-maps that are more accurate than previous methods.

To summarize, the contributions we present are three-fold:

1. We show that the Contextual loss [19] can be viewed as an approximation to KL divergence. This makes it suitable for reconstruction problems.
2. We show that training with a statistical loss yields networks that maintain the natural internal statistics of images. This approach is easy to train and reduces significantly the training set size.
3. We present state-of-the-art results on both perceptual super-resolution and high-resolution surface normal estimation.

2 Training CNNs to Match Image Distributions

2.1 Setup

Our approach is very simple, as depicted in Figure 2. To train a generator network G we use pairs of source s and target y images. The network outputs an image $G(s) \equiv x$, that should be as similar as possible to the target y . To

measure the similarity we extract from both x and y dense features $X = \{x_i\}$ and $Y = \{y_j\}$, respectively, e.g., by using pretrained network or vectorized RGB patches. We denote by P_Y and P_X the probability distribution functions over Y and X , respectively. When the patch set X correctly models Y then the underlying probabilities P_Y and P_X are equal. Hence, we need to train G by minimizing an objective that measures the divergence between P_X and P_Y .

2.2 Review of the KL-divergence

There are many common measures for the divergence between two distributions, e.g., χ^2 , Kullback-Leibler (KL), and Earth Movers Distance (EMD). We opted for the KL-divergence in this paper.

The KL-divergence between the two densities P_X and P_Y is defined as:

$$KL(P_X || P_Y) = \int P_X \log \frac{P_X}{P_Y} \quad (1)$$

It computes the expectation of the logarithmic difference between the probabilities P_X and P_Y , where the expectation is taken using the probabilities P_X . We can rewrite this formula in terms of expectation:

$$KL(P_X || P_Y) = E[\log P_X - \log P_Y] \quad (2)$$

This requires approximating the parametrized densities P_X, P_Y from the feature sets $X = \{x_i\}$ and $Y = \{y_j\}$. The most common method for doing this is to use Multivariate Kernel Density Estimation (MKDE), that estimates the probabilities $P_X(p_k)$ and $P_Y(p_k)$ as:

$$P_X(p_k) = \sum_{x_i \in X} K_H(p_k, x_i) \quad P_Y(p_k) = \sum_{y_j \in Y} K_H(p_k, y_j) \quad (3)$$

Here $K_H(p_k, \xi)$ is an affinity measure (the kernel) between some point p_k and the samples ξ , typically taken as standard multivariate normal kernel $K_H(z) = (2\pi)^{-d/2} |H|^{-1/2} \exp(-\frac{1}{2} z^T H z)$, with $z = dist(p_k, \xi)$, d is the dimension and H is a $d \times d$ bandwidth matrix. We can now write the expectation of the KL-Divergence over the MKDE with uniform sample grid as follows:

$$KL(P_X || P_Y) = \frac{1}{N} \sum_{p_k} [\log P_X(p_k) - \log P_Y(p_k)] \quad (4)$$

A common simplification that we adopt is to compute the MKDE not over a regular grid of $\{p_k\}$ but rather on the samples $\{x_i\}$ directly, i.e., we set $p_k = x_k$, which implies $K_H(p_k, x_i) = K_H(x_k, x_i)$ and $K_H(p_k, y_j) = K_H(x_k, y_j)$. Putting this together with Eq. (3) into Eq. (4) yields:

$$KL(P_X || P_Y) = \frac{1}{N} \sum_k [\log \sum_{x_i \in X} K_H(x_k, x_i) - \log \sum_{y_j \in Y} K_H(x_k, y_j)] \quad (5)$$

To use Eq. (5) as a loss function we need to choose a kernel K_H . The most common choice of a standard multivariate normal kernel requires setting the bandwidth matrix H , which is non-trivial. Multivariate KDE is known to be sensitive to the optimal selection of the bandwidth matrix and the existing solutions for tuning it require solving an optimization problem which is not possible as part of a training process. Hence, we next propose an approximation, which is both practical and tractable.

2.3 Approximating the KL-divergence with the Contextual loss

Our approximation is based on one further observation. It is insufficient to produce an image with the same distribution of features, but rather we want additionally that the samples will be similar. That is, we ask each point $x_i \in X$ to be close to a specific $y_j \in Y$.

