Single Image Reflection Removal Exploiting Misaligned Training Data and Network Enhancements

Kaixuan Wei[1] Jiaolong Yang[2] Ying Fu[1*] David Wipf[2] Hua Huang[1]
[1]Beijing Institute of Technology [2]Microsoft Research

Abstract

Removing undesirable reflections from a single image captured through a glass window is of practical importance to visual computing systems. Although state-of-the-art methods can obtain decent results in certain situations, performance declines significantly when tackling more general real-world cases. These failures stem from the intrinsic difficulty of single image reflection removal – the fundamental ill-posedness of the problem, and the insufficiency of densely-labeled training data needed for resolving this ambiguity within learning-based neural network pipelines. In this paper, we address these issues by exploiting targeted network enhancements and the novel use of misaligned data. For the former, we augment a baseline network architecture by embedding context encoding modules that are capable of leveraging high-level contextual clues to reduce indeterminacy within areas containing strong reflections. For the latter, we introduce an alignment-invariant loss function that facilitates exploiting misaligned real-world training data that is much easier to collect. Experimental results collectively show that our method outperforms the state-of-the-art with aligned data, and that significant improvements are possible when using additional misaligned data.

1. Introduction

Reflection is a frequently-encountered source of image corruption that can arise when shooting through a glass surface. Such corruptions can be addressed via the process of single image reflection removal (SIRR), a challenging problem that has attracted considerable attention from the computer vision community [22, 25, 39, 2, 5, 47, 44, 38]. Traditional optimization-based methods often leverage manual intervention or strong prior assumptions to render the problem more tractable [22, 25]. Recently, alternative learning-based approaches rely on deep Convectional Neural Networks (CNNs) in lieu of the costly optimization and hand-crafted priors [5, 47, 44, 38]. But promising results notwithstanding, SIRR remains a largely unsolved problem across disparate imaging conditions and varying scene content.

For CNN-based reflection removal, our focus herein, the challenge originates from at least two sources: **(i)** The extraction of a background image layer devoid of reflection artifacts is fundamentally ill-posed, and **(ii)** Training data from real-world scenes, are exceedingly scarce because of the difficulty in obtaining ground-truth labels.

Mathematically speaking, it is typically assumed that a captured image I is formed as a linear combination of a background or transmitted layer T and a reflection layer R, *i.e.*, $I = T + R$. Obviously, when given access only to I, there exists an infinite number of feasible decompositions. Further compounding the problem is the fact that both T and R involve content from real scenes that may have overlapping appearance distributions. This can make them difficult to distinguish even for human observers in some cases, and simple priors that might mitigate this ambiguity are not available except under specialized conditions.

On the other hand, although CNNs can perform a wide variety visual tasks, at times exceeding human capabilities, they generally require a large volume of labeled training data. Unfortunately, real reflection images accompanied with densely-labeled, ground-truth transmitted layer intensities are scarce. Consequently, previous learning-based approaches have resorted to training with synthesized images [5, 38, 47] and/or small real-world data captured from specialized devices [47]. However, existing image synthesis procedures are heuristic and the domain gap may jeopardize accuracy on real images. On the other hand, collecting sufficient additional real data with precise ground-truth labels is tremendously labor-intensive.

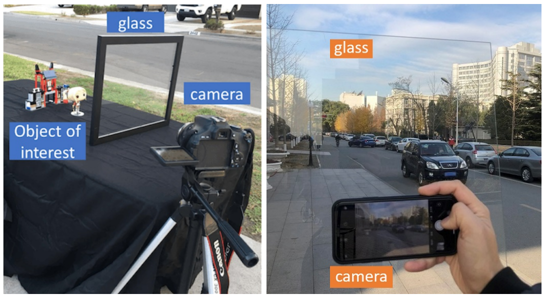
This paper is devoted to addressing both of the aforementioned challenges. First, to better tackle the intrinsic ill-posedness and diminish ambiguity, we propose to leverage a network architecture that is sensitive to contextual information, which has proven useful for other vision tasks such as semantic segmentation [11, 48, 46, 13]. Note that at a high level, our objective is to efficiently convert prior information mined from labeled training data into network structures capable of resolving this ambiguity. Within a tra-

[*] Corresponding author: fuying@bit.edu.cn

ditional CNN model, especially in the early layers where the effective receptive field is small, the extracted features across all channels are inherently local. However, broader non-local context is necessary to differentiate those features that are descriptive of the desired transmitted image, and those that can be discarded as reflection-based. For example, in image neighborhoods containing a particularly strong reflection component, accurate separation by any possible method (even one trained with arbitrarily rich training data) will likely require contextual information from regions without reflection. To address this issue, we utilize two complementary forms of context, namely, channel-wise context and multi-scale spatial context. Regarding the former, we apply a channel attention mechanism to the feature maps from convolutional layers such that different features are weighed differently according to global statistics of the activations. For the latter, we aggregate information across a pyramid of feature map scales within each channel to reach a global contextual consistency in the spatial domain. Our experiments demonstrate that significant improvement can be obtained by these enhancements, leading to state-of-the-art performance on two real-image datasets.

