Non-Uniform Blind Motion Deblurring using an Algorithm Unrolling Neural Network

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

11-01-2022 / 19-01-2022

CITATION

Richmond, Greig (2022): Non-Uniform Blind Motion Deblurring using an Algorithm Unrolling Neural Network. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.18133175.v1>

DOI

[10.36227/techrxiv.18133175.v1](https://doi.org/10.36227/techrxiv.18133175.v1)

Non-Uniform Blind Motion Deblurring using an Algorithm Unrolling Neural Network

Greig Richmond, Arlene Cole-Rhodes *Senior Member, IEEE*

Abstract— In this work we develop a compact neural network that is designed to deblur images that have been affected by a non-uniform blur. We develop this network by unrolling a traditional iterative image deblurring algorithm and adapt it to independently deblur regions of an image. The network is evaluated by comparing its deblurring capabilities with that of other state-of-the-art networks, such as the SRN-Deblur network [13] and the DUBLID network [4]. We investigate the effect that varying the patch size and the size of the point spread function has on the deblurring performance of our network. We evaluate each deblurring network using the industry standard GOPRO [12] and Kohler [25] datasets.

Index Terms— Algorithm unrolling, Convolutional neural network, Image deblurring, Image restoration, Non-uniform blur

1 INTRODUCTION

Algorithm unrolling attempts to bridge the gap between iterative optimization methods and neural networks. It can be used to build efficient interpretable networks, which tend to be computationally faster with a lower number of parameters than the traditional neural networks. The technique of algorithm unrolling was originally developed by K. Gregor and Y. LeCun, [32] to solve sparse coding algorithms. It has also been shown by Li et al. [4] to be effective for image deblurring algorithms, and by Yang et al. [14] to be effective for image super-resolution algorithms. The algorithm unrolling approach for blind image deblurring was first applied in [4] and the developed network was observed to be effective, interpretable, and compact (with a low number of parameters). That work however was tailored to deblur images with a uniform blur and it did not perform as well when dealing with a spatially variant blur.

Non-uniform blur can affect real-world images in cases, such as the blur caused by camera shake, in-and-out of focus objects in a scene, or blur caused by the motion of an object within a stationary scene. Traditional algorithms that remove non-uniform blur can be slow to reach an ideal solution, and convolutional neural network (CNN) approaches are often only practical on powerful computers with ample resources. In this paper we develop a CNN that tailors and unrolls a traditional image deblurring algorithm for images with a non-uniform blur. Our aim is to keep this new approach compact, while still being effective in deblurring images.

Digital cameras are installed on a wide variety of computing devices including desktop computers, mobile phones, surveillance equipment, automobiles, and robots. The computational capabilities of these devices span a wide range from stationary systems with a connected power supply that will often have access to powerful processors and increased memory, to mobile devices or devices in remote locations which will often have more limited hardware. Images taken by camera devices may suffer from non-uniformly blurred content, such as motion blur or camera shake, and this must be removed prior to

other applications such as object segmentation, detection, and/or classification. It is beneficial for the software on a resource constrained system to be able to deblur images locally since it may not always be possible for these images to be sent to a remote computer for processing. As the image processing software will only be given a blurred image without context of the true scene, the process of removing the blur is called blind image deblurring. The presence of non-uniform blur adds complexity to the basic blind deblurring problem.

The typical blind deblurring algorithm estimates the latent sharp image, X from the given the blurred image, Y assuming the image has been blurred based on the linear model

$$Y = K * X + N \quad (1)$$

where K is the point spread function (PSF), $*$ is the convolution operator, and N is additive white Gaussian noise (AWGN). This models an image that has been uniformly blurred, and so the PSF is the same at all locations in the image. Non-uniform blur due to motion of an object in frame, or due to camera shake, can cause the observed blur to vary spatially across the image. This type of blur cannot be accurately modeled with (1). Papers that address the non-uniform deblurring problem ([1], [2], [3]) typically formulate the blurring model in matrix form

$$\underline{y} = A\underline{x} + \underline{n} \quad (2)$$

where $\underline{y}$ and $\underline{x}$ are the column vector forms of Y and X , and A is an image filtering matrix used to compute a discrete convolution which results in a blurred image. The rows of A contain the PSF at each pixel location unraveled into a row vector. For an image, X of dimension $H \times W$, A is a square matrix of dimension $(H \times W) \times (H \times W)$. For a uniform blur, each row of A contains the same values with a spatial shift from row to row, but for a non-uniform blur the rows of A do not have the same values since the blur varies spatially.