To achieve this, while assuming that the number of samples is large, we set the MKDE kernel such that it approximates a delta function. When the kernel is a delta, the first log term of Eq. (5) becomes a constant since:

$$K_H(x_k, x_i) = \begin{cases} \approx 1 & \text{if } k = i \\ \approx 0 & \text{otherwise} \end{cases} \quad (6)$$

The kernel in the second log term of Eq. (5) becomes:

$$K_H(x_k, y_j) = \begin{cases} \approx 1 & \text{if } dist(x_k, y_j) \ll dist(x_k, y_l) \forall l \neq j \\ \approx 0 & \text{otherwise} \end{cases} \quad (7)$$

We can thus simplify the objective of Eq. (5):

$$E(x, y) = -\log \left(\frac{1}{N} \sum_j \max_i A_{ij} \right) \quad (8)$$

where we denote $A_{ij} = K_H(x_k, y_j)$. Next, we suggest two alternatives for the kernel A_{ij} , and show that one implies that the objective of Eq. (8) is equivalent to the Chamfer Distance [22], while the other implies it is equivalent to the Contextual loss of [19].

The Contextual Loss: As it turns out, the objective of Eq. (8) is identical to the Contextual loss recently proposed by [19]. Furthermore, they set the kernel A_{ij} to be close to a delta function, such that it fits Eq. (7). First the Cosine (or L2) distances $dist(x_i, y_j)$ are computed between all pairs x_i, y_j . The distances are then normalized: $\tilde{d}_{ij} = dist(x_i, y_j) / (\min_k dist(x_i, y_k) + \epsilon)$ (with $\epsilon = 1e-5$), and finally the pairwise affinities $A_{ij} \in [0, 1]$ are defined as:

$$A_{ij} = \frac{\exp \left(1 - \tilde{d}_{ij}/h \right)}{\sum_l \exp \left(1 - \tilde{d}_{il}/h \right)} = \begin{cases} \approx 1 & \text{if } \tilde{d}_{ij} \ll \tilde{d}_{il} \quad \forall l \neq j \\ \approx 0 & \text{otherwise} \end{cases} \quad (9)$$

where $h > 0$ is a scalar bandwidth parameter that we fix to $h = 0.1$, as proposed in [19]. When using these affinities our objective equals the Contextual loss of [19] and we denote $E(x, y) = \mathcal{L}_{CX}(x, y)$.

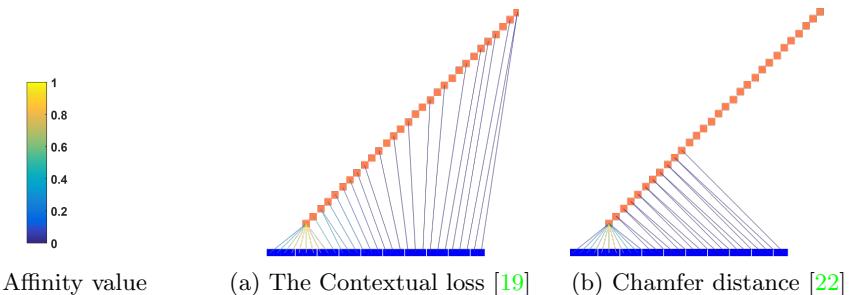


Fig. 3. The Contextual loss vs. Chamfer Distance: We demonstrate via a 2D example the difference between two approximations to the KL-divergence (a) The Contextual loss and (b) Chamfer Distance. Point sets Y and X are marked with blue and orange squares respectively. The colored lines connect each y_j with x_i with the largest affinity. The KL approximation of Eq. (8) sums over these affinities. It can be seen that the normalized affinities used by the Contextual loss lead to more diverse and meaningful matches between Y and X .

The Chamfer Distance A simpler way to set A_{ij} is to take a Gaussian kernel with a fixed $H = I$ s.t. $A_{ij} = \exp(-\text{dist}(x_i, y_j))$. This choice implies that minimizing Eq. (8) is equivalent to minimizing the asymmetric Chamfer Distance [22] between X, Y defined as

$$\text{CD}(X, Y) = \frac{1}{|X|} \sum_i \min_j \text{dist}(x_i, y_j) \quad (10)$$

CD has been previously used mainly for shape retrieval [23,24] were the points are in \mathbb{R}^3 . For each point in set X , CD finds the nearest point in the set Y and minimizes the sum of these distances. A downside of this choice for the kernel A_{ij} is that it does not satisfy the requirement in Eq. (7).