Secondly, orthogonal to architectural considerations, we seek to expand the sources of viable training data by facilitating the use of misaligned training pairs, which are considerably easier to collect. Misalignment between an input image I and a ground-truth reflection-free version T can be caused by camera and/or object movements during the acquisition process. In the previous works [37, 46], data pairs (I, T) were obtained by taking an initial photo through a glass plane, followed by capturing a second one after the glass has been removed. This process requires that the camera, scene, and even lighting conditions remain static. Adhering to these requirements across a broad acquisition campaign can significantly reduce both the quantity and diversity of the collected data. Additionally, post-processing may also be necessary to accurately align I and T to compensate for spatial shifts caused by the refractive effect [37]. In contrast, capturing unaligned data is considerably less burdensome, as shown in Fig. 1. For example, there is no need for a tripod, table, or other special hardware; the camera can be hand-held and the pose can be freely adjusted; dynamic scenes in the presence of vehicles, humans, *etc.* can be incorporated; and finally no post-processing of any type is needed.

To handle such misaligned training data, we require a loss function that is, to the extent possible, invariant to the alignment, *i.e.*, the measured image content discrepancy between the network prediction and its unaligned reference should be similar to what would have been observed if the reference was actually aligned. In the context of image style transfer [17] and others, certain perceptual loss functions have been shown to be relatively invariant to various transformations. Our study shows that the using only the highest-level feature from a deep network (VGG-19 in our case) leads to satisfactory results for our reflection removal task. In both simulation tests and experiments using a newly collected dataset, we demonstrate for the first time that training/fine-tuning a CNN with unaligned data improves the reflection removal results by a large margin.



[46] Ours

Figure 1: Comparison of the reflection image data collection methods in [46] and this paper.

2. Related Work

This paper is concerned with reflection removal from a single image. Previous methods utilizing multiple input images of, *e.g.*, flash/non-flash pairs [1], different polarization [20], multi-view or video sequences [6, 35, 30, 7, 24, 34, 9, 43, 45] will not be considered here.

Traditional methods. Reflection removal from a single image is a massively ill-posed problem. Additional priors are needed to solve the otherwise prohibitively-difficult problem in traditional optimization-based method [22, 25, 39, 2, 36]. In [22], user annotations are used to guide layer separation jointly with a gradient sparsity prior [23]. [25] introduces a relative smoothness prior where the reflections are assumed to be blurry thus their large gradients are penalized. [39] explores a variant of the smoothness prior where a multi-scale Depth-of-Field (DoF) confidence map is utilized to perform edge classification. [31] exploits the ghost cues for layer separation. [2] proposes a simple optimization formulation with an l_0 gradient penalty on the transmitted layer inspired by image smoothing algorithms [42]. Despite decent results can be obtained by these methods where their assumptions hold, the vastly-different imaging conditions and complex scene content in the real world render their generalization problematic.

Deep learning based methods. Recently, there is an emerging interest in applying deep convolutional neural networks for single image reflection removal such that the handcrafted priors can be replaced by data-driven learning [5, 38, 47, 44]. The first CNN-based method is due to [5], where a network structure is proposed to first predict the background layer in the edge domain followed by reconstructing it the color domain. Later, [38] proposes to predict the edge and image intensity concurrently by two cooperative sub-networks. The recent work of [44] presents a cascade network structure which predicts the background layer and reflection layer in an interleaved fashion. The earlier CNN-based methods typical use the raw image intensity discrepancy such as mean squared error (MSE) to train the networks. Several recent works [47, 16, 3] adopt the perceptual loss [17] which uses the multi-stage features of a deep network pre-trained on ImageNet [29]. [47]. Adversarial loss is investigated in [47, 21] to improve the realism of the predicted background layers.

3. Approach

Given an input image I contaminated with reflections, our goal is to estimate a reflection-free trasmitted image $\hat{T}$. To achieve this, we train a feed-forward CNN G_{θ_G} parameterized by θ_G to minimize a reflection removal loss function l. Given training image pairs $\{(I_n, T_n)\}$, $n = 1, \cdots, N$, this involves solving:

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} l(G_{\theta_G}(I_n), T_n). \tag{1}$$

We will first introduce the details of network architecture G_{θ_G} followed by the loss function l applied to both aligned data (the common case) and newly proposed unaligned data extensions. The overall system is illustrated in Fig. 2.

3.1. Basic Image Reconstruction Network

Our starting point can be viewed as the basic image reconstruction neural network component from [5] but modified in three aspects: (1) We simplify the basic residual block [12] by removing the batch normalization (BN) layer [14]; (2) we increase the capacity by widening the network from 64 to 256 feature maps; and (3) for each input image I, we extract hypercolumn features [10] from a pretrained VGG-19 network [32], and concatenate these features with I as an augmented network input. As explained in [47], such an augmentation strategy can help enable the network to learn semantic clues from the input image.

Note that removing the BN layer from our network turns out to be critical for optimizing performance in the present context. As shown in [41], if batch sizes become too small, prediction errors can increase precipitously and stability issues can arise. Moreover, for a dense prediction task such as SIRR, large batch sizes can become prohibitively expensive in terms of memory requirements. In our case, we found that within the tenable batch sizes available for reflection removal, BN led to considerably worse performance, including color attenuation/shifting issues as sometimes observed in image-to-image translation tasks [5, 15, 49]. BN layers have similarly been removed from other dense prediction tasks such as image super-resolution [26] or deblurring [28].

At this point, we have constructed a useful base architecture upon which other more targeted alterations will be applied shortly. This baseline, which we will henceforth refer to as *BaseNet*, performs quite well when trained and tested on synthetic data. However, when deployed on real-world reflection images we found that its performance degraded by an appreciable amount, especially on the 20 real images from [47]. Therefore, to better mitigate the transition from the make-believe world of synthetic images to real-life photographs, we describe two modifications for introducing broader contextual information into otherwise local convolutional filters.

