Blur2Blur: Blur Conversion for Unsupervised Image Deblurring on Unknown Domains

Bang-Dang Pham[1] Phong Tran[1,2] Anh Tran[1] Cuong Pham[1,3] Rang Nguyen[1] Minh Hoai[1,4*]

[1]VinAI Research, Vietnam [2]MBZUAI, UAE [3]Posts & Telecommunications Inst. of Tech., Vietnam [4]The University of Adelaide, Australia

{v.dangpb1, v.anhtt152, v.hoainm}@vinai.io cuongpv@ptit.edu.vn the.tran@mbzuai.ac.ae

[*]Work done when being at Stony Brook University, USA

Figure 1. We address the unsupervised image deblurring problem by training a blur translator that converts an input image with unknown blur to an image with a predefined known blur. The figure shows the effectiveness of our approach. The blurry images before and after translation (left image in each box) exhibit similar visual content but have different blur patterns (zoomed-in patches). While a standard image deblurring technique fails to restore the unknown-blur image, it successfully recovers the known-blur version, yielding an approximate 2.2 dB increase in PSNR score (noted below each deblurred image on the right side of each box).

Abstract

This paper presents an innovative framework designed to train an image deblurring algorithm tailored to a specific camera device. This algorithm works by transforming a blurry input image, which is challenging to deblur, into another blurry image that is more amenable to deblurring. The transformation process, from one blurry state to another, leverages unpaired data consisting of sharp and blurry images captured by the target camera device. Learning this blur-to-blur transformation is inherently simpler than direct blur-to-sharp conversion, as it primarily involves modifying blur patterns rather than the intricate task of reconstructing fine image details. The efficacy of the proposed approach has been demonstrated through comprehensive experiments on various benchmarks, where it significantly outperforms state-of-the-art methods both quantitatively and qualitatively.

1. Introduction

Motion blur in images and videos is a common issue, often resulting from camera shake or rapid movement within the scene. Such blur can detract from the aesthetic quality of the content and may undermine the performance of downstream computer vision applications. Consequently, an effective image deblurring method is essential in various contexts.

While the idea of deblurring images from arbitrary, diverse sources sounds impressive and broadly useful, the practical necessity, commercial value, and societal impact of image deblurring are frequently connected to specific application scenarios and particular cameras. For example, a mobile phone manufacturer might focus on integrating the most effective deblurring algorithm for the camera types used in their latest phone models. Similarly, a factory manager might consider installing ceiling-mounted cameras to identify errors on the assembly line, enhancing workforce efficiency. However, motion blur could significantly degrade the performance of computer vision algorithms meant to detect and track workers' hands and tools. In law enforcement, a police officer using a body-worn camera coupled with face recognition technology might find that motion blur hampers the accuracy of detecting faces and identifying fugitives. Therefore, in these scenarios, the development of a framework to customize a deblurring algorithm for specific cameras or camera types becomes crucial and represents a significant and growing need.

In this paper, we explore the question: How can we deblur images captured by specific cameras? Classical deblurring

algorithms, which use signal processing or theoretical models of motion blur, are one option. Yet, their reliance on oversimplified blur models limits their effectiveness in addressing the complex motion blur encountered in real-world scenarios. An alternative is a data-driven approach that leverages advancements in machine learning. This approach involves using pre-trained deblurring networks developed through supervised learning, as illustrated by works such as [2, 3, 14, 30, 32, 37]. These networks, trained on extensive datasets of paired images, aim to transform blurry images into sharp ones. However, they often suffer from overfitting and tend to underperform on novel blurred images that were not captured by the cameras used to create their training datasets. Our empirical findings indicate that the performance of these models is still unsatisfactory when confronting unseen blurs produced by real-world cameras.

When pre-trained networks are unsuitable, the alternative is to develop a deblurring network specifically for our camera. However, this approach faces the challenge of not having access to paired training data, consisting of corresponding blurry and sharp images. Generating such data typically involves a sophisticated setup with a beam splitter, identical cameras operating at varying speeds, and capabilities for time synchronization, geometrical alignment, and color calibration [15, 16, 26]. Often, the camera targeted for deblurring may not meet these stringent requirements, and arranging this setup is not feasible for many. Therefore, we are left with the option of utilizing unpaired data. Yet, training on unpaired data presents its own set of challenges due to the lack of supervision for restoring fine details that are missing or distorted in the blurry input images. Existing methods [18, 35, 39, 41], which attempt to recreate these absent details, frequently fall short, particularly when dealing with blur typical of real-world images.

In this paper, we introduce **Blur2Blur**, a novel framework designed to train an image deblurring algorithm specifically for a chosen camera device. Similar to other unsupervised deblurring methods, we utilize unpaired data. More precisely, we use the target camera to capture a set of blurry images and sharp images, without requiring a one-to-one correspondence between the images in these two sets. This approach makes data collection relatively simple and straightforward. Our method diverges from existing unsupervised methods by not attempting to directly learn a function from the domain of blurry images captured by our camera (the unknown blur domain, denoted as C), to the domain of sharp images. Instead, our strategy involves first learning a mapping G from the domain C to another domain C' of blurry images, where deblurring techniques are already well-established. To deblur an image taken by our camera, we first convert it into an image in C' using the learned mapping G, then apply a pre-trained network to deblur this transformed image. Consequently, our primary goal is to learn the blur-to-blur mapping from C to C', which is inherently less challenging than the direct blur-to-sharp mapping, because the former primarily involves altering blur patterns rather than the more complex task of reconstructing detailed image features.

To learn the blur-to-blur mapping, we propose a novel learning framework to leverage the collected set of blurry and sharp images as well as the blurry images from the known blur domain C'. To train the blur-to-blur mapping network, we carefully define various loss terms, including perceptual, adversarial, and gradient penalty terms. The details of our approach are illustrated in Fig. 1.

We conducted extensive experiments to compare the effectiveness of our model with other state-of-the-art image deblurring approaches on both real-world and synthetic blur datasets. The results demonstrate that Blur2Blur outperforms other methods by a significant margin, highlighting its superior performance in addressing the challenges of image deblurring in real-world settings. Notably, when combined with our blur translation method, supervised methods achieve an impressive boost up to **2.91 dB** in PSNR.

2. Related Work

Many methods have been proposed for image deblurring. Beyond classical methods that do not necessitate training data, many contemporary approaches are grounded in machine learning. Learning-based methods can be broadly categorized based on their data requirements, whether it be paired, synthetic, or unpaired data. This section reviews representative works from these categories.

Classical Image Deblurring. Early deblurring methods assume that the blur operator is linear and uniform. In other words, the blur can be approximated by a single convolution operator: $y = x * k + \eta$, where y, x, k, and η represent the blurry image, sharp image, blur kernel, and noise, respectively. Based on this assumption, given a blurry image y, the sharp image x and the blur kernel k can be obtained by maximizing the posterior distribution: $x^*, k^* = \text{argmax}_{x,k}\, P(x, k|y)P(x)P(k)$. Traditional methods primarily focus on finding prior distributions for either x [1, 7, 12, 13] or k [15, 17, 23]. However, these methods generalize poorly to real-world blurry images because blur kernels are often non-uniform and non-linear.