The Contextual Loss vs. Chamfer Distance To provide intuition on the difference between the two choices of affinity functions we present in Figure 3 an illustration in 2D. Recall, that both the Contextual loss \mathcal{L}_{CX} and the Chamfer distance CD find for each point in Y a single match in X , however, these matches are computed differently. CD selects the closest point, hence, we could get that multiple points in Y are matched to the same few points in X . Differently, \mathcal{L}_{CX} computes normalized affinities, that consider the distances between every point x_i to all the points $y_j \in Y$. Therefore, it results in more diverse matches between the two sets of points, and provides a better measure of similarity between the two distributions. Additional example is presented in the supplementary.

Training with \mathcal{L}_{CX} guides the network to match between the two point sets, and as a result the underlying distributions become closer. In contrast, training with CD, does not allow the two sets to get close. Indeed, we found empirically that training with CD does not converge, hence, we excluded it from our empirical reports.

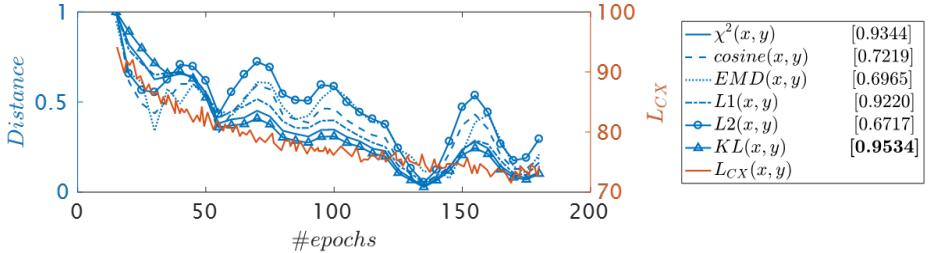


Fig. 4. Minimizing the KL-divergence: The curves represents seven measures of dissimilarity between the patch distribution of the target image y and that of the generated image x , during training a super-resolution network (details in supplementary) and using \mathcal{L}_{CX} as the objective. The correlation between \mathcal{L}_{CX} and all other measures, reported in square brackets, show highest correlation to the KL-divergence. It is evident that training with the Contextual loss results in minimizing also the KL-divergence.

3 Empirical analysis

The Contextual loss has been proposed in [19] for measuring similarity between non-aligned images. It has therefore been used for applications such as style transfer, where the generated image and the target style image are very different. In the current study we assert that the Contextual loss can be viewed as a statistical loss between the distributions of features. We further assert that using such a loss during training would lead to generating images with realistic characteristics. This would make it a good candidate also for tasks where the training image pairs are aligned.

To support these claims we present in this section two experiments, and in the next section two real applications. The first experiment shows that minimizing the Contextual loss during training indeed implies also minimization of the KL-divergence. The second experiment evaluates the relation between the Contextual loss and human perception of image quality.

Empirical analysis of the approximation Since we are proposing to use the Contextual loss as an approximation to the KL-divergence, we next show empirically, that minimizing it during training also minimizes the KL-divergence.

To do this we chose a simplified super-resolution setup, based on SRResNet [9], and trained it with the Contextual loss as the objective. The details on the setup are provided in the supplementary. We compute during the training the Contextual loss, the KL-divergence, as well as five other common dissimilarity measures. To compute the KL-divergence, EMD and χ^2 , we need to approximate the density of each image. As discussed in Section 2, it is not clear how the multivariate solution to KDE can be smoothly used here, therefore, instead, we generate a random projection of all 5×5 patches onto 2D and fit them using KDE

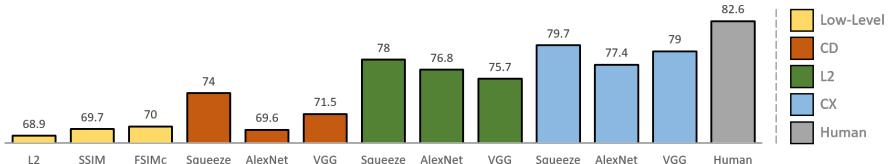


Fig. 5. How perceptual is the Contextual loss? Test on the 2AFC [26] data set with traditional and CNN-based distortions. We compare between low-level metrics (such as SSIM) and off-the-shelf deep networks trained with L2 distance, Chamfer distance (CD) and the Contextual loss (\mathcal{L}_{CX}). \mathcal{L}_{CX} yields a performance boost across all three networks with the largest gain obtained for VGG – the most common loss network (i.e., perceptual loss).

with a Gaussian kernel in 2D [25]. This was repeated for 100 random projections and the scores were averaged over all projections (examples of projections are shown in the supplementary).