3.2. Context Encoding Modules

As mentioned previously, we consider both context between channels and multi-scale context within channels.

Channel-wise context. The underlying design principle here is to introduce global contextual information across channels, and a richer overall structure within residual blocks, without dramatically increasing the parameter count. One way to accomplish this is by incorporating a channel attention module originally developed in [13] to recalibrate feature maps using global summary statistics.

Let $U = [u_1, \ldots, u_c, \ldots, u_C]$ denote original, uncalibrated activations produced by a network block, with C feature maps of size of $H \times W$. These activations generally only reflect local information residing within the corresponding receptive fields of each filter. We then form scalar, channel-specific descriptors $z_c = f_{gp}(u_c)$ by applying a global average pooling operator f_{gp} to each feature map $u_c \in \mathbb{R}^{H \times W}$. The vector $z = [z_1, \ldots, z_C] \in \mathbb{R}^C$ represents a simple statistical summary of global, per-channel activations and, when passed through a small network structure, can be used to adaptively predict the relative importance of each channel [13].

More specifically, the channel attention module first computes $s = \sigma(W_U \delta(W_D z))$ where W_D is a trainable weight matrix that downsamples z to dimension $R < C$, δ is a ReLU non-linearity, W_U represents a trainable upsampling weight matrix, and σ is a sigmoidal activation. Elements of the resulting output vector $s \in \mathbb{R}^C$ serve as channel-specific gates for calibrating feature maps via $\hat{u}_c = s_c \cdot u_c$.

Consequently, although each individual convolutional filter has a local receptive field, the determination of which channels are actually important in predicting the transmission layer and suppressing reflections is based on the processing of a global statistic (meaning the channel descriptors computed as activations pass through the network during inference). Additionally, the parameter overhead introduced by this process is exceedingly modest given that W_D and W_U are just small additional weight matrices associated with each block.

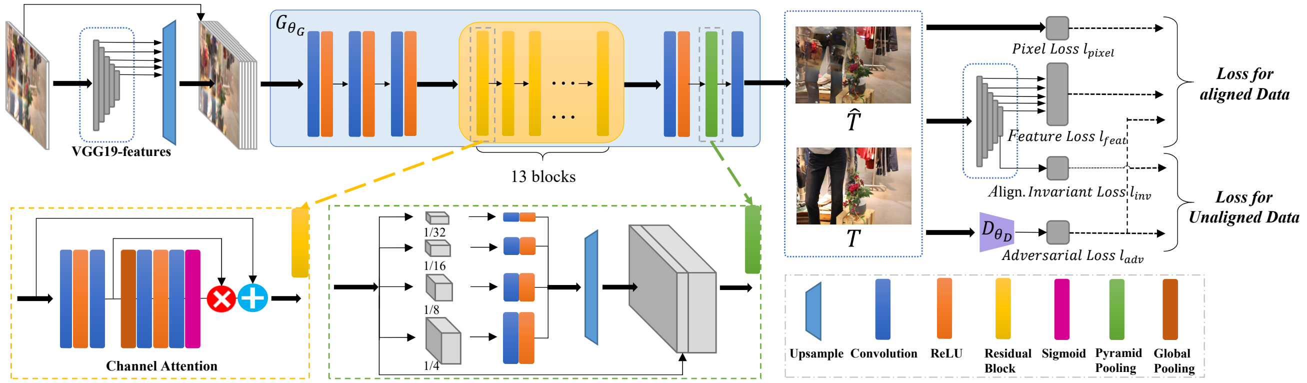


Figure 2: Overview of our approach for single image reflection removal.

Multi-scale spatial context. Although we have found that encoding the contextual information across channels already leads to significant empirical gains on real-world images, utilizing complementary multi-scale spatial information within each channel provides further benefit. To accomplish this, we apply a pyramid pooling module [11], which has proven to be an effective global-scene-level representation in semantic segmentation [48]. As shown in Fig. 2, we construct such a module using pooling operations at sizes 4, 8, 16, and 32 situated in the tail of our network before the final construction of $\hat{T}$. Pooling in this way fuses features under four different pyramid scales. After harvesting the resulting sub-region representations, we perform a non-linear transformation (*i.e.* a Conv-ReLU pair) to reduce the channel dimension. The refined features are then upsampled via bilinear interpolation. Finally, the different levels of features are concatenated together as a final representation reflecting multi-scale spatial context within each channel; the increased parameter overhead is negligible.

3.3. Training Loss for Aligned Data

In this section, we present our loss function for aligned training pairs (I, T), which consists of three terms similar to previous methods [47, 44].

Pixel loss. Following [5], we penalize the pixel-wise intensity difference of T and $\hat{T}$ via $l_{pixel} = \alpha\|\hat{T} - T\|_2^2 + \beta(\|\nabla_x\hat{T} - \nabla_x T\|_1 + \|\nabla_y\hat{T} - \nabla_y T\|_1)$ where ∇_x and ∇_y are the gradient operator along x- and y-direction, respectively. We set $\alpha = 0.2$ and $\beta = 0.4$ in all our experiments.