When only the blurred image is provided, the

problem as modeled in both (1) and (2) is ill-posed as both the PSF, K and the latent sharp image, X are unknown. To aid in solving the blind image deblurring problem, assumptions about the sources of the blur and prior information about a sharp image are exploited. Assumptions about the sharp image in recent works include (a) an image gradient sparsity priors ([4], [5]), based on the fact that the gradient map of a sharp image is generally sparse; (b) a dark channel prior used by Pan et al. [6], which takes advantage of the fact that a sharp image has significantly more regions where pixel values across all color channels are close to zero; and (c) a minimal pixel intensity prior used by Wen et al. [7], which is similar to the dark channel prior except rather than using a sliding window that samples overlapping patches, non-overlapping patches are used. In both [6] and [7] it was observed that the minimal pixel within a patch in the sharp image would be higher in value for a blurred image than its sharp counterpart since the blur usually has a smoothing effect. PSF models for non-uniform blurring include, but are not limited to, camera motion models ([8], [9]), depth variation models [10], and other specific types of blur ([1], [2], [11]).

Neural networks have been successfully applied to solving the non-uniform image deblurring problem, and they have become more prevalent over the years with the increase in available memory and computing power. Sun et al. [14] developed a neural network that was used to estimate PSFs in images with non-uniform blur by using a Markov random field model, and they then use a traditional iterative method to recover the latent image. An end-to-end multi-scale convolutional neural network (CNN) was developed by Nah et al. [12] that was shown to be highly capable of recovering the latent sharp image without determining the blurring PSFs used. Tao et al. [13] modified this work to produce a scale-recurrent CNN that performed better in most cases, while also using fewer parameters overall. CNN-based solutions have been shown to be versatile and effective, as they are able to recover images suffering from multiple types of blur, and they will also recover higher quality images as measured by the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics ([4], [12], [13]).

While these and other CNN methods have been able to recover high quality images, these networks often come with a heavy computational price. Modern image processing CNNs are often built using millions of floating-point parameters which require the use of millions of computations. To make these networks feasible, they often require GPU acceleration which can limit their use in small or mobile devices. Storing these parameters and intermediate feature maps in memory also comes with a price, as the millions of parameters can take up a sizeable amount of space and GPU memory for processing can also be limited. CNN methods are also sometimes considered black boxes, where the input is processed by a set of filters, whose meaning or purpose may be unclear. Traditional methods (not involving neural networks) are interpretable because the parameters are defined in the development of the algorithm. However, such traditional

methods often require lengthy run times to achieve high levels of performance [4]. A combination of iterative algorithms and neural networks is seen in algorithm unrolling, where each iteration of a traditional algorithm is made into a layer of the neural network and algorithm specific parameters are learned through training the network.

2 RELATED WORK

Neural networks have been tailored to solve many aspects of the blind image deblurring problem. Sun et al. [14] developed a network that iteratively estimated the PSFs, which had caused the distortion in an image with a non-uniform motion blur. The blurred input image was divided into 30x30 overlapping patches and a local blur was estimated in each patch by a trained CNN. The CNN was used to classify the region as being blurred by one of 73 different PSFs that represented motion blur in varying directions or no blur (a delta function). The blur estimates for each patch were then combined to give a motion “blur field” using a Markov random field model of the estimates. The true image was finally recovered using a traditional iterative algorithm based on the estimates of the PSFs. Although the CNN used was relatively small (~12,000 parameters), it was limited in that it only classified magnitude and orientation of motion blur. While the final step of estimating the latent sharp image was then solved through iteration, this adds to the overall time complexity of the technique.

Modern neural networks are often trained to perform the complete end-to-end task of deblurring an image. Nimisha et al. [15] and Chen et al. [16] use a generative adversarial network (GAN) to train a CNN to deblur images. A GAN consists of a pair of neural networks, a generator and a discriminator. The discriminator is given a set of data with some common characteristics and it is trained to determine whether the new input data is similar to the set or is not. The generator is trained to generate data that meets the characteristics of the set of data to trick the discriminator. The generator and a discriminator of a GAN are used to help train and improve each other. The GAN discriminator in [15] is used to judge features extracted by the encoder of an autoencoder style network, while [16] uses the discriminator to judge whether the image is a deblurred image or an original image. The generator and decoder used for deblurring in [15] is smaller than a typical deep learning image deblurring network with a slightly over 850,000 parameters so it can be considered as a compact network, while the Inception-resnetnet network of [16] contains over 56,000,000 parameters and it can be considered large.