Figure 4 presents the values of all seven measures during the iterations of the training. It can be seen that all of them are minimized during the iterations. The KL-divergence minimization curve is the one most similar to that of the Contextual loss, suggesting that the Contextual loss forms a reasonable approximation.

Evaluating on Human Perceptual Judgment Our ultimate goal is to train networks that generate images with high perceptual quality. The underlying hypothesis behind our approach is that training with an objective that maintains natural statistics will lead to this goal. To asses this hypothesis we repeated the evaluation procedure proposed in [26], for assessing the correlation between human judgment of similarity and loss functions based on deep features.

In [26] it was suggested to compute the similarity between two images by comparing their corresponding deep embeddings. For each image they obtained a deep embedding via a pre-trained network, normalized the activations and then computed the L_2 distance. This was then averaged across spatial dimension and across all layers. We adopted the same procedure while replacing the L_2 distance with the Contextual loss approximation to KL.

Our findings, as reported in Figure 5 (the complete table is provided in the supplementary material), show the benefits of our proposed approach. The Contextual loss between deep features is more closely correlated with human judgment than L_2 or Chamfer Distance between the same features. All of these perceptual measures are preferable over low-level similarity measures such as SSIM [27].

4 Applications

In this section we present two applications: single-image super-resolution, and high-resolution surface normal estimation. We chose the first to highlight the advantage of using our approach in concert with GAN. The second was selected since its output is not an image, but rather a surface of normals. This shows the generic nature of our approach to other domains apart from image generation, where GANs are not being used.

4.1 Single-Image Super-Resolution

To asses the contribution of our suggested framework for image restoration we experiment on single-image super-resolution. To place our efforts in context we start by briefly reviewing some of the recent works on super-resolution. A more comprehensive overview of the current trends can be found in [28].

Recent solutions based on CNNs can be categorized into two groups. The first group relies on the L_2 or L_1 losses [9,29,30], which lead to high PSNR and SSIM [27] at the price of low perceptual quality, as was recently shown in [31,26]. The second group of works aim at high perceptual quality. This is done by adopting perceptual loss functions [32], sometimes in combination with GAN [9] or by adding the Gram loss [33], which nicely captures textures [34].

Proposed solution: Our main goal is to generate natural looking images, with natural internal statistics. At the same time, we do not want the structural similarity to be overly low (the trade-off between the two is nicely analyzed in [31,26]). Therefore, we propose an objective that considers both, with higher importance to perceptual quality. Specifically, we integrate three loss terms: (i) The Contextual loss – to make sure that the internal statistics of the generated image are similar to those of the ground-truth high-resolution image. (ii) The L_2 loss, computed at **low** resolution – to drive the generated image to share the spatial structure of the target image. (iii) Finally, following [9] we add an adversarial term, which helps in pushing the generated image to look “real”.

Given a low-resolution image s and a target high-resolution image y , our objective for training the network G is:

$$\mathcal{L}(G) = \lambda_{\text{CX}} \cdot \mathcal{L}_{\text{CX}}(G(s), y) + \lambda_{L2} \cdot \|G(s)^{\text{LF}} - y^{\text{LF}}\|_2 + \lambda_{\text{GAN}} \cdot \mathcal{L}_{\text{GAN}}(G(s)) \quad (11)$$

where in all our experiments $\lambda_{\text{CX}} = 0.1$, $\lambda_{\text{GAN}} = 1e-3$, and $\lambda_{L2} = 10$. The images $G(s)^{\text{LF}}$, y^{LF} are low-frequencies obtained by convolution with a Gaussian kernel of width 21×21 and $\sigma = 3$. For the Contextual loss feature extraction we used layer *conv3_4* of VGG19 [35].

Implementation details: We adopt the SRGAN architecture [9]¹ and replace only the objective. We train it on just 800 images from the DIV2K dataset [36], for 1500 epochs. Our network is initialized by first training using only the L_2 loss for 100 epochs.