Feature loss. We define the feature loss based on the activations of the 19-layer VGG network [33] pretrained on ImageNet [29]. Let ϕ_l be the feature from the l-th layer of VGG-19, we define the feature loss as $l_{feat} = \sum_l \lambda_l \|\phi_l(T) - \phi_l(\hat{T})\|_1$ where $\{\lambda_l\}$ are the balancing weights. Similar to [47], we use the layers 'conv2_2', 'conv3_2', 'conv4_2', and 'conv5_2' of VGG-19 net.

Adversarial loss. We further add an adversarial loss to improve the realism of the produced background images. We define an opponent discriminator network D_{θ_D} and minimize the relativistic adversarial loss [18] defined as $l_{adv} = l_{adv}^G = -\log(D_{\theta_D}(T,\hat{T})) - \log(1 - D_{\theta_D}(\hat{T},T))$ for G_{θ_G} and $l_{adv}^D = -\log(1 - D_{\theta_D}(T,\hat{T})) - \log(D_{\theta_D}(\hat{T},T))$ for D_{θ_D} where $D_{\theta_D}(T,\hat{T}) = \sigma(C(T) - C(\hat{T}))$ with $\sigma(\cdot)$ being the sigmoid function and $C(\cdot)$ the non-transformed discriminator function (refer to [18] for details).

To summarize, our loss for aligned data is defined as:

$$l_{aligned} = \omega_1 l_{pixel} + \omega_2 l_{feat} + \omega_3 l_{adv} \tag{2}$$

where we empirically set the weights as $\omega_1 = 1, \omega_2 = 0.1$, and $\omega_3 = 0.01$ respectively throughout our experiments.

3.4. Training Loss for Unaligned Data

To use misaligned data pairs (I, T) for training, we need a loss function that is invariant to the alignment, such that the true similarity between T and the prediction $\hat{T}$ can be reasonably measured. In this regard, we note that human observers can easily assess the similarity of two images even if they are not aligned. Consequently, designing a loss measuring image similarity on the perceptual-level may serve our goal. This motivates us to directly use a deep feature loss for unaligned data.

Intuitively, the deeper the feature, the more likely it is to be insensitive to misalignment. To experimentally verify this and find a suitable feature layer for our purposes, we conducted tests using a pre-trained VGG-19 network as follows. Given an unaligned image pair (I, T), we use gradient descent to finetune the weights of our network G_{θ_G} to minimize the feature difference of T and $\hat{T}$, with features extracted at different layers of VGG-19. Figure 3 shows that using low-level or middle-level features from 'conv2_2' to 'conv4_2' leads to blurry results (similar to directly using a pixel-wise loss), although the reflection is more thoroughly removed. In contrast, using the highest-level feature from 'conv5_2' gives rise to a striking result: the predicted background image is sharp and almost reflection-free.

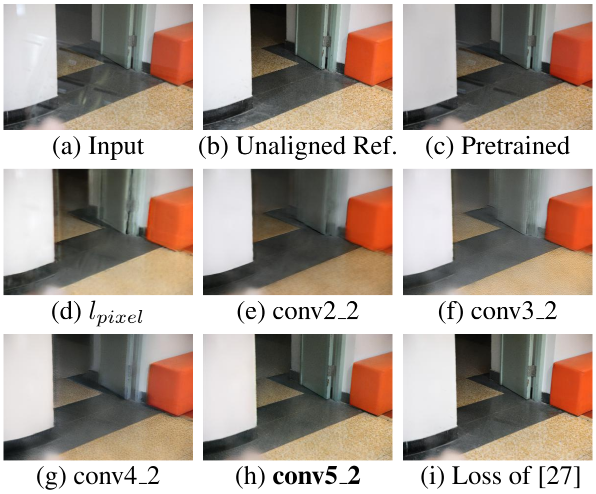


Figure 3: The effect of using different loss to handle misaligned real data. (a) and (b) are the unaligned image pair (I, T). (c) shows the reflection removal result of our network trained on synthetic data and a small number of aligned real data (see Section 4 for details). Reflection can still be observed in the predicted background image. (d) is the result finetuned on (I, T) with pixel-wise intensity loss. (e)-(h) are the results finetuned with features at different layers of VGG-19. Only the highest-level feature from 'conv5_2' yields satisfactory result. (i) shows the results finetuned with the loss of [27]. (**Best viewed on screen with zoom**)

Recently, [27] introduced a "contextual loss" which is also designed for training deep networks with unaligned data for image-to-image translation tasks like image style transfer. In Fig 3, we also present the finetuned result using this loss for our reflection removal task. Upon visual inspection, the results are similar to our highest-level VGG feature loss (quantitative comparison can be found in the experiment section). However, our adopted loss (formally defined below) is much simpler and more computationally efficient than the loss from [27].

Alignment-invariant loss. Based on the above study, we now formally define our invariant loss component designed for unaligned data as $l_{inv} = \|\phi_h(T) - \phi_h(\hat{T})\|_1$, where ϕ_h denotes the 'conv5_2' feature of the pretrained VGG-19 network. For unaligned data, we also apply an adversarial loss which is not affected by misalignment. Therefore, our overall loss for unaligned data can be written as

$$l_{unaligned} = \omega_4 l_{inv} + \omega_5 l_{adv} \tag{3}$$

where we set the weights as $\omega_4 = 0.1$ and $\omega_5 = 0.01$.

4. Experiments

4.1. Implementation Details

Training data. We adopt a fusion of synthetic and real data as our train dataset. The images from [5] are used as sythetic data, *i.e.* 7,643 cropped images with size 224 × 224 from PASCAL VOC dataset [4]. 90 real-world training images from [47] are adopted as real data. For image synthesis, we use the same data generation model as [5] to create our synthetic data. In the following, we always use the same dataset for training, unless specifically stated.