Supervised learning with paired data. Going beyond the assumptions of uniformity and linearity, several deep deblurring neural networks have been proposed [3, 14, 31, 36], demonstrating promising results. These networks are typically trained on large-scale datasets containing pairs of blurry and sharp images. The distinguishing factors among these works primarily lie on their architectural designs. For instance, Tao et al. [31] introduced a multi-scale recurrent network architecture specifically tailored for image deblurring. Other methods [4, 20] leveraged a coarse-to-fine strategy,

utilizing multi-scale inputs to incrementally refine the deblurring process. Kupyn et al. [14] was the first to incorporate GAN-based loss into the image deblurring framework, aiming to enhance the realism of deblurred images. Meanwhile, Zamir et al. [36] proposed a multi-stage framework that breaks down the image restoration task into smaller, more manageable stages. Lastly, Chen et al. [3] presented a simple yet efficient architecture by reducing the complexity both between and within blocks based on UNet [27] architecture.

In supervised learning, training convolutional networks effectively requires extensive datasets comprising both sharp and blurry image pairs. Acquiring these datasets can be a complex and lengthy process, often necessitating advanced hardware and careful setup. Recent studies [22, 24, 40] have introduced real-world deblurring datasets created using a dual-camera system, consisting of a high-speed and a low-speed camera, synchronized and aligned precisely with a time trigger and a beam splitter. This method ensures the collection of perfectly matched pairs of blurry and sharp images. Nonetheless, a limitation arises as deblurring networks trained on these specific datasets may become too tailored to the characteristics of the cameras used, resulting in reduced performance when applied to images from different cameras. Moreover, the dual-camera system is an advanced setup, requiring specific camera types that meet certain criteria, which means not all cameras are suitable for this purpose.

Supervised learning with synthesized data. One common approach for synthesizing blurry images is to average multiple consecutive sharp frames from a video sequence [20, 21]. Although this synthesis method mimics the way blurry images are captured, it has been demonstrated that models trained on these datasets often underperform when tested on real-world blurry images [25, 34]. Recent studies have proposed more advanced techniques to synthesize deblurring datasets, aiming to improve the generalization of models trained on these datasets to unseen blur. For instance, Zhang et al. [38] created a synthesized dataset by combining multiple types of degradation operators initially developed for the super-resolution task. Rim et al. [25] compared real and synthetic blurry images to design a more realistic blur synthesis pipeline. However, as demonstrated in Sec. 4.2, the degradation augmentation employed in [38] significantly impairs the quality of input images, leading to distorted outputs. On the other hand, models trained on the realistically synthesized deblurring dataset in [25] exhibit signs of overfitting to the training data.

One promising direction was to leverage the known relationship between blurry and sharp image pairs from existing datasets [34]. This method involves capturing the blur distribution characteristic of each pair, which can then be applied to construct a synthesized blurred dataset. Inspired by the effectiveness of this strategy in capturing blur attributes from the known dataset, Blur2Blur adopts this approach. It is designed to discern and retain the blur kernel while selectively ignoring the camera-specific attributes of the target dataset.

Unupservised learning with unpaired data. Another approach to address the overfitting problem is through unpaired deblurring [18, 35, 39, 41]. Unlike supervised methods, these techniques do not require paired sharp and blurry images for training. However, they often face limitations, such as being domain-specific [18], or making low-level statistical assumptions about blur operators [39], which may not be valid for real-world blurry images. To facilitate domain adaptation between blurred and sharp images, other methods [35, 41] have been explored. However, these approaches struggle to bridge the gap between these domains effectively due to (1) the significant variation in the degree of blur across different images, which affects the perceived semantics of the objects within, and (2) the complex and unpredictable nature of real-world blur patterns, often contradicting the simplistic assumptions used in these models. Consequently, the challenge of achieving truly blind image deblurring remains unsolved.

Considering these limitations, our Blur2Blur approach is centered around the innovative idea of blur kernel transfer. This involves transforming the blur kernel from any particular camera into a familiar blur kernel from a dataset or camera that has a strong, pre-trained deblurring model. This method enables us to utilize the benefits of supervised techniques within an unsupervised framework, effectively tackling the challenge of deblurring images with a wide range of unknown blur distributions.

3. Methodology

3.1. Approach Overview

We formulate a blurry image y as a function of the corresponding sharp image x through a blur operator $\mathcal{F}_C(\cdot, k)$, which is associated with a device-dependent blur domain C and a blur kernel k:

$$y = \mathcal{F}_C(x, k) + \eta, \tag{1}$$

where η is a noise term. Our task is to find a deblurring function $\mathcal{G}_C^*$ that can recover the sharp image from the blurry input, i.e., $\mathcal{G}_C^*(y) = x$.

One strategy is to utilize an existing, pre-trained deblurring network to approximate the desired function $\mathcal{G}_C^*$, and then use it for deblurring. However, this approach often leads to unsatisfactory results. The pre-trained network is generally trained on a dataset from a camera with a unique blur space C', which is likely to be different from the blur space C of our camera. In essence, this would mean approximating $\mathcal{G}_C^*$ with $\mathcal{G}_{C'}^*$, an approach that is not ideal due to the differences between C and C', resulting in suboptimal deblurring performance.

When a pre-trained network is not a good choice, our remaining option is to train a new deblurring network tailored to our camera. The obstacle here is that the specific blur space C of our device is unknown, and we cannot rely on having paired training data of corresponding blurry and sharp images. Paired training data requires a complex hardware setup, involving a beam splitter, along with identical devices capturing at different speeds, and the capability for time synchronization, geometrical alignment, and color calibration. Not all camera devices meet these requirements, and setting up such a system is beyond the expertise of many. Consequently, our only feasible option is to use unpaired data. Fortunately, we can access the camera device to capture sets of blurry images $\mathcal{B}$ and sharp images $\mathcal{S}$, which are unpaired and do not necessitate correspondence between images in $\mathcal{B}$ and images in $\mathcal{S}$. Thus, gathering these datasets is relatively easy and straightforward. The downside, however, is that learning from unpaired data is challenging. The deblurring process, which transforms a blurred image y into a sharp image x, typically requires an understanding of the blurring domain C. For unpaired data, this necessity poses a significant hurdle, especially in reconstructing fine details absent or distorted in the blurred input. Traditional deblurring networks [18, 35, 39, 41], attempting to 'hallucinate' these missing details, often produce unsatisfactory results, particularly with images affected by real-world blurring.

In this section, we introduce an innovative method to learn $\mathcal{G}_C^*$. Rather than directly learning this function, which is extremely challenging, or resorting to a rough approximation using a function learned for another blur domain $\mathcal{G}_{C'}^*$, we propose to treat $\mathcal{G}_C^*$ as a composition of $\mathcal{G}_{C'}^*$ and a translation function G, i.e., $\mathcal{G}_C^* = \mathcal{G}_{C'}^* \circ G$. Our goal then shifts to learning this translation function G to bridge the gap between domains C and C'.

More specifically, our task is to learn a mapping function G that maps each blurry input image y defined in Eq. (1) to an image y' with the same sharp visual representation x but belongs to a known blur distribution C':

$$G : y \rightarrow y', \text{where } y' = \mathcal{F}_{C'}(x, k') + \eta'. \tag{2}$$

Our approach breaks a complex task into two manageable ones. One task requires deblurring from C', which, while challenging, benefits from existing research. We can select a well-performing pre-trained network $\mathcal{G}_{C'}^*$, which has been trained with supervised learning using paired data in its domain. The other task is to learn a translation from an unknown blur domain C to a known domain C'. The difficult of this task depends on the differences between C and C', yet it is surely easier than directly learning a mapping from C to a sharp domain. This is because a blur-to-blur transformation primarily modifies the blur patterns, avoiding the need to reconstruct intricate image details. Moreover, we have the flexibility to choose the most appropriate C' and $\mathcal{G}_{C'}^*$ for our specific blur domain. This flexibility extends to the possibility of utilizing synthetic data, which allows for the generation of extensive datasets, ensuring that the deblurring network is thoroughly trained.