¹ We used the implementation in <https://github.com/tensorlayer/SRGAN>

Method	Loss function	Distortion SSIM [27]	Perceptual NRQM [38]	# Training images
Johnson[32]	L_P	0.631	7.800	10K
SRGAN[9]	$\mathcal{L}_{GAN} + L_P$	0.640	8.705	300K
EnhanceNet[34]	$\mathcal{L}_{GAN} + L_P + L_T$	0.624	8.719	200K
Ours full	$\mathcal{L}_{GAN} + \mathcal{L}_{CX} + L_2^{LF}$	0.643	8.800	800
Ours w/o L_2^{LF}	$\mathcal{L}_{GAN} + \mathcal{L}_{CX}$	0.67	8.53	800
Ours w/o \mathcal{L}_{CX}	$\mathcal{L}_{GAN} + L_2^{LF}$	0.510	8.411	800
SRGAN-MSE*	$\mathcal{L}_{GAN} + L_2$	0.643	8.4	800

*our reimplementation

Table 1. Super-resolution results: The table presents the mean scores obtained by our method and three others on BSD100 [37]. SSIM [27] measures structural similarity to the ground-truth, while NRQM [38] measures no-reference perceptual quality. Our approach provides an improvement on both scores, even though it required orders of magnitude fewer images for training. The table further provides an ablation test of the loss function to show that \mathcal{L}_{CX} is a key ingredient in the quality of the results. L_P is the perceptual loss [32], L_T is the Gram loss [33], and both use VGG19 features.

Evaluation: Empirical evaluation was performed on the BSD100 dataset [37]. As suggested in [31] we compute both structural similarity (SSIM [27]) to the ground-truth and perceptual quality (NRQM [38]). The “ideal” algorithm will have both scores high.

Table 1 compares our method with three recent solutions whose goal is high perceptual quality. It can be seen that our approach outperforms the state-of-the-art on both evaluation measures. This is especially satisfactory as we needed only 800 images for training, while previous methods had to train on tens or even hundreds of thousands of images. Note, that the values of the perceptual measure NRQM are not normalized and small changes are actually quite significant. For example the gap between [9] and [34] is only 0.014 yet visual inspection shows a significant difference. The gap between our results and [34] is 0.08, i.e., almost 6 times bigger, and visually it is significant.

Figure 6 further presents a few qualitative results, that highlight the gap between our approach and previous ones. Both SRGAN [9] and EnhanceNet [34] rely on adversarial training (GAN) in order to achieve photo-realistic results. This tends to over-generate high-frequency details, which make the image look sharp, however, often these high-frequency components do not match those of the target image. The Contextual loss, when used in concert with GAN, reduces these artifacts, and results in natural looking image patches.

An interesting observation is that we achieve high perceptual quality while using for the Contextual loss features from a mid-layer of VGG19, namely *conv3_4*. This is in contrast to the reports in [9] that had to use for SRGAN the high-layer *conv5_4* for the perceptual loss (and failed when using low-layer such as *conv2_2*). Similarly, EnhanceNet required a mixture of *pool2* and *pool5*.

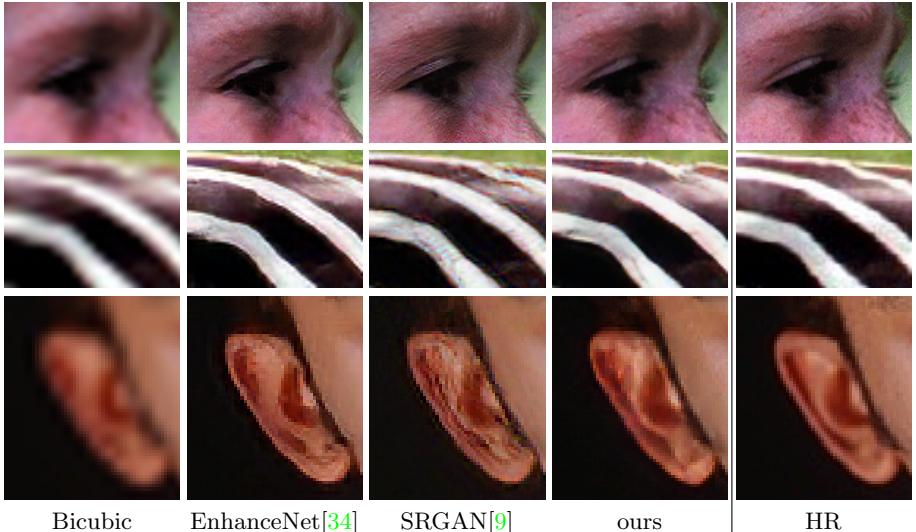


Fig. 6. Super-resolution qualitative evaluation: Training with GAN and a perceptual loss sometimes adds undesired textures, e.g., SRGAN added wrinkles to the girl’s eyelid, unnatural colors to the zebra stripes, and arbitrary patterns to the ear. Our solution replaces the perceptual loss with the Contextual loss, thus getting rid of the unnatural patterns.