Table 1: Comparison of different settings. Our full model (*i.e.* ERRNet) leads to best performance among all comparisons.

	Synthetic		Real20	
Model	PSNR	SSIM	PSNR	SSIM
CEILNet-F [5]	24.70	0.884	20.32	0.739
BaseNet only	25.71	0.926	21.51	0.780
BaseNet + CSC	27.64	0.940	22.61	0.796
BaseNet + MSC	26.03	0.928	21.75	0.783
ERRNet	**27.88**	**0.941**	**22.89**	**0.803**

Training details. Our implementation[1] is based on PyTorch. We train the model with 60 epoch using the Adam optimizer [19]. The base learning rate is set to 10^{-4} and halved at epoch 30, then reduced to 10^{-5} at epoch 50. The weights are initialized as in [26].

4.2. Ablation Study

In this section, we conduct an ablation study for our method on 100 synthetic testing images from [5] and 20 real testing images from [47] (denoted by 'Real20').

Component analysis. To verify the importance of our network design, we compare four model architectures as described in Section 3, including (1) Our basic image reconstruction network *BaseNet*; (2) *BaseNet* with channel-wise context module (*BaseNet* + CWC); (3) *BaseNet* with multi-scale spatial context module (*BaseNet* + MSC); and (4) Our enhanced reflection removal network, denoted ERRNet, *i.e.*, *BaseNet* + CWC + MSC. The result from the CEILNet [5] fine-tuned on our training data (denoted by CEILNet-F) is also provided as an additional reference.

As shown in Table 1, our *BaseNet* has already achieved a much better result than CEILNet-F. The performance of our *BaseNet* could be obviously boosted by using channel-wise context and multi-scale spatial context modules, especially by using them together, *i.e.* ERRNet. Figure 4 visually shows the results from BaseNet and our ERRNet. It can be observed that *BaseNet* struggles to discriminate the reflection region and yields some obvious residuals, while the ERRNet removes the reflection and produces much cleaner transmitted images. These results suggest the effectiveness of our network design, especially the components tailored to encode the contextual clues.

Efficacy of the training loss for unaligned data. In this

[1] Code is released at https://github.com/Vandermode/ERRNet

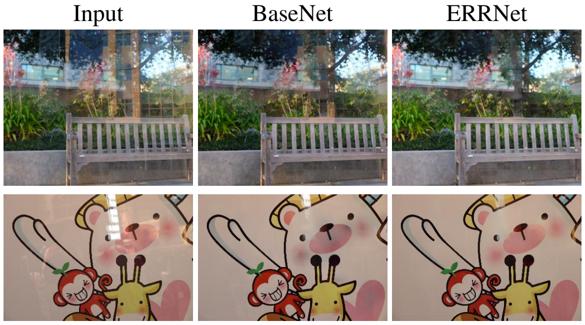


Figure 4: Comparison of the results with (ERRNet) and without (BaseNet) the context encoding modules.

Table 2: Simulation experiment to verify the efficacy our alignment-invariant loss

Training Scheme	PSNR	SSIM
Synthetic only	19.79	0.741
+ 50 aligned	22.00	0.785
+ 90 aligned	22.89	0.803
+ 50 aligned, + 40 unaligned trained with:		
l_{pixel}	21.85	0.766
l_{inv}	22.38	**0.797**
l_{cx}	**22.47**	0.796
$l_{inv} + l_{cx}$	22.43	0.796

experiment, we first train our ERRNet with only 'synthetic data', 'synthetic + 50 aligned real data', and 'synthetic + 90 aligned real data'. The loss function in Eq. (2) is used for aligned data. We can see that the testing results become better with the increasing real data in Table 2.

Then, we synthesize misalignment through performing random translations within $[-10, 10]$ pixels on real data[2], and train ERRNet with 'synthetic + 50 aligned real data + 40 unaligned data'. Pixel-wise loss l_{pixel} and alignment-invariant loss l_{inv} are used for 40 unaligned images. Table 2 shows employing 40 unaligned data with l_{pixel} loss degrades the performance, even worse than that from 50 aligned images without additional unaligned data.

In addition, we also investigate the contextual loss l_{cx} of [27]. Results from both contextual loss l_{cx} and our alignment-invariant loss l_{inv} (or combination of them $l_{inv} + l_{cx}$) surpass analogous results obtained with only aligned images by appreciable margins, indicating that these losses provide useful supervision to networks granted unaligned data. Note although l_{inv} and l_{cx} perform equally well, our l_{inv} is much simpler and computationally efficient than l_{cx}, suggesting l_{inv} is lightweight alternative to l_{cx} in terms of our reflection removal task.

4.3. Method Comparison on Benchmarks

In this section, we compare our ERRNet against state-of-the-art methods including the optimization-based method of [25] (LB14) and the learning-based approaches (CEILNet [5], Zhang *et al*. [47], and BDN [44]). For fair comparison, we finetune these models on our training dataset and report results of both the original pretrained model and finetuned version (denoted with a suffix '-F').

The comparison is conducted on four real-world datasets, *i.e*. 20 testing images in [47] and three sub-datasets from SIR^2 [37]. These three sub-datasets are captured under different conditions: (1) 20 controlled indoor scenes composed by solid objects; (2) 20 different controlled scenes on postcards; and (3) 55 wild scenes[3] with ground truth provided. In the following, we denote these datasets by 'Real20', 'Objects', 'Postcard', and 'Wild', respectively.