Nah et al. [12] also solved the non-uniform motion blur problem by developing a scale-based end-to-end CNN. This network processed the blurry input image at three scales (quarter-size, half-size, and original size) in a coarse-to-fine manor. Each scaled image passed through an auto-encoder style network where the output at a one scale is concatenated to the input at the next finer scale of processing. Tao et al. [13] improved and condensed this

scale-based network by adding a recurrent connection between bottleneck layers for each scale and reusing network parameters. The recurrent connection was used to share scale-based contextual information from coarser scales to finer scales. Since the weights for processing each scale of the input image are the same, the network is trained to deblur the image at each scale while using information learned at the coarser scale to improve the result at the finer scales. While Nah et al. [12] used a GAN model for training, Tao et al. [13] noted that the performance of their network did not improve much when compared to training with the commonly used L_2 loss. These scale-based approaches can successfully deblur images with any type of non-uniform blur but they both contain millions of parameters.

There are several techniques that can reduce the number of parameters in a neural network, but these methods can also reduce the deblurring capabilities of the networks. The following methods have been used for neural network compression as indicated in the survey paper by Zhang et al. [21]: (i) Parameter Quantization, where the exact value of a network parameter is truncated from a floating-point number to an integer; (ii) Binarization, where the network parameters are reduced to either being one or zero; (iii) Parameter Pruning where the unused, unnecessary, or ‘low impact’ network parameters are discarded; (iv) Parameter Sharing where frequently used network parameters can be shared when needed in attempt to save space; (v) Knowledge Distillation where a larger network is used to help train a smaller network. In our previous work ([22]), filter compression was used as a method for neural network compression, where the number of network parameters of the filters are reduced by using a principal component filter compression technique.

In contrast to the large number of parameters required by neural networks, traditional optimization-based deblurring algorithms have fewer parameters [4]. It is a difficult task to determine both the latent sharp image and the PSF simultaneously, so it can be beneficial to estimate the PSF first then use a non-blind deconvolution algorithm to recover the latent sharp image [17]. But one well-known approach to solve the blind image deblurring problem is to estimate both the PSF, K and the latent sharp image, X simultaneously using a Maximum a posteriori (MAP) approach. The MAP approach aims to find a pair (X, K) that maximizes the probability distribution

$$p(X, K|Y) = p(Y|X, K)p(X)p(K) \quad (3)$$

The likelihood term, $p(Y|X, K)$ is typically given by the estimates of X and K that minimizes the error given by $\|K * X - Y\|_2^2$. The probability distributions $p(X)$ and $p(K)$ can be estimated by priori information about the sharp image and PSF respectively ([17], [18], [19]). The blind image deblurring problem can then be formulated as

$$\underset{\{X, K\}}{\text{Minimize}} \{ \|K * X - Y\|_2^2 + \lambda J(X) + \gamma Q(K) \} \quad (4)$$

where λ and γ are non-negative parameters, J and Q are

functions that represent estimated a priori information about X and K respectively, and $\|\cdot\|_p$ denotes the l_p -norm. The likelihood portion of the cost function in (4) is often specified in the image gradient domain ([4], [20]), as it has been shown that the prominent edges in the image gradient aid in deblurring. The square of the l_2 -norm of the PSF, K is often used for $Q(K)$ unless additional information is known about the PSF.

There have been many different choices for an image prior, $J(X)$ (e.g., The dark channel prior used by Pan et al. [6] and the minimal intensity prior used by Wen et al. [21]). The pixel values in the image gradient for a sharp image are known to have a small sum, so the image prior term forces the estimated image gradient to have a low sum of pixel values. This can be accomplished with a total-variation (TV) regularizer (l_1 -norm) regularizer on the image estimate ([4], [18], [20]).

The MAP optimization problem can be solved iteratively using an alternating minimization technique [4, 6, 20, 21]. Li et al. [4] unrolled an iterative algorithm in a neural network where the parameters were learned through training. Their approach also increased the number of filters typically used for gradient domain processing; so instead of known horizontal and vertical derivative filters (e.g., Sobel or Prewitt filters), the parameters for a larger set of filters was learned through network training. The unrolled total-variation deblurring algorithm [4] was shown to be highly effective at deblurring images with a uniform blur, but it did not perform as well deblurring images with a non-uniform blur. This neural network approach was highly compact however, as it contained approximately 23,000 parameters.

3 NON-UNIFORM DEBLURRING ALGORITHM

3.1 Model of Spatially-Variant Blur

A spatially variant or non-uniform blur cannot be as accurately estimated in the same way as a uniform blur since the blurring PSF in each location can be different. Thus, using the model of (2) each region can have its own unique PSF to create a non-uniformly blurred image. Hirsch et al. [23] show that observed non-uniform blur in a real image can be accurately modeled by considering the blur as locally invariant.