4.2 High-resolution Surface Normal Estimation

The framework we propose is by no means limited to networks that generate images. It is a generic approach that could be useful for training networks for other tasks, where the natural statistics of the target should be exhibited in the network’s output. To support this, we present a solution to the problem of surface normal estimation – an essential problem in computer vision, widely used for scene understanding and new-view synthesis.

Data: The task we pose is to estimate the underlying normal map from a single monocular color image. Although this problem is ill-posed, recent CNN based approaches achieve satisfactory results [40,41,42,43] on the NYU-v2 dataset [44]. Thanks to its size, quality and variety, NYU-v2 is intensively used as a test-bed for predicting depth, normals, segmentation etc. However, due to the acquisition protocol, its data does not include the fine details of the scene and misses the underlying high frequency information, hence, it does not provide a detailed representation of natural surfaces.

Therefore, we built a new dataset of images of surfaces and their corresponding normal maps, where fine details play a major role in defining the surface structure. Examples are shown in Figure 7. Our dataset is based on 182 different textures and their respective normal maps that were collected from the Internet², originally offered for usages of realistic interior home design, gaming,

² www.polygon.com and www.textures.com

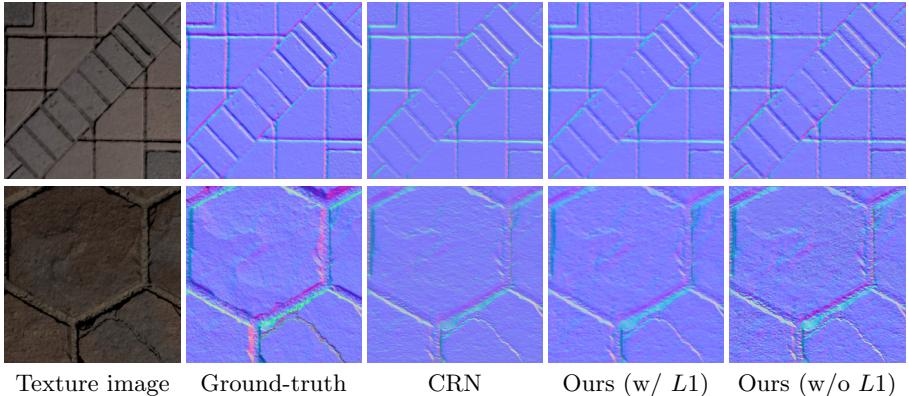


Fig. 7. Estimating surface normals: A visual comparison of our method with CRN [39]. CRN trains with $L1$ as loss, which leads to coarse reconstruction that misses many fine details. Conversely, using our objective, especially without $L1$, leads to more natural looking surfaces, with delicate texture details. Interestingly, the results without $L1$ look visually better, even though quantitatively they are inferior, as shown in Table 2. We believe this is due to the usage of structural evaluation measures rather than perceptual ones.

arts, etc. For each texture we obtained a high resolution color image (1024×1024) and a corresponding normal-map of a surface such that its underlying plane normal points towards the camera (see Figure 7). Such image-normals pairs lack the effect of lighting, which plays an essential role in normal estimation. Hence, we used Blender³, a 3D renderer, to simulate each texture under 10 different point-light locations. This resulted in a total of 1820 pairs of image-normals. The textures were split into 90% for training and 10% for testing, such that the test set includes all the 10 rendered pairs of each included texture.

The collection offers a variety of materials including wood, stone, fabric, steel, sand, etc., with multiple patterns, colors and roughness levels, that capture the appearance of real-life surfaces. While some textures are synthetic, they look realistic and exhibit imperfections of real surfaces, which are translated into the fine-details of both the color image as well as its normal map.

Proposed solution: We propose using an objective based on a combination of three loss terms: (i) The Contextual loss – to make sure that the internal statistics of the generated normal map match those of the target normal map. (ii) The $L2$ loss, computed at **low** resolution, and (iii) The $L1$ loss. Both drive the generated normal map to share the spatial layout of the target map. Our overall objective is:

$$\mathcal{L}(G) = \lambda_{CX} \cdot \mathcal{L}_{CX}(G(s), y) + \lambda_{L2} \cdot \|G(s)^{LF} - y^{LF}\|_2 + \lambda_{L1} \cdot \|G(s) - y\|_1 \quad (12)$$

³ www.blender.org

where, $\lambda_{CX} = 1$, and $\lambda_{L2} = 0.1$. The normal-maps $G(s)^{LF}$, y^{LF} are low-frequencies obtained by convolution with a Gaussian kernel of width 21×21 and $\sigma = 3$. We tested with both $\lambda_{L1} = 1$ and $\lambda_{L1} = 0$, which removes the third term.