Table 3 summarizes the results of all competing methods on four real-world datasets. The quality metrics include PSNR, SSIM [40], NCC [43, 37] and LMSE [8]. Larger values of PSNR, SSIM, and NCC indicate better performance, while a smaller value of LMSE implies a better result. Our ERRNet achieves the state-of-the-art performance in 'Real20' and 'Objects' datasets. Meanwhile, our result is comparable to the best-performing BDN-F on 'Postcard' data. The quantitative results on 'Wild' dataset reveal a frustrating fact, namely, that no method could outperform the naive baseline 'Input', suggesting that there is still large room for improvement.

Figure 5 displays visual results on real-world images. It can be seen that all compared methods fail to handle some strong reflections, but our network more accurately removes many undesirable artifacts, *e.g*. removal of tree branches reflected on the building window in the fourth photo of Fig 5.

4.4. Training with Unaligned Data

To test our alignment-invariant loss on real-world unaligned data, we first collected a dataset of unaligned image pairs with cameras and a portable glass, as shown in Fig. 1 . Both a DSLR camera and a smart phone are used to capture the images. We collected 450 image pairs in total, and some samples are shown in Fig 6. These image pairs are randomly split into a training set of 400 samples and a testing set with 50 samples.

We conduct experiments on the BDN-F and ERRNet models, each of which is first trained on aligned dataset (w/o unaligned) as in Section 4.3, and then finetuned with our alignment-invariant loss and unaligned training data. The resulting pairs before and after finetuning are assembled for human assessment, as no existing numerical metric is available for evaluating unaligned data.

We asked 30 human observers to provide a preference

[2] Our alignment-invariant loss l_{inv} can handle shifts of up to 20 pixels. See *suppl. material* for more details.

[3] Images indexed by 1, 2, 74 are removed due to misalignment.

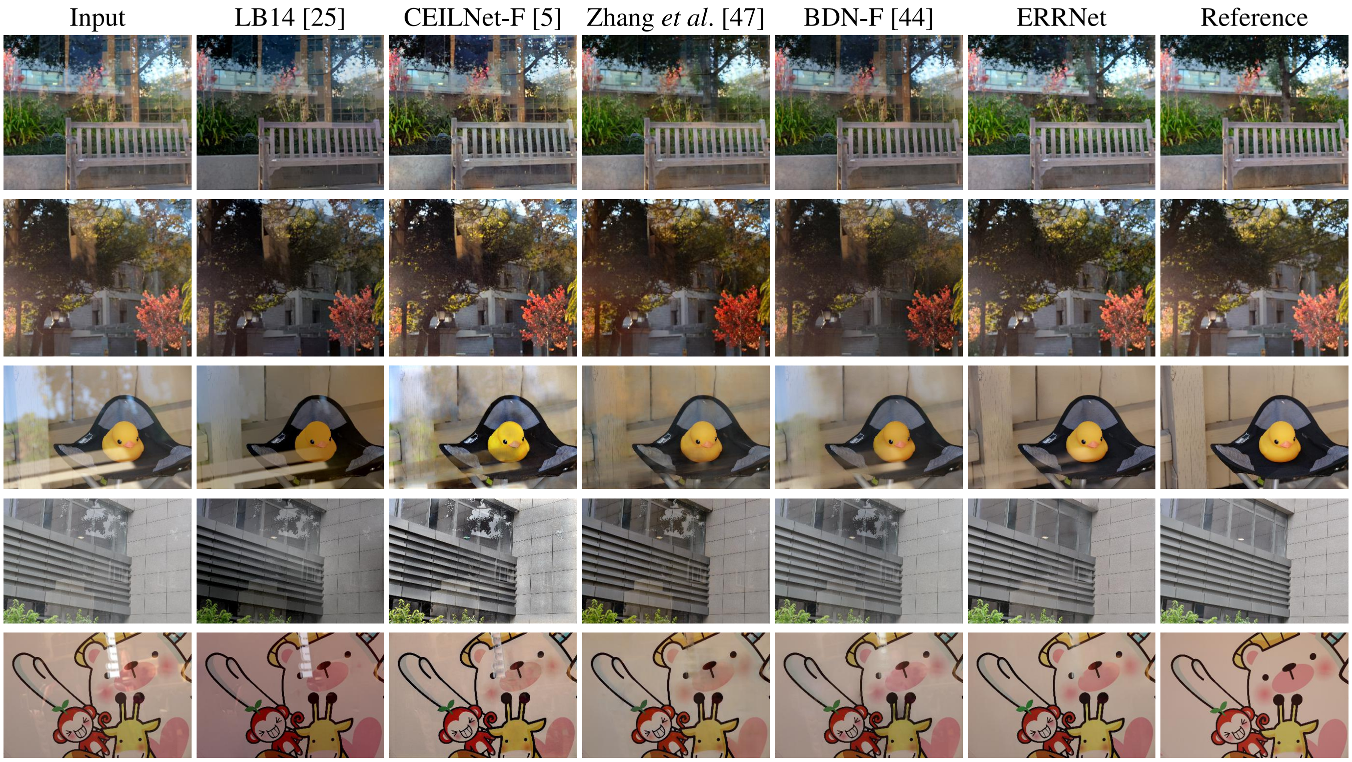


Figure 5: Visual comparison on real-world images. The images are obtained from 'Real20' (Rows 1-3) and our collected unaligned dataset (Rows 4- 5). More results can be found in the *suppl. material*.