In the remaining of this section, we will discuss two main components of our method, including the blur-to-blur translation network G and the target blur space C'.

3.2. Blur-to-blur translation

Our objective here is to train a blur-to-blur translation network G, capable of converting any blurry image from the unknown blur domain C to a known blur domain C' while preserving the image content. To train G, we require two datasets: $\mathcal{B}$, which consists of blurry images from the unknown blur domain, and $\mathcal{K}$ contains images with known blur, for which a deblurring model has already been trained. We design the translation network G to work at multiple scales and carefully design the training losses to achieve the desired outcome.

Adversarial Loss. We employ an adversarial loss [5] to enforce the translation network G to produce images with the desired target blur. To achieve this, we introduce a discriminator network D, which is responsible for distinguishing between real images from the known blur domain and generated images. Two networks G and D are trained alternately in a minimax game. The adversarial loss is defined as:

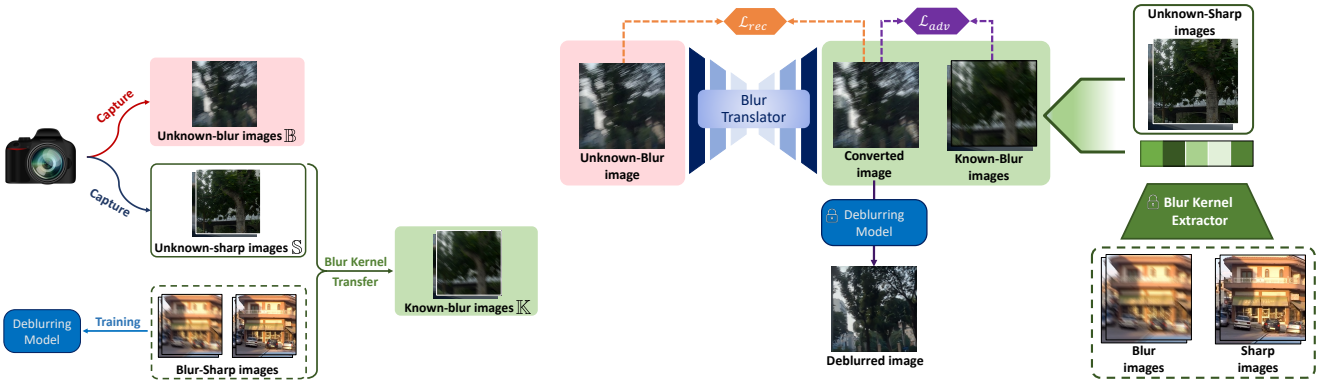
$$\begin{aligned}\mathcal{L}_{adv}(G, D) = \mathbb{E}_{y\sim\mathcal{K}}[\log D(y)] \\ + \mathbb{E}_{y\sim\mathcal{B}}[\log(1 - D(G(y)))].\end{aligned} \tag{3}$$

The blur translation network is trained to minimize the above loss term, while the discriminator D is trained to maximize it. We also force the Lipschitz continuity constraint on the discriminator using the gradient penalty regularization [6]:

$$\mathcal{L}_{grad}^D(D) = \mathbb{E}_{\hat{y}\sim\hat{\mathcal{B}}}[(\|\nabla_{\hat{y}} D(\hat{y})\|_2 - 1)^2], \tag{4}$$

where $\hat{\mathcal{B}}$ is the set of samples $\hat{y}$ randomly interpolated between a real image $y \in \mathcal{B}$ and the generated image $G(y)$ using a random mixing ratio $\epsilon \in [0, 1]$, i.e., $\hat{y} = \epsilon y + (1 - \epsilon)G(y)$.

Reconstruction Loss. Given a blurry image y, the desired function G should translate the blur characteristics from C to C' while maintaining the elements belonging to the sharp image x. Using the adversarial loss helps translate the image to the target blur domain but does not guarantee sharp content preservation. Hence, we integrate a reconstruction loss to enforce the visual consistency between the generated blurry image $G(y)$ and the original image y. This loss term has two benefits: (1) it prevents G from modifying the image content and only focuses on the blur kernel translation, and (2) it provides additional supervision to our network, enhancing the training stability. Moreover, to make G focus



(a) Unknown- and known-blur datasets

(b) Blur translation

Figure 2. **Overview of our problem and proposed method.** a) Given a camera, we aim to develop an algorithm to deblur its captured blurry images. We assume access to the camera to collect *unpaired* sets of blurry images ($\mathcal{B}$) and sharp image sequences ($\mathcal{S}$). b) The key component in our proposed system is a blur translator that converts unknown-blur images captured by the camera to have the target known-blur presented in $\mathcal{K}$. This translator is trained using reconstruction and adversarial losses. The converted images have known blur and can be successfully deblurred using the previously trained deblurring model (Zoom for best view).

on preserving the input semantic content rather than being overly constrained by pixel-wise accuracy, we (1) employ perceptual loss [9] instead of the common L_1 or L_2 loss function and (2) adopt a multi-scale deblurring architecture [4] to reconstruct the image content from coarse to fine:

$$\mathcal{L}^G_{rec}(G) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{t_i}\mathbb{E}_{y_i\sim\mathcal{B}}[||\phi(y_i) - \phi(G(y_i))||_1], \quad (5)$$

where M is the number of levels, y_i is the input image at scale level i, $\phi(.)$ is a pre-trained feature extractor with the VGG19 backbone [29]. We divide the loss by the number of total elements t_i for normalization.

Total Loss. Our final objective function for G combines the adversarial and reconstruction loss terms:

$$\mathcal{L}^G_{total}(G, D) = \mathcal{L}_{adv}(G, D) + \lambda_{rec}\mathcal{L}_{rec}(G), \quad (6)$$

where λ_{rec} is the weight factor for the reconstruction loss, ensuring the image's input content is maintained. Concurrently, the definitive objective function for D is established as follows:

$$\mathcal{L}^D_{total}(G, D) = -\mathcal{L}_{adv}(G, D) + \lambda_{grad}\mathcal{L}_{grad}(D). \quad (7)$$

Here λ_{grad} is a hyperparameter that controls the importance of the gradient penalty loss component.

3.3. Known Blur Selection

The choice of C' and its representative dataset $\mathcal{K}$ is important because the difficulty of learning the blur translation network depends on the discrepancy between the two blur domains. As described in Sec. 3.2, the representative dataset $\mathcal{K}$ only affects the adversarial training losses. The translation network G aims to convert images in $\mathcal{B}$ to have similar blur characteristics as images in $\mathcal{K}$ so that the discriminator D cannot differentiate between the generated images and the real images in $\mathcal{K}$. However, if $\mathcal{K}$ and $\mathcal{B}$ have different characteristics besides the blur kernel distribution, such as color tone, image resolution, or device-dependent noise pattern, D may rely on them to differentiate real and generated images. It can cause G to either fail to converge or introduce undesired characteristics from the representative dataset $\mathcal{K}$ into the transferred outcomes.

To avoid this issue, we propose generating images in $\mathcal{K}$ from a set of sharp images $\mathcal{S}$ captured with the same camera as $\mathcal{B}$, thus sharing identical characteristics. These images are then augmented by blur kernels from a known domain, characterized by a dataset of blurry-sharp image pairs using the blur transfer technique [34]. The blurry-sharp image pair dataset can be selected from commonly used image deblurring datasets like REDS [21], GOPRO [20], RSBlur [26], and RB2V [22], and we can utilize any deblurring network pre-trained on that dataset. A key component in [34] is a Blur Kernel Extractor F that can isolate and transfer blur kernels from random blurry-sharp image pairs to the target sharp inputs. After applying this blur synthesis procedure, we obtain a known-blur image set $\mathcal{K}$ that carries blur kernels from the known-blur domain while maintaining other camera-based characteristics similar to the unknown-blur images in $\mathcal{B}$. Consequently, the discriminator can focus on distinguishing based on blur kernels, facilitating effective blur-to-blur translation training. The overview problem and pipeline of our method is illustrated in Fig. 2.

4. Experiments

We first evaluate our proposed Blur2Blur method on challenging unsupervised image deblurring benchmarks in comparison to state-of-the-art supervised and unsupervised tech-

niques. We then verify the benefit of our known blur selection strategy and examine different aspects in our proposal.

4.1. Experimental Setups

4.1.1 Datasets and implementation details

We evaluate our proposed method on four datasets. **REDS dataset [21]** consists of 300 high-speed videos used to create synthetic blur. By ramping up the frame rate from 120 to 1920 fps and averaging frames with an inverse Camera Response Function (CRF), it simulates more realistic motion blur, differentiating it from other synthetic datasets [19, 28]. **GoPro dataset [20]** comprises 3,142 paired frames of sharp and blurred images, recorded at 240 frames per second. The synthesis method employed for these frames is akin to that of the REDS dataset but with a different camera response function selected. We utilize this dataset as the main target data for evaluating deblurring methods in combination with Blur2Blur. **RSBlur dataset [26]** contains 13,358 real blurred images. It provides sequences of sharp images alongside blurred ones for in-depth blur analysis and offers the higher resolution than similar datasets. Noise levels are also estimated to assess and compare to the noise present in real-world blur scenarios. **RB2V dataset [22]** comprises about 11,000 real-world pairs of a blurry image and a sharp image sequence for street categories, denoted as *RB2V_street*. This dataset was collected using a beam splitter camera system. Experiments on this dataset are crucial for confirming the effectiveness of our algorithm in handling real-world, camera-specific data.

Train and test data. To address practical deblurring problems, our method assumes access to unpaired sets containing blurry images $\mathcal{B}$ and sharp images $\mathcal{S}$. When selecting a dataset as the source for our deblurring evaluation, we divide its training data into two disjoint subsets that capture different scenes with a specific ratio of 0.6:0.4. In the first subset, we select blurry images to form the unknown-blur image set $\mathcal{B}$, while in the second subset, we choose sharp images to construct the sharp set $\mathcal{S}$. For the chosen target dataset, representing the domain for blur kernel translation via the Blur2Blur mechanism, we employ the entire training dataset to train our Blur Kernel Extractor [34] and subsequently apply this extractor to map captured blur embeddings onto the sharp image set $\mathcal{S}$, creating the known-blur image set $\mathcal{K}$. The blurry images in the test data of the source dataset are used to evaluate image deblurring algorithms. The statistics of source image sets are reported in Tab. 1.

Implementation Details. We implemented the blur-to-blur translation network G using MIMO-UNet [4] with the default configuration in Pix2Pix [8] implementation. For all experiments, we set the hyper-parameters $\lambda_{rec} = 0.8, \lambda_{grad} = 0.005$ and batch size of 16. To enhance our understanding of the network G during its initial iterations, we sorted images based on their blur degree. Initially, we optimized approximately 50% of the data within a single batch. Subsequently, after 200K iterations, we incrementally scaled this proportion to encompass the full batch. We evaluated Blur2Blur in combination with different state-of-the-arts deblurring network backbones, including NAFNet [3] and Restormer [37]. During training, we randomly cropped these images to obtain a square shape of 256×256 pixels and augmented with rotation, flip and colorjitter. This served as the standard input for our Blur2Blur model and other evaluations.

Dataset	Number of data samples		
	U. blur ($\mathcal{B}$)	U. sharp ($\mathcal{S}$)	Test
RB2V_Street	5400	3600	2053
REDS	14400	9600	3000
RSBlur	8115	5410	8301
GoPro	1261	842	1111

Table 1. Statistics of datasets used as unknown domains.

All experiments were performed using the Adam optimizer [11]. Training our model required roughly three days for one million iterations on two Nvidia A100 GPU. We conducted experiments using a constant learning rate and applying linear decay scheduler. The learning rate is maintained constant for the first 500K iterations and then linearly reduced during the remaining iterations as in [33].

4.1.2 Baselines

We compared Blur2Blur with a comprehensive list of baseline methods from three categories: supervised methods (NAFNet [3], Restormer [37]), unpaired training (CycleGAN [41], DualGAN [35]), and generalized image deblurring (BSRGAN+NAFNet [38], RSBlur+NAFNet [25]).

For fair comparisons, we retrained the supervised models using the blur-sharp pairs from the source dataset. Furthermore, to replicate real-world scenarios with the absence of paired data for deblurring network training, we generated synthetic motion-blur data derived from the unknown-sharp image set $\mathcal{S}$ by adding motion blur synthesis techniques, such as the one provided by the *imgaug* library [10]. This approach synthesized motion blur independently on each image. For the unpaired training and generalized image deblurring approaches, we used the blurry images in $\mathcal{B}$ and the sharp images from $\mathcal{S}$ for training the deblurring network. BSRGAN was originally designed for blind image super-resolution, and we adapted it to work on blind image deblurring by adding motion blur augmentation (via averaging with neighboring frames) into its augmentation pipeline.

4.2. Image Deblurring Results

To evaluate the performance of the Blur2Blur mechanism, we defined three data configurations. Each configuration

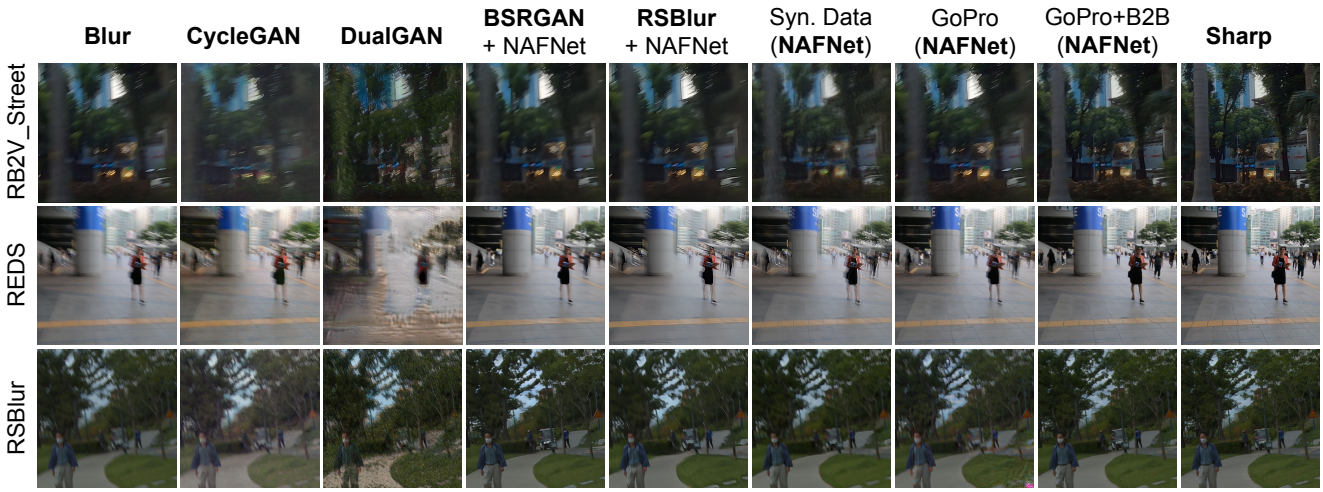


Figure 3. Comparing image deblurring results on three benchmark datasets with NAFNet. Due to space limit, we skip the results with Restormer backbone, which is similar but slightly worse than those with NAFNet. Best viewed when magnified on a digital display.

consists of the Known-Blur dataset $\mathcal{K}$, sourced from the GoPro dataset, and two unpaired datasets, $\mathcal{B}$ and $\mathcal{S}$, derived from the training partitions of the deblurring dataset REDS, RSBlur, or RB2V_Street. For a comprehensive evaluation of the Blur2Blur model, we integrated it with two supervised image deblurring backbones, Restormer and NAFNet. Additionally, we compared the results with state-of-the-art baselines. The quantitative results are summarized in Tab. 2.

As observed, both unsupervised image deblurring and generalized deblurring approaches, despite expecting generalization power, exhibit poor performance on these challenging real-world datasets. Their scores are similar to, and sometimes significantly lower than, state-of-the-art supervised methods such as Restormer and NAFNet. In contrast, Blur2Blur demonstrates remarkable deblurring results. When combined with Restormer, Blur2Blur helps to increase the PSNR score by 2.63 dB on RB2V_Street, 2.12 dB on REDS, and 2.91 dB on RSBlur. When combined with NAFNet, it provides consistent score increases, with 2.20 dB on RB2V_Street, 2.31 dB on REDS, and 2.67 dB on RSBlur. NAFNet outperforms Restormer overall, making the combination of NAFNet and Blur2Blur the most effective deblurring approach. Moreover, our method comes close to matching the best results of supervised models trained on source datasets.

We provide a qualitative comparison between image deblurring results in Fig. 3. The qualitative comparison highlights a significant performance disparity between supervised methods and their counterparts. Unsupervised methods like DualGAN and CycleGAN struggle notably in deblurring, with DualGAN particularly unable to navigate the blur-to-sharp domain, tending instead to bridge the content and color distribution gap between the blurry ($\mathcal{B}$) and sharp ($\mathcal{S}$) datasets. Synthesis-based methods such as BSRGAN and RSBlur also fall short, failing to address unseen blurs, indicating the limitations of augmentation strategies, including those using the *imgaug* library. Supervised method NAFNet fails to handle unseen blurs, often yielding output mostly identical to the blurred inputs. However, our Blur2Blur method effectively transforms unknown blurs into known ones. Our translation process successfully focuses on the blur kernel, minimizing bias from other image characteristics. By integrating Blur2Blur with NAFNet, we achieve a substantial recovery of high-quality sharp images, demonstrating the practical strength of our approach. Additional qualitative results for Restormer can be found in the supplementary material.

4.3. Ablation Study for Blur to Sharp Ratio

We evaluate the significance of the blur-to-sharp ratio, represented as the ratio between datasets Unknown-Blur ($\mathcal{B}$) and Unknown-Sharp ($\mathcal{S}$). Specifically, we consider NAFNet as the image deblurring backbone, and consider the GoPro-RB2V_Street and GoPro-REDS dataset settings, where GoPro represents our target camera device for which we have tailored a deblurring model. We conducted experiments across a range of ratios from 5:5 to 9:1, training the B2B model with each. The deblurring performance is reported in Tab. 3. The result demonstrates that a greater proportion of blurry images in the dataset, as seen in the 6:4 and 7:3 ratios, allows for a deeper understanding of the blur patterns characteristic of the target device. This enhanced knowledge of the blur kernel translates to improved deblurring performance. However, excessively few sharp images, as in the 9:1 ratio, may cause the Blur2Blur method to overfit to limited sharp content, which is used to create the Known-blur image set

	RB2V_Street	REDS	RSBlur
NAFNet [3]			
w/ GoPro	24.78 / 0.714	25.80 / 0.880	26.33 / 0.790
w/ Synthetic Data	22.10 / 0.644	25.07 / 0.853	23.53 / 0.659
w/ Blur2Blur (GoPro)	**26.98** / **0.812**	**28.11** / **0.893**	**29.00** / **0.857**
w/ *the source domain**	28.72 / 0.883	29.09 / 0.927	33.06 / 0.888
Restormer [37]			
w/ GoPro	23.34 / 0.698	25.43 / 0.775	25.98 / 0.788
w/ Synthetic Data	23.78 / 0.655	24.76 / 0.753	23.34 / 0.651
w/ Blur2Blur (GoPro)	25.97 / 0.750	27.55 / 0.885	28.89 / 0.850
w/ *the source domain**	27.43 / 0.849	28.23 / 0.916	32.87 / 0.874
Generalized Deblurring			
BSRGAN [38]	23.31 / 0.645	26.39 / 0.803	27.11 / 0.810
RSBlur [25]	23.42 / 0.603	26.32 / 0.812	26.98 / 0.798
Unpaired Training			
CycleGAN [41]	21.21 / 0.582	23.92 / 0.775	23.34 / 0.782
DualGAN [35]	21.02 / 0.556	23.50 / 0.700	22.78 / 0.704

Table 2. Comparison of different deblurring methods on various datasets. For each test, we report PSNR↑ and SSIM↑ scores as evaluation metrics. The best scores are in **bold** and the second best score are in underline. For a supervised method, NAFNet or Restormer, we assess its upper-bound of deblurring performance by training it on the *training set of the source dataset**.

Ratio $\mathcal{B}:\mathcal{S}$	5:5	6:4	7:3	8:2	9:1
GoPro–RB2V_Street	26.02	26.98	26.92	25.98	24.32
GoPro–REDS	27.53	28.11	28.10	27.00	26.43

Table 3. PSNR debluring results with different Blur-to-Sharp ratios.

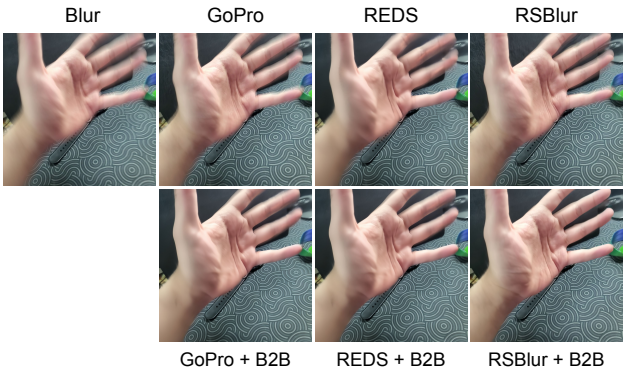


Figure 4. Qualitative comparison of deblurring models on the PhoneCraft dataset with multiple target datasets.

$\mathcal{K}$. To balance learning and prevent overfitting, a 6:4 ratio has been selected for all experiments in this study.

4.4. Practicality Evaluation

We evaluated the practicality of Blur2Blur in two imagined yet realistic scenarios. The first scenario involved a user desiring a deblurring algorithm for images taken with their smartphone camera. To facilitate this, we compiled a dataset named **PhoneCraft**, featuring images captured using a Samsung Galaxy Note 10 Plus. This dataset includes videos

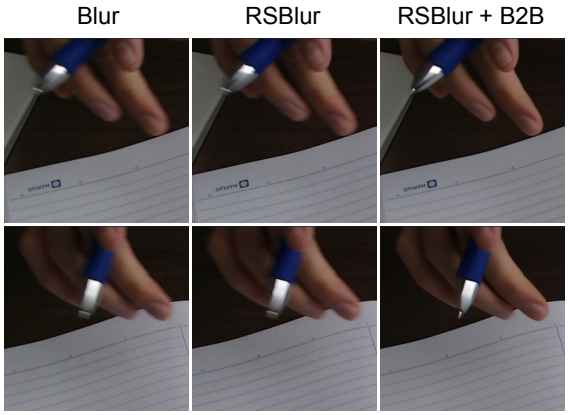


Figure 5. Results of using Blur2Blur on the WritingHands dataset.

with motion-induced blur, refined through post-processing to remove other blur types, and clear, sharp videos recorded at 60fps. Over two hours, a variety of scenes and motions were captured, producing 12 blurry and 11 sharp video clips, each between 30 and 40 seconds long.

In deblurring PhoneCraft images, we used well-known blur datasets GoPro, REDS, and RSBlur. Results in Fig. 4 show Blur2Blur significantly improved image clarity over pre-trained models, especially with RSBlur's complex blur patterns. This demonstrates Blur2Blur's ability to handle real-world blurs effectively.

In our second scenario, we explored a webcam-based application for monitoring hand movements during writing exercises, aimed at assisting in rehabilitation therapy. The challenge here is motion blur, which complicates hand and object tracking. To test our approach, we created a dataset named WritingHands with four 30fps webcam-recorded videos, each about 40 seconds long. From these, two videos provided over 1100 frames with motion blur for training, and one video offered sharp reference images. Leveraging insights from the PhoneCraft dataset, we used the RSBlur dataset and its pre-trained NAFNet model for a two-day training session. Results, shown in Fig. 5, indicate that while RSBlur's model alone leaves some blur, integrating it with Blur2Blur significantly improves image clarity, effectively interpreting the blur kernel and restoring the sharpness.

5. Conclusions

We have proposed Blur2Blur, an effective approach to address the practical challenge of adapting image deblurring techniques to handle unseen blur. The key is to learn to convert an unknown blur to a known blur that can be effectively deblurred using a deblurring network specifically trained to handle the known blur. We substantiated the practical versatility of our method by conducting evaluations with real-world blurry datasets, affirming its role as a versatile deblurring model for general applications. Throughout extensive benchmark experiments, Blur2Blur consistently exhibited superior performance, delivering impressive quan-

titative and qualitative outcomes.

References

[1] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE TIP*, 7(3):370–375, 1998. 2

[2] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *CVPR*, pages 182–192, 2021. 2

[3] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, pages 17–33. Springer, 2022. 2, 3, 6, 8, 1

[4] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021. 2, 5, 6, 1

[5] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016. 4

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 4

[7] Michal Hradiv̊s, Jan Kotera, Pavel Zemcik̊, and Filip v̊Sroubek. Convolutional neural networks for direct text deblurring. In *Proceedings of BMVC*, 2015. 2

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 6

[9] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 5

[10] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, Franccois-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020. 6

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[12] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *NeurIPS*, 22:1033–1041, 2009. 2

[13] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR 2011*, pages 233–240. IEEE, 2011. 2

[14] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jivr´i Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *CVPR*, 2018. 2, 3

[15] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1964–1971. IEEE, 2009. 2

[16] Chih-Hung Liang, Yu-An Chen, Yueh-Cheng Liu, and Winston H Hsu. Raw image deblurring. *IEEE Transactions on Multimedia*, 24:61–72, 2020. 2

[17] Guangcan Liu, Shiyu Chang, and Yi Ma. Blind image deblurring using spectral properties of convolution operators. *IEEE Transactions on image processing*, 23(12):5047–5056, 2014. 2

[18] Boyu Lu, Jun-Cheng Chen, and Rama Chellappa. Unsupervised domain-specific deblurring via disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10225–10234, 2019. 2, 3, 4

[19] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 6

[20] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 2, 3, 5, 6, 1

[21] Seungjun Nah, Radu Timofte, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 3, 5, 6, 1

[22] Bang-Dang Pham, Phong Tran, Anh Tran, Cuong Pham, Rang Nguyen, and Minh Hoai. Hypercut: Video sequence from a single blurry image using unsupervised ordering. *arXiv preprint arXiv:2304.01686*, 2023. 3, 5, 6, 1

[23] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3341–3350, 2020. 2

[24] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*. Springer, 2020. 3

[25] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 487–503. Springer, 2022. 3, 6, 8

[26] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 5, 6, 1

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3, 1

[28] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, 2019. 6

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[30] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *CVPR*, pages 3606–3615, 2020. 2

[31] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, pages 8174–8182, 2018. 2

[32] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018. 2

[33] Dmitrii Torbunov, Yi Huang, Huan-Hsin Tseng, Haiwang Yu, Jin Huang, Shinjae Yoo, Meifeng Lin, Brett Viren, and Yihui Ren. Rethinking cyclegan: Improving quality of gans for unpaired image-to-image translation. *arXiv preprint arXiv:2303.16280*, 2023. 6

[34] Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *CVPR*, 2021. 3, 5, 6

[35] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017. 2, 3, 4, 6, 8

[36] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021. 2, 3

[37] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022. 2, 6, 8, 1

[38] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021. 3, 6, 8

[39] Suiyi Zhao, Zhao Zhang, Richang Hong, Mingliang Xu, Yi Yang, and Meng Wang. Fcl-gan: A lightweight and real-time baseline for unsupervised blind image deblurring. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6220–6229, 2022. 2, 3, 4

[40] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*. Springer, 2020. 3

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2, 3, 4, 6, 8

Blur2Blur: Blur Conversion for Unsupervised Image Deblurring on Unknown Domains

Supplementary Material

Abstract

In this supplementary PDF, we first provide the qualitative results obtained by methods with the Restormer backbone [37] and some additional qualitative results of each dataset to show our effectiveness in deblurring unknown-blur images compared to other baselines. Next, we illustrate the performance with different backbones for the Blur2Blur translator model. Finally, we provide details of our collected PhoneCraft dataset and validate the video deblurring performance of Blur2Blur, demonstrating significant enhancements in hand movement visualization and thus leaving room for practical application. We also include our code and a video of sample deblurring results in the supplementary package.

6. Additional Qualitative Results

6.1. Restormer model

In Fig. 3 in the main paper, we omit the results with the Restormer backbone due to the space limit. We provide these results in this supplementary in Fig. 6. As can be seen, Restormer shows behavior similar to NAFNet. The original network produces blurry images that are close to the input images. However, when combined with Blur2Blur, it can successfully deblur the images and produce sharper outputs. From quantitative numbers, Restormer-based models perform slightly worse than the NAFNet-based counterparts.

6.2. Additional Deblurring Results

In this section, we provide additional qualitative figures comparing the image deblurring results of our Blur2Blur and other baselines. Figures 7, 8, and 9 show samples where $\mathcal{K}$ is built upon the GoPro dataset [20], with the Unknown set derived respectively from the REDS dataset [21], RB2V_Street [22], and RSBlur [26].

7. Blur2Blur Backbone Analysis

We explore the integration of multi-scale architectures into the Blur2Blur mechanism by experimenting with different backbones. The UNet architecture [27] has been adapted to handle inputs at various scales, allowing for a more nuanced understanding of blur at multiple scales. Concurrently, we employed the NAFNet backbone in its original form, taking advantage of its robust feature extraction capabilities without modifications. The result on Tab. 4 shows that MIMO-UNet clearly surpasses the standard UNet, even in its modified form. Moreover, the results also reveal that the NAFNet does not perform as well as the multi-scale variants, highlighting the importance of multi-scale level optimization in the Blur2Blur framework for deblurring tasks.

Backbone	PSNR↑	SSIM↑
UNet [27]	22.54	0.732
MIMO-UNet [4]	26.98	0.812
NAFNet [3]	20.54	0.686

Table 4. Ablation studies with the Blur2Blur backbone.

8. Real-world Application

8.1. Details of PhoneCraft collection

The data collection process for training Blur2Blur is actually inexpensive. Although the number of images required looks high (several thousand for each subset), they are mostly video frames and thus can be collected effectively. For example, in the PhoneCraft experiment above, we only need to collect 11 sharp videos and 12 blurry ones, with a total collection time of less than 2 hours. More specifically, the dataset contains more than 12500 diverse blurry images and 11000 sharp images.

8.2. Video Deblurring Performance

As mentioned in the main paper, to enrich our practical evaluation with more tangible visual examples and to demonstrate one real-world application of our Blur2Blur mode, we incorporated a video from the collected dataset. This video simulates scenarios with significant motion blur, which is common in dynamic environments. The clarity of visual details in such situations is crucial for various applications, including rehabilitation therapy. Accurate hand movement visualization is vital for tasks like hand pose detection and gesture-based interactive rehabilitation systems.

To evaluate our Blur2Blur model, we used a video with pronounced hand movements, pre-training the deblurring model on the RSBlur dataset. The results, demonstrated in video1.mp4, clearly show that our Blur2Blur framework significantly enhances visual clarity compared to using the pre-trained deblurring model alone. Moreover, to further assess the enhancement in hand movement recognition, we validated the deblurred videos using the Hand Pose Estimation model from MediaPipe[1]. The results, shown in the

[1]https://developers.google.com/mediapipe

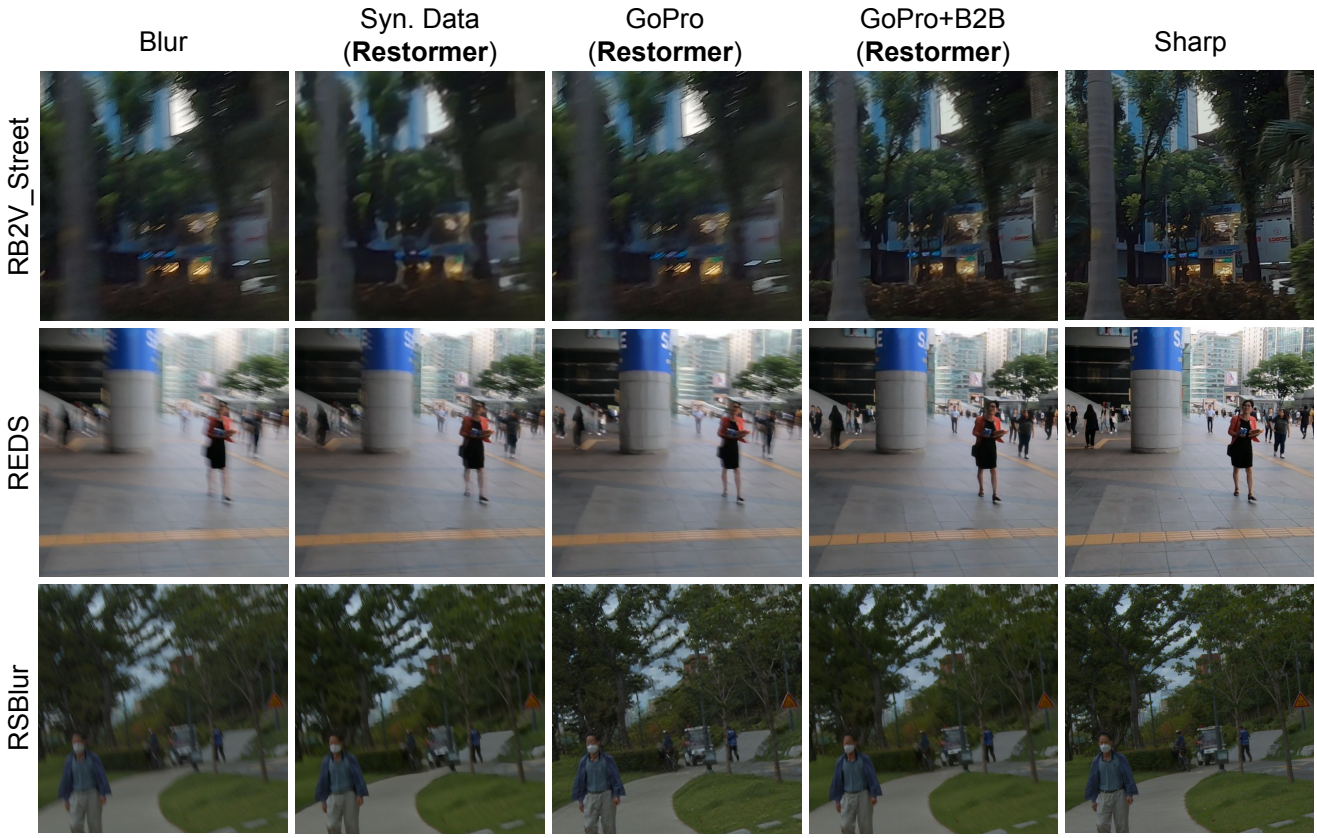


Figure 6. Qualitative results of Restormer [37] on three datasets.

video, highlight a notable improvement in hand pose estimation when using our method. The enhanced sharpness and detail achieved by Blur2Blur enable more accurate and reliable recognition of hand poses. This demonstrates the potential of our Blur2Blur model in applications demanding high-fidelity visualization of hand movements, especially in advanced rehabilitation therapy tools that rely on precise hand movement tracking for effective patient care and recovery.

Besides that, we also provide the additional qualitative video deblurring result in PhoneCraft dataset is illustrated in video2.mp4.

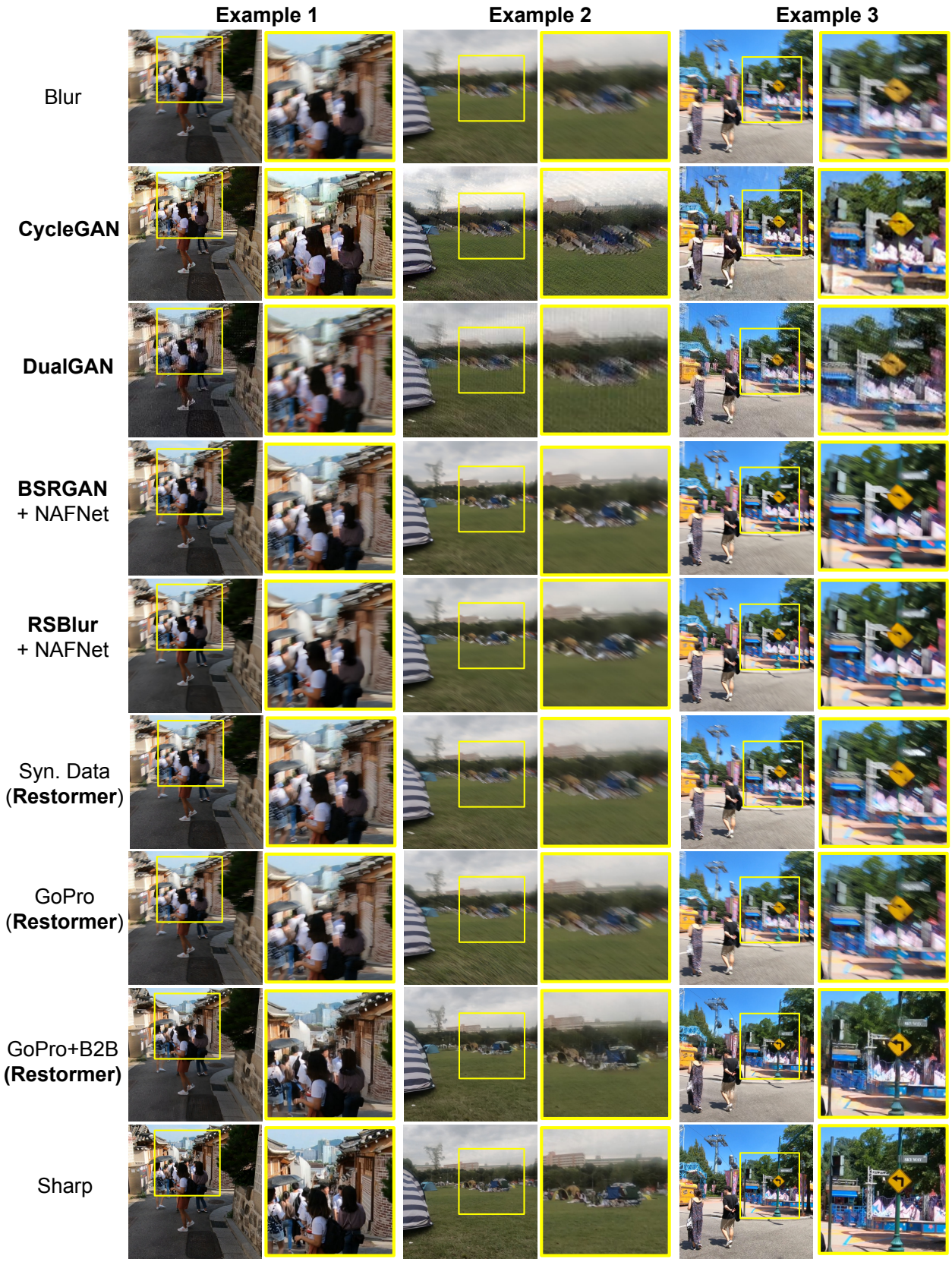


Figure 7. Extra qualitative results on the REDS dataset.

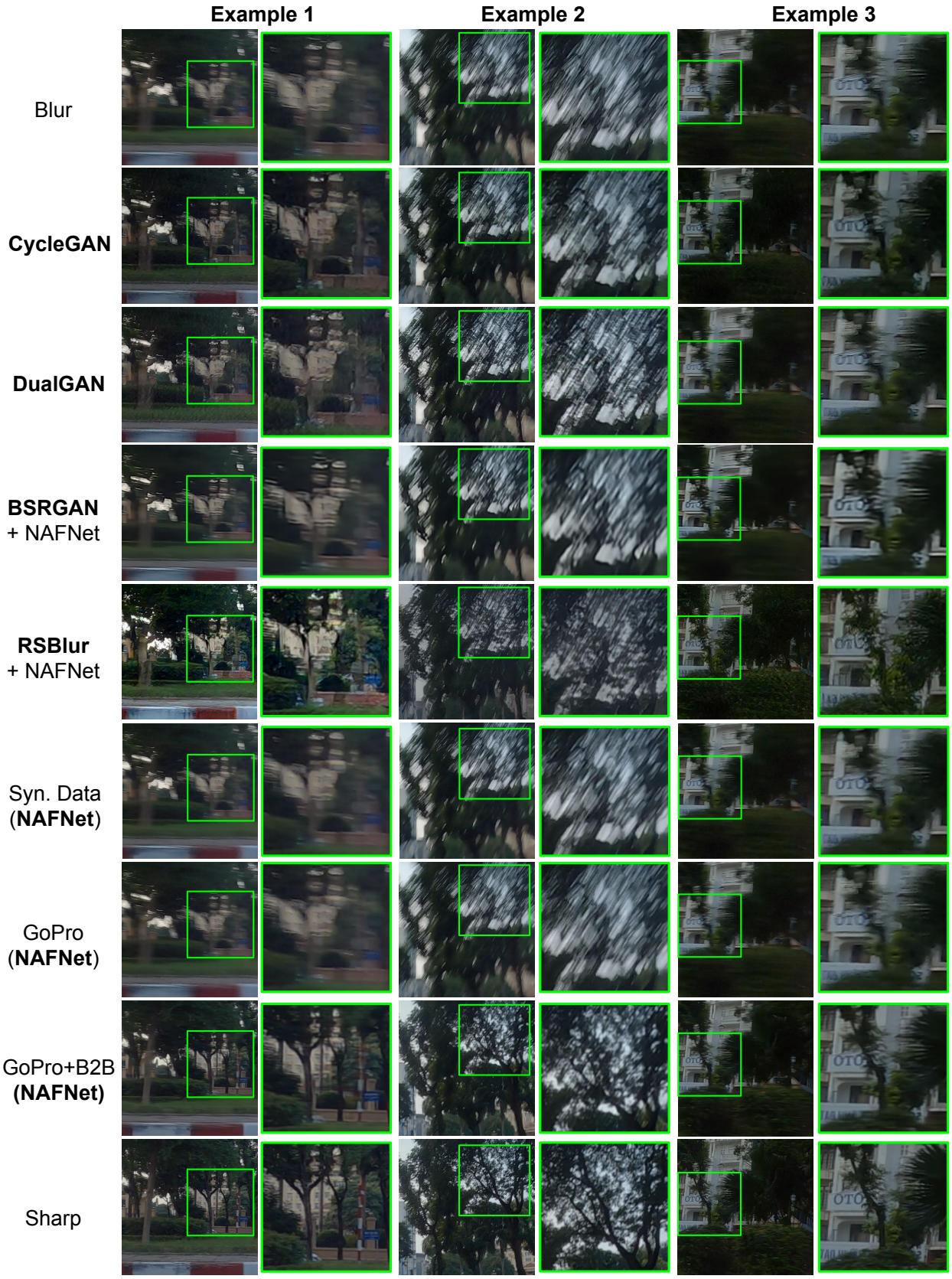


Figure 8. Extra qualitative results on the RB2V_Street dataset.

Figure 9. Extra qualitative results on the RSBlur dataset.