Implementation details: We chose as architecture the Cascaded Refinement Network (CRN) [39] originally suggested for label-to-image and was shown to yield great results in a variety of other tasks [19]. For the contextual loss we took as features 5×5 patches of the normal map (extracted with stride 2) and layers $conv1_2$, $conv2_2$ of VGG19. In our implementation we reduced memory consumption by random sampling of all three layers into 65×65 features.

Evaluation: Table 2 compares our results with previous solutions. We compare to the recently proposed PixelNet [40], that presented state-of-the-art results on NYU-v2. Since PixelNet was trained on NYU-v2, which lacks fine details, we also tried fine-tuning it on our dataset. In addition, we present results obtained with CRN [39]. While the original CRN was trained with both $L1$ and the perceptual loss [32], this combination provided poor results on normal estimation. Hence, we excluded the perceptual loss and report results with only $L1$ as the loss, or when using our objective of Eq. (12). It can be seen that CRN trained with our objective (with $L1$) leads to the best quantitative scores.

Method	Mean (°)↓	Median (°)↓	RMSE (°)↓	11.25° (%)↑	22.5° (%)↑	30° (%)↑
PixelNet [40]	25.96	23.76	30.01	22.54	50.61	65.09
PixelNet [40]+FineTune	14.27	12.44	16.64	51.20	85.13	91.81
CRN [39]	8.73	6.57	11.74	74.03	90.96	95.28
Ours (without $L1$)	9.67	7.26	12.94	70.39	89.84	94.73
Ours (with $L1$)	8.59	6.50	11.54	74.61	91.12	95.28

Table 2. Quantitative evaluation of surface normal estimation (on our new high-resolution dataset). The table presents six evaluation statistics, previously suggested by [40,41], over the angular error between the predicted normals and ground-truth normals. For the first three criteria lower is better, while for the latter three higher is better. Our framework leads to the best results on all six measures.

We would like to draw your attention to the inferior scores obtained when removing the $L1$ term from our objective (i.e., setting $\lambda_{L1} = 0$ in Eq. (12)). Interestingly, this contradicts what one sees when examining the results visually. Looking at the reconstructed surfaces reveals that they look more natural and more similar to the ground-truth without $L1$. A few examples illustrating this are provided in Figures 7, 8. This phenomena is actually not surprising at all. In fact, it is aligned with the recent trends in super-resolution, discussed in Section 4.1, where perceptual evaluation measures are becoming a common assessment tool. Unfortunately, such perceptual measures are not the common evaluation criteria for assessing reconstruction of surface normals.

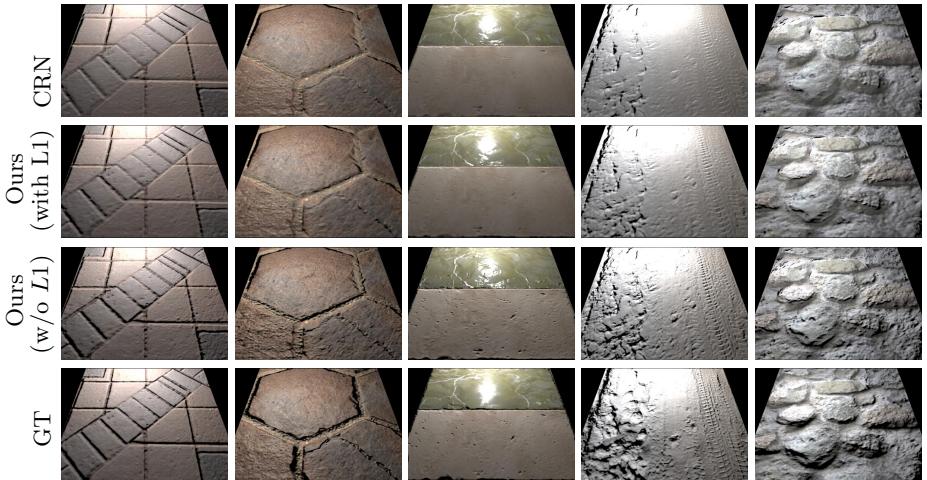


Fig. 8. Rendered images: To illustrate the benefits of our approach we render new view images using the estimated (or ground-truth) normal maps. Using our framework better maintains the statistics of the original surface and thus results with more natural looking images that are more similar to the ground-truth. Again, as was shown in Figure 7, the results without $L1$ look more natural than those with $L1$.