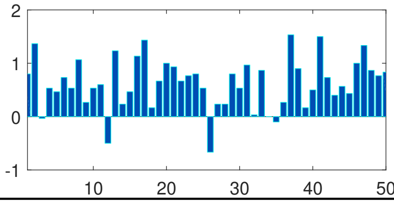
Table 3: Quantitative results of different methods on four real-world benchmark datasets. The best results are indicated by red color and the second best results are denoted by blue color. The results of 'Average' are obtained by averaging the metric scores of all images from these four real-world datasets.

Dataset	Index	Methods							
		Input	LB14 [25]	CEILNet [5]	CEILNet F	Zhang *et al.* [47]	BDN [44]	BDN F	ERRNet
Real20	PSNR	19.05	18.29	18.45	20.32	21.89	18.41	20.06	22.89
	SSIM	0.733	0.683	0.690	0.739	0.787	0.726	0.738	0.803
	NCC	0.812	0.789	0.813	0.834	0.903	0.792	0.825	0.877
	LMSE	0.027	0.033	0.031	0.028	0.022	0.032	0.027	0.022
Objects	PSNR	23.74	19.39	23.62	23.36	22.72	22.73	24.00	24.87
	SSIM	0.878	0.786	0.867	0.873	0.879	0.856	0.893	0.896
	NCC	0.981	0.971	0.972	0.974	0.964	0.978	0.978	0.982
	LMSE	0.004	0.007	0.005	0.005	0.005	0.005	0.004	0.003
Postcard	PSNR	21.30	14.88	21.24	19.17	16.85	20.71	22.19	22.04
	SSIM	0.878	0.795	0.834	0.793	0.799	0.859	0.881	0.876
	NCC	0.947	0.929	0.945	0.926	0.886	0.943	0.941	0.946
	LMSE	0.005	0.008	0.008	0.013	0.007	0.005	0.004	0.004
Wild	PSNR	26.24	19.05	22.36	22.05	21.56	22.36	22.74	24.25
	SSIM	0.897	0.755	0.821	0.844	0.836	0.830	0.872	0.853
	NCC	0.941	0.894	0.918	0.924	0.919	0.932	0.922	0.917
	LMSE	0.005	0.027	0.013	0.009	0.010	0.009	0.008	0.011
Average	PSNR	22.85	17.51	22.30	21.41	20.22	21.70	22.96	23.59
	SSIM	0.874	0.781	0.841	0.832	0.838	0.848	0.879	0.879
	NCC	0.955	0.937	0.948	0.943	0.925	0.951	0.950	0.956
	LMSE	0.006	0.011	0.009	0.010	0.007	0.007	0.006	0.005

score among {-2,-1,0,1,2} with 2 indicating the finetuned result is significantly better while -2 the opposite. To avoid bias, we randomly switch the image positions of each pair. In total, 3000 human judgments are collected (2 methods, 30 users, 50 images pairs). More details regarding this evaluation process can be found in the *suppl. material*.

Figure 6: Image samples in our unaligned image dataset. Our dataset covers a large variety of indoor and outdoor environments including dynamic scenes with vehicles, human, *etc*.

Score Range Ratio	BDN-F	ERRNet
$(0.25, 2]$	78%	54%
$[-0.25, 0.25]$	18%	36%
$[-2, -0.25)$	4%	10%
Average Score	0.62	0.51



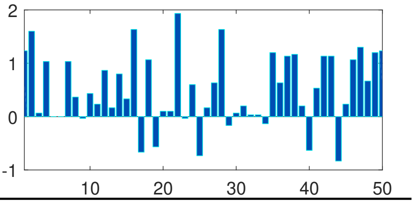


Table 4: Human preference scores of self-comparsion experiments. Left: results of BDN-F; Right: results of ERRNet. X axis of each sub-figure represents the image # of testing images (50 in total).

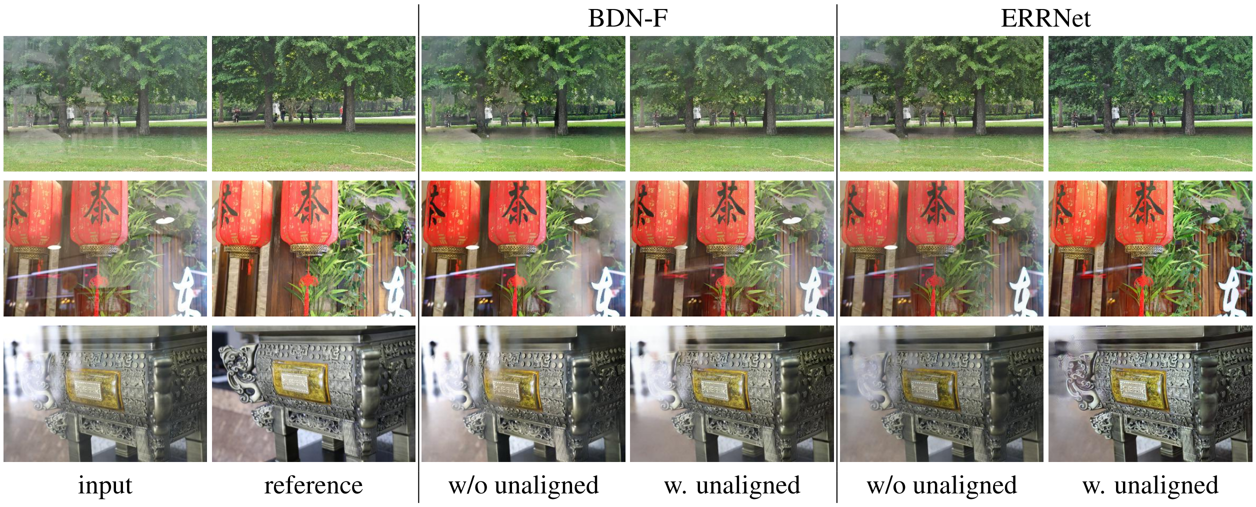


Figure 7: Results of training with and without unaligned data. See *suppl. material* for more examples. (**Best view on screen with zoom**)