A patch-based convolution process divides an image into P overlapping patches, damps the values of borders of the patches using a windowing function w , convolves each patch with a PSF, and then returns the patch to its original location. If each patch is blurred with the same PSF, a uniformly blurred image will be created. If the PSF for each patch is different, this process will create a non-uniform blur. Using this approach, we can blur an image with different PSF combinations to create a spatially variant blur. By making use of the discrete Fourier transform (DFT) we can modify the blur model (1) to apply to a non-uniform blurred image as follows:

$$Y = \sum_{r=1}^P \mathcal{M}_r^{-1} \left(\mathcal{F}^{-1} \left(\mathcal{F} \left(Z_K(K^{(r)}) \right) \mathcal{F} \left(\mathcal{M}_r(w^{(r)}X) \right) \right) \right) \quad (5)$$

where $w^{(r)}$ is a windowing function for patch $r \in [1, P]$, $\mathcal{M}_r$ is an operator to chop the r -th patch from an image, $\mathcal{M}_r^{-1}$ is an operator that returns the patch to the image, $\mathcal{F}$ is the discrete Fourier transform (DFT), $\mathcal{F}^{-1}$ is the inverse DFT. Z_k is a zero-padding operator that appends zeros to $K^{(r)}$ (the PSF for patch r) so that its size matches the corresponding image patch size. Figure 1 shows the windowing, patch isolation, and return process that occurs in the alternative blur model specified by (5).

Fig. 1. Example of Windowing operation, and Patch operations which isolate and return a patch, r to an image

The windowing function, $w^{(r)}(i, j)$ for patch, r plays an important role in this process and needs to be chosen correctly. For any overlap in the image patches, the sum of the weights in the window function for each patch must be equal to one, otherwise there will be artifacts in the resultant image ([6], [23]). For our windowing function we choose a Bartlett-Hanning window with 50% overlap, which was developed by Ha and Pearce [24] and also used by Hirsch et al. [23]. In 2D the windowing function is given by

$$w(i, j) = \left(0.62 - 0.48 \left| \frac{i}{N_i} - 0.5 \right| - 0.38 \cos \left(\frac{2\pi i}{N_i} \right) \right) * \left(0.62 - 0.48 \left| \frac{j}{N_j} - 0.5 \right| - 0.38 \cos \left(\frac{2\pi j}{N_j} \right) \right) \quad (6)$$

where N_i and N_j is the length of the filters in the i and j direction respectively. Note that i and j are the indices within the input image.

3.2 Blind Non-Uniform Image Deblurring

3.2.1 Problem Development

A uniform deblur can efficiently be removed using a total-variation (TV) deblurring algorithm as shown in ([4], [18], [20]) and this is summarized here. The deblurring approach is formulated using the MAP approach specified in (3) and (4). The problem is formulated in the image gradient domain, as edge features are prominent in the image gradient. In the gradient domain, the blur model in (1) becomes

$$\nabla Y = K * \nabla X + \nabla N \quad (7)$$

where ∇ is the gradient operator. Working in the gradient domain, we specify the function $J(X)$ to be a TV regularizer for X , and specify the function $Q(K)$ to be the square of the l_2 -norm. The deblurring problem in (4) becomes

$$\text{Minimize}_{\{K, G_1, G_2\}} \left\{ \frac{1}{2} \left(\|D_x Y - K * G_1\|_2^2 + \|D_y Y - K * G_2\|_2^2 \right) + \lambda_1 \|G_1\|_1 + \lambda_2 \|G_2\|_1 + \frac{\epsilon}{2} \|K\|_2^2 \right\} \quad (8)$$

subject to: $\|K\|_1 = 1$ and $K \geq 0$

where, D_x and D_y are derivative operators in the horizontal and vertical plane respectively, λ_1 , λ_2 , and ϵ are positive constant parameters, G_1 and G_2 are estimates of the gradient of the target sharp image (i.e., estimates of $D_x X$ and $D_y X$ respectively), and the $\geq$ symbol is to be interpreted as elementwise. The first two l_2 -norms in (8) restrict the solution to be generally similar to the blurred image while the l_1 -norm components restrict the introduction of jitter in the solution by limiting the total variation in the gradient domain. The final l_2 -norm penalizes large estimates of the PSF.

It was noted in [4] that the derivative operators D_x and D_y are commonly implemented using filters that produce the derivative in orthogonal directions. Several works ([28], [29]) show that it is useful to apply multiple types of filters to extract different types of information from the blurred image and enhance the representation of sparsity. Thus, the derivative filters in (8) are generalized in [4] to filter the images through a set of C filters, $\{f_i\}_{i=1}^C$ by

$$\text{Minimize}_{\{K, \{G_i\}_{i=1}^C\}} \left\{ \sum_{i=1}^C \left(\frac{1}{2} \|f_i * Y - K * G_i\|_2^2 + \lambda_i \|G_i\|_1 \right) + \frac{\epsilon}{2} \|K\|_2^2 \right\} \quad (9)$$

subject to: $\|K\|_1 = 1$ and $K \geq 0$

where G_i , for $i = 1, 2, \dots, C$ are estimates of the filtered image. The optimization problem in (9) is non-smooth but can be solved with a half-quadratic splitting technique as described in [30]. This technique splits the problem by introducing auxiliary variables, Z_i and regularization parameters ζ_i , for $i = 1, 2, \dots, C$ to move G_i out of the non-differentiable l_1 -norm term. The problem in (9) then becomes

$$\text{Minimize}_{K, \{G_i, Z_i\}_{i=1}^C} \left\{ \sum_{i=1}^C \left(\frac{1}{2} \|f_i * Y - K * G_i\|_2^2 + \lambda_i \|Z_i\|_1 \right) + \frac{1}{2\zeta_i} \|G_i - Z_i\|_2^2 + \frac{\epsilon}{2} \|K\|_2^2 \right\} \quad (10)$$

subject to: $\|K\|_1 = 1$ and $K \geq 0$

Once the problem is split, optimization can be performed by fixing all but one term and optimizing this term, and similarly repeating the optimization process for all other terms. It is noted in [30] that as the $\zeta_i \rightarrow 0$ that the cost function in (10) converges to that in (9).

3.2.2 Algorithm

A solution to (10) can be obtained by alternately minimizing over $\{G_i\}_{i=1}^C$, $\{Z_i\}_{i=1}^C$ and K until convergence, as shown in [4] and [20]. At the l -th iteration, the sub problems are

$$G_i^{l+1} \leftarrow \text{Minimize}_{\{G_i^l\}} \left\{ \frac{1}{2} \|f_i^l * Y - K^l * G_i^l\|_2^2 + \frac{1}{2\zeta_i^l} \|G_i^l - Z_i^l\|_2^2 \right\}, \quad \text{for } 1 \leq i \leq C \quad (11)$$

$$Z_i^{l+1} \leftarrow \text{Minimize}_{\{Z_i^l\}} \left\{ \frac{1}{2\zeta_i^l} \|G_i^l - Z_i^l\|_2^2 + \lambda_i^l \|Z_i^l\|_1 \right\}, \quad \text{for } 1 \leq i \leq C \quad (12)$$

$$K^{l+1} \leftarrow \underset{\{K^l\}}{\text{Minimize}} \sum_{i=1}^C \left(\frac{1}{2} \|f_i^l * Y - K^l * G_i^{l+1}\|_2^2 \right) + \frac{\epsilon}{2} \|K^l\|_2^2$$

subject to: $\|K\|_1 = 1$ and $K \geq 0$ (13)

When the algorithm is ‘unrolled’, the optimization sub-problems (11), (12), and (13) are solved at each layer, l of the resulting neural network and the parameters ζ_i and λ_i and the coefficients of the filters, f_i^l are learned through training the network. As shown in (11) and (12), the parameters $\{\zeta_i\}_{i=1}^C$ and $\{\lambda_i\}_{i=1}^C$ are different at each iteration of the algorithm (i.e., each layer of the network). Each iteration/layer, l also has a different set of C filters, $\{f_i^l\}_{i=1}^C$ to filter the blurred image. We will denote the filtered versions of the image, Y as Y_i^l when they have been filtered by f_i^l to be concise.

$$Y_i^l = \begin{cases} f_i^l * Y, & l = L \\ f_i^l * Y_i^{l-1}, & 1 < l < L \end{cases}, \text{ for } 1 \leq i \leq C \quad (14)$$

Sub-problem (11) in a patch-based model is solved with the by taking of Fourier transforms of (11), taking the derivative with respect to G_i^l and setting the result equal to zero, using the same notation as in (5) we obtain

$$G_i^{l+1} \leftarrow \sum_{r=1}^P \mathcal{M}_r^{-1} \left(\mathcal{F}^{-1} \left(\frac{\zeta_i^l \mathcal{F}^* \left(Z_K(K^{l(r)}) \right) \mathcal{F} \left(\mathcal{M}_r(w^{(r)} Y_i^l) \right) + \mathcal{F} \left(\mathcal{M}_r(Z_i^l) \right)}{\zeta_i^l \mathcal{F}^* \left(Z_K(K^{l(r)}) \right) \mathcal{F} \left(Z_K(K^{l(r)}) \right) + 1} \right) \right), \right.$$

for $1 \leq i \leq C$ (15)

where $\mathcal{F}^*$ is the complex conjugate of the DFT, and $w^{(r)}$ is a windowing function for patch r as was specified in (6), and P is the total number of patches. The arithmetic operations in (14) and in subsequent problems are to be interpreted as elementwise unless explicitly stated.

Sub-problem (12) is a proximal mapping problem applied to each patches updated filtered image estimate and its solution given as

$$Z_i^{l+1} \leftarrow \sum_{r=1}^P \mathcal{M}_r^{-1} \left(\text{sgn} \left(\mathcal{M}_r(G_i^l) \right) \max \{ (|\mathcal{M}_r(G_i^l)| - \zeta_i^l \lambda_i^l), 0 \} \right),$$

for $1 \leq i \leq C$ (16)

where $\text{sgn}(\cdot)$ is sign of the element (i.e., +1 or -1). The $\max$ operator can be implemented within neural network software with the rectified linear unit (ReLU) function. This is a thresholding operation applied to the new estimates of each patch in G_i^l .

The solution for sub-problem (13) is approximated in three steps. First the unconstrained optimization problem is solved in (17), and then the result is manipulated to satisfy the constraints in (18) and (19). Once again, we use the patch-based convolution model from (5) and the DFT in a manner similar to the solution of (11) to solve the unconstrained version of (13) to obtain

$$\bar{K}^{l+1} \leftarrow \sum_{r=1}^P \mathcal{F}^{-1} \left(\frac{\sum_{i=1}^C \left(\mathcal{F}^* \left(\mathcal{M}_r(Z_i^{l+1}) \right) \mathcal{F} \left(\mathcal{M}_r(w^{(r)}(Y_i^l)) \right) \right)}{\sum_{i=1}^C \left(\mathcal{F}^* \left(\mathcal{M}_r(Z_i^{l+1}) \right) \mathcal{F} \left(\mathcal{M}_r(Z_i^{l+1}) \right) \right) + \epsilon} \right) \quad (17)$$

The non-negativity constraint is enforced by applying a Softmax thresholding operation (log-sum-exp) and is applied to all patches with

$$\bar{K}^{l+1} \leftarrow \sum_{r=1}^P \mathcal{M}_r^{-1} \left(\max \left[\left(\mathcal{M}_r(\bar{K}^l) - B^l \log \left(\sum_i \exp \left(\mathcal{M}_r(\bar{K}^l) \right) \right) \right), 0 \right] \right) \quad (18)$$

where B^l is a non-negative parameter which weights the thresholding of PSF coefficients, $\bar{K}^l$. The unit sum constraint is enforced by normalizing the result of (18) by

$$K^{l+1} \leftarrow \sum_{r=1}^P \mathcal{M}_r^{-1} \left(\frac{\mathcal{M}_r(\bar{K}^{l+1})}{\mathcal{M}_r(\|\bar{K}^{l+1}\|_1)} \right) \quad (19)$$

In blind image deblurring literature, a coarse-to-fine scheme is often used with large filters in the early stages of the algorithm and a transition to smaller filters as the algorithm approaches convergence. As shown in [31], this can lead to a large number of parameters, but smaller filters can be used to obtain the same effective receptive field when they are used in consecutive layers. Thus, to achieve the same effect, the authors of [4] filtered the blurred image with L sets of 3x3 filters saving the feature maps after each convolutional layer. The image after the L -th set of filters is used as input to the first layer of the unrolled network. The input to layer l is the filtered output from $(L - l + 1)$ sets of consecutive filters. We apply the same approach in our patch-based approach by filtering each image patch.

After completing L iterations of the algorithm (passing through L layers of the neural network), we obtain an estimate of the PSF at each image patch, K^L and an estimate of the filtered sharp image, G^L . Using these estimates, we can then estimate the latent sharp image by solving

Fig. 2. Flow diagram for the proposed image deblurring algorithm

$$X \leftarrow \underset{\{X\}}{\text{Minimize}} \left\{ \frac{1}{2} \|Y - K^L * X\|_2^2 + \sum_{i=1}^C \left(\frac{\eta_i}{2} \|f_i^L * X - G_i^L\|_2^2 \right) \right\} \quad (20)$$

where the $\{\eta_i\}_{i=1}^C \in \mathbb{R}_+$ are regularization parameters. The solution to (20) in a patch-based image model is obtained by again making use of the DFT as

$$X \leftarrow \sum_{r=1}^P \mathcal{M}_r^{-1} \left(\mathcal{F}^{-1} \left(\frac{\mathcal{F}^* \left(Z_K(K^L(r)) \right) \mathcal{F} \left(\mathcal{M}_r(w(r)(Y)) + \sum_{i=1}^C \eta_i \mathcal{F}^* (f_i^L) \mathcal{F} (\mathcal{M}_r(w(r)(Z_i^L)) \right) \right)}{\mathcal{F}^* \left(Z_K(K^L(r)) \right) \mathcal{F} \left(Z_K(K^L(r)) \right) + \sum_{i=1}^C \eta_i \mathcal{F}^* (z_f(f_i^L)) \mathcal{F} (z_f(f_i^L))} \right) \right) \quad (21)$$

Through experimentation [4], the number of layers (L) in the uniform deblurring network (iterations through the algorithm) needed for adequate deblurring performance has been determined to be 10 and the number of filters (C) was determined to be 16. We use the same values for the L and C parameters in our patch-based deblurring network.

Figure 2 shows a flow chart of the proposed algorithm for a blurring input image, Y . We note that at each layer, l a set of filters $\{f_i^l\}_{i=1}^C$ is applied to the input image and its subsequent feature maps, Y^l . The “Layer l updates” for $l = 1 \dots L$ are performed using equations (15), (16), (17), (18), and (19). The final image estimate, X is recovered using (21). The inputs to the network are the blurred image Y , an initial estimate of the PSF, K^0 and the filtered sharp image G^0 .

When this patch-based total-variation (TV) deblurring algorithm is ‘unrolled’, each iteration of the algorithm becomes a layer of the unrolled network, i.e. each layer of the network implements (15), (16), (17), (18), and (19) to determine K , an estimate of a deblurring PSF and Z , an estimate of the filtered sharp image. We recover an estimate of the latent sharp image using the estimated PSF, the estimated filtered sharp image, the blurry input image, and f^L (the first filter that was applied to blurry image). The image patches are all deblurred independent-

ly of each other and then easily recombined at each update step. As described in section 3.2.2, to increase the observed receptive field [31] Figure 2 shows that the input blurred image is filtered by L consecutive sets of filters before being passed as input to the first layer.

4 EXPERIMENTS

4.1 Training

To train the network, the algorithm parameters must be initialized. The filter weights, f_i^L were initialized using Xavier initialization [27] and all parameters were updated using the Adam optimizer [26]. Other network specific network parameters (applied in equations (15), (16), (18), and (21)) were initialized as shown in Table 1. Note that the product of variables λ_i^l and ζ_i^i are reparameterized as b_i^l in equation (16) for network training. A non-negativity constraint is applied to each of the parameters in Table 1 after each update step. At the start of each pass through the network, the initial estimate of the filtered image, G^0 is a set of zero valued images for all patches, and the initial PSF estimate, K^0 is set to the delta function for all image patches.

Several variations of the patch based deblurring method were created and trained by varying the patch size and the maximum allowable PSF size. The patch size was varied between 100x100, 200x200, and 400x400 pixels each using a maximum PSF size of 21x21 pixels or 45x45 pixels (yielding six variations in total as shown in Table 2).

Table 1. Parameter Initialization Values.

Parameter	Value
b_i^l	0.02
ζ_i^i	1.0
B^l	0.0
η_i	20

Table 2. Parameter Initialization Values.

		Size of the patches		
		100x100	200x200	400x400
Maximum PSF	21x21	Net-A	Net-C	Net-E
Size	45x45	Net-B	Net-D	Net-F

Each network was trained to deblur the images in a uniformly blurred dataset of 800 images [4], therefore the blur in each patch is identical but the networks learn by deblurring each patches independently. Each network was trained for 100 epochs measuring the sum of the mean squared error (MSE) loss of both the true PSF at each patch and the true patch of the sharp image. The learning rate for each network was set to $1.0e-3$ and was set to decay by 5% every 5 epochs.

4.2 Results of the Network Evaluation

Evaluation of networks was performed by deblurring images GOPRO test dataset [12] and Kohler dataset [25]. The GOPRO test dataset [12] is composed of 1,111 blurry and sharp image pairs featuring simulated motion blur (non-uniform), created by averaging consecutive images of the same scene. The Kohler dataset [25] is composed of blurry images that simulate camera shake using twelve different non-uniform blurs each applied to the same four images to yield 48 blurry images.

We compared the deblurring capabilities of the six patch-based networks (Net-A through Net-F) with the SRN-Deblur network of Tao et al. [13] and the DUBLID algorithm of Li et al. [4]. The PSNR and MSSIM results are obtained by evaluating the deblurred images of the Kohler dataset with the provided script in [25] with results shown in Table 3.

From Table 3, we observe that Net-F (400x400 patch size with a maximum PSF of 45x45) achieved the best overall results with the highest PSNR and SSIM values when deblurring images from the Kohler dataset. Although Net-F has the largest the patch size, this differs from the expectation as the larger window size for the PSF gives less space to accurately estimate that PSF [17]. The results from Net-E were comparable to that of Net-F even though the PSF window for this network (21x21 pixels) is smaller than some of the PSF sizes in the Kohler dataset [25]. The networks with larger patch sizes (Net-E and Net-F with 400x400 pixel patches) did better in general. Since we are able to deblur the images in the Kohler dataset [25] using a large patch size we conclude that the blur in the Kohler dataset does not have high spatial variation.

In contrast from Table 4, we observe that when deblurring images from the GOPRO dataset [12] the networks with the smallest patch size (Net-A and Net-B with 100x100 pixel patches) performed the best overall. We note that Net-A has the highest average SSIM and Net-B has the highest overall average PSNR. The images in the GOPRO dataset are created to simulate motion blur, often

Table 3. Average Kohler Dataset [25] Results.

	SRN-Deblur [13]	DUBLID [4]	Net-A	Net-B	Net-C	Net-D	Net-E	Net-F
PSNR	26.75	29.83	29.73	29.88	29.90	29.96	30.10	30.19
MSSIM	0.837	0.892	0.889	0.893	0.892	0.894	0.895	0.898

Table 4. Average GOPRO Dataset [12] Results.

	SRN-Deblur [13]	DUBLID [4]	Net-A	Net-B	Net-C	Net-D	Net-E	Net-F
PSNR	30.26	29.6	30.19	30.27	29.77	29.65	29.60	26.17
SSIM	0.934	0.930	0.944	0.938	0.939	0.931	0.936	0.830

with individual objects in the scene moving (sometimes in different directions). This means that the blur can have a high spatial variation and explains why the smaller patch

Fig. 3. Evaluation Metrics for GOPRO [12] and Kohler [25] datasets.

sizes of Net-A and Net-B performed better when deblurring images from the GOPRO dataset [12] while Net-F performed significantly worse than the other networks. The performance of the other test networks (Net-C, Net-D, and Net-E) was also comparable to the SRN-Deblur network [13] even though this network was trained to deblur images which are very similar to the test dataset (i.e., a subset of the GOPRO dataset not used in testing the network). Figure 3 shows a graphical representation of the data in Table 3 and Table 4.

Table 5. Number of Parameters.

	SRN-Deblur [13]	DUBLID [4]	Net-A – Net-F
Number of Trainable Parameters	$\sim 3.76 \times 10^6$	$\sim 2.3 \times 10^4$	$\sim 2.3 \times 10^4$

Figure 4 shows a qualitative comparison of the deblurring performance of the Net-A through Net-F as well as SRN-Deblur [13] and DUBLID [4] when given a sample blurry image from the Kohler dataset [25]. The green and blue boxes show closeups of two detailed regions. The qualitative results from Net-A through Net-F and the SRN-Deblur network are comparable and slightly

Fig. 4. Sample Images from the Kohler dataset [26]. Blurry Image and Each network's deblurring result.

sharper than that of the SRN-Deblur network. Figure 5 shows a qualitative comparison of the deblurring performance of the Net-A through Net-F as well as SRN-Deblur [13] and DUBLID [4] when given a blurry sample image from the GOPRO dataset [12]. We observe that the results from Net-A and Net-B are very similar as they did perform best overall. The closeup of the result including text from SRN-Deblur [13] looks sharper than the results from Net-A and Net-B, but the closeup of the building looks sharper in Net-A and Net-B.

In Table 7, we observe that Net-A through Net-F are as compact as is DUBLID [4], i.e., they contain approximately 23,400 parameters, as compared to the 3.7 million parameters in the SRN-Deblur network [13] and the millions of parameters in other deblurring Neural Networks ([12], [13], [14], [15], [16]). This allows for portability and efficiency even on devices with limited computational power. A possible future direction in this work could be to investigate a method to vary the patch size based on the level of spatial variation in the blurred image. This can be implemented by allowing the optimum patch size and position to be determined by blur identification, to improve blind deblurring performance.

5 CONCLUSIONS

In this work we developed a patch-based deblurring method to deblur images blurred in a non-uniform manner (e.g. motion blur or camera shake). This approach modified the total-variation (TV) minimization algorithm

Fig. 5. Sample Images from the GOPRO dataset [12]. Blurry Image and Each network's deblurring result.

unrolled by Li et al [27] to make a network specifically designed to handle non-uniform blur. This approach improved upon the performance of the algorithm unrolling network in cases with non-uniform blur and it performed better than or comparably to