Finally, we would like to emphasize, that our approach generalizes well to other textures outside our dataset. This can be seen from the results in Figure 1 where two texture images, found online, were fed to our trained network. The reconstructed surface normals are highly detailed and look natural.

5 Conclusions

In this paper we proposed using loss functions based on a statistical comparison between the output and target for training generator networks. It was shown via multiple experiments that such an approach can produce high-quality and state-of-the-art results. While we suggest adopting the Contextual loss to measure the difference between distributions, other loss functions of a similar nature could (and should, may we add) be explored. We plan to delve into this in our future work.

Acknowledgements the Israel Science Foundation under Grant 1089/16 and by the Ollendorf foundation.

References

1. Ruderman, D.L.: The statistics of natural images. Network: computation in neural systems (1994) [2](#)

2. Levin, A., Zomet, A., Weiss, Y.: Learning how to inpaint from global image statistics. In: ICCV. (2003) [2](#)
3. Weiss, Y., Freeman, W.T.: What makes a good model of natural images? In: CVPR. (2007) [2](#)
4. Roth, S., Black, M.J.: Fields of experts. IJCV (2009) [2](#)
5. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV, IEEE (2011) [2](#)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014) [2](#)
7. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR. (2015) [2](#)
8. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017) [2](#)
9. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. (2017) [2](#), [7](#), [9](#), [10](#), [11](#)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. (2017) [2](#)
11. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS. (2016) [2](#)
12. Levin, A.: Blind motion deblurring using image statistics. In: Advances in Neural Information Processing Systems. (2007) 841–848 [2](#)
13. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing (2006) [2](#)
14. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research (2010) [2](#)
15. Hassner, T., Basri, R.: Example based 3d reconstruction from single 2d images. In: CVPR Workshop. (2006) [2](#), [3](#)
16. Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: ECCV. (2014) [2](#)
17. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: ICCV. (2009) [2](#)
18. Zontak, M., Irani, M.: Internal statistics of a single natural image. In: CVPR. (2011) [2](#)
19. Mechrez, R., Talmi, I., Zelnik-Manor, L.: The contextual loss for image transformation with non-aligned data. (2018) [2](#), [3](#), [5](#), [6](#), [7](#), [13](#)
20. Gal, R., Shamir, A., Hassner, T., Pauly, M., Cohen-Or, D.: Surface reconstruction using local shape priors. In: Symposium on Geometry Processing. (2007) [3](#)
21. Huang, J., Lee, A.B., Mumford, D.: Statistics of range images. In: CVPR. (2000) [3](#)
22. Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., Wolf, H.C.: Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical report (1977) [5](#), [6](#)
23. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3d: Dataset and methods for single-image 3d shape modeling. In: CVPR. (2018) [6](#)
24. de Villiers, H.A., van Zijl, L., Niesler, T.R.: Vision-based hand pose estimation through similarity search using the earth mover’s distance. IET computer vision (2012) [6](#)

25. Botev, Z.I., Grotowski, J.F., Kroese, D.P., et al.: Kernel density estimation via diffusion. *The annals of Statistics* (2010) 8
26. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. (2018) 8, 9
27. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* (2004) 8, 9, 10
28. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., et al.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: CVPR Workshops. (2017) 9
29. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops. (2017) 9
30. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: CVPR. (2017) 9
31. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: CVPR. (2018) 9, 10
32. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. (2016) 9, 10, 13
33. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR. (2016) 9, 10
34. Sajjadi, M.S., Scholkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: ICCV. (2017) 9, 10, 11
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 9
36. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: CVPR Workshops. (2017) 9
37. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV, IEEE (2001) 10
38. Ma, C., Yang, C.Y., Yang, X., Yang, M.H.: Learning a no-reference quality metric for single-image super-resolution. Computer Vision and Image Understanding (2017) 10
39. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV. (2017) 12, 13
40. Bansal, A., Chen, X., Russell, B., Ramanan, A.G., et al.: Pixelnet: Representation of the pixels, by the pixels, and for the pixels. In: CVPR. (2016) 11, 13
41. Wang, X., Fouhey, D., Gupta, A.: Designing deep networks for surface normal estimation. In: CVPR. (2015) 11, 13
42. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658 11
43. Chen, W., Xiang, D., Deng, J.: Surface normals in the wild. In: ICCV. (2017) 11
44. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV. (2012) 11