Table 4 shows the average of human preference scores for the resulting pairs of each method. As can be seen, human observers clearly tend to prefer the results produced by the finetuned models over the raw ones, which demonstrates the benefit of leveraging unaligned data for training independent of the network architecture. Figure 7 shows some typical results of the two methods; the results are significantly improved by training on unaligned data.

5. Conclusion

We have proposed an enhanced reflection removal network together with an alignment-invariant loss function to help resolve the difficulty of single image reflection removal. We investigated the possibility to directly utilize misaligned training data, which can significantly alleviate the burden of capturing real-world training data. To efficiently extract the underlying knowledge from real training data, we introduce context encoding modules, which can be seamlessly embedded into our network to help discriminate and suppress the reflection component. Extensive experiments demonstrate our approach set a new state-of-the-art on real-world benchmarks of single image reflection removal, both quantitatively and visually.

Acknowledgments

We thank Yunhao Zou for great help collecting the reflection image dataset. This work was supported by the National Natural Science Foundation of China under Grants No. 61425013 and No. 61672096.

References

[1] A. Agrawal, R. Raskar, S. K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics (TOG)*, 24(3):828–835, 2005.

[2] N. Arvanitopoulos, R. Achanta, and S. Susstrunk. Single image reflection suppression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] Z. Chi, X. Wu, X. Shu, and J. Gu. Single image reflection removal using deep encoder-decoder network. *arXiv preprint arXiv:1802.00094*, 2018.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[5] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] H. Farid and E. H. Adelson. Separating reflections and lighting using independent components analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 1999.

[7] K. Gai, Z. Shi, and C. Zhang. Blind separation of superimposed moving images using image statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(1):19–32, 2012.

[8] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct 2009.

[9] X. Guo, X. Cao, and Y. Ma. Robust separation of reflection from multiple images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2014.

[10] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(9):1904–1916, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] M. Jin, S. Ssstrunk, and P. Favaro. Learning to see through reflections. In *IEEE International Conference on Computational Photography (ICCP)*, May 2018.

[17] J. Johnson, A. Alahi, and L. Feifei. Perceptual losses for real-time style transfer and super-resolution. *European Conference on Computer Vision (ECCV)*, pages 694–711, 2016.

[18] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations (ICLR)*, 2019.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] N. Kong, Y.-W. Tai, and J. S. Shin. A physically-based approach to reflection separation: from physical modeling to constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(2):209–221, 2014.

[21] D. Lee, M.-H. Yang, and S. Oh. Generative single image reflection separation. *arXiv preprint arXiv:1801.04102*, 2018.

[22] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(9):1647–1654, 2007.

[23] A. Levin, A. Zomet, and Y. Weiss. Learning to perceive transparency from the statistics of natural scenes. In *Advances in Neural Information Processing Systems (NIPS)*. December 2002.

[24] Y. Li and M. S. Brown. Exploiting reflection change for automatic reflection removal. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[25] Y. Li and M. S. Brown. Single image layer separation using relative smoothness. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2752–2759, 2014.

[26] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[27] R. Mechrez, I. Talmi, and L. Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[28] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[30] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In *European Conference on Computer Vision (ECCV)*, September 2004.

[31] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Machine Learning (ICLR)*, 2015.

[34] S. N. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski. Image-based rendering for scenes with reflections. *ACM Transactions on Graphics (TOG)*, 31(4):100–1, 2012.

[35] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2000.

[36] R. Wan, B. Shi, L. Duan, A. Tan, W. Gao, and A. C. Kot. Region-aware reflection removal with unified content and gradient priors. *IEEE Transactions on Image Processing*, 27(6):2927–2941, 2018.

[37] R. Wan, B. Shi, L.-Y. Duan, A.-H. Tan, and A. C. Kot. Benchmarking single-image reflection removal algorithms. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[38] R. Wan, B. Shi, L.-Y. Duan, A.-H. Tan, and A. C. Kot. Crrn: Multi-scale guided concurrent reflection removal network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[39] R. Wan, B. Shi, T. A. Hwee, and A. C. Kot. Depth of field guided reflection removal. In *IEEE International Conference on Image Processing*, September 2016.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[41] Y. Wu and K. He. Group normalization. In *European Conference on Computer Vision (ECCV)*, September 2018.

[42] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via L_0 gradient minimization. In *ACM Transactions on Graphics (TOG)*, volume 30, page 174, 2011.

[43] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34(4):79, 2015.

[44] J. Yang, D. Gong, L. Liu, and Q. Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[45] J. Yang, H. Li, Y. Dai, and R. T. Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[46] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[47] X. Zhang, R. Ng, and Q. Chen. Single image reflection separation with perceptual losses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[48] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[49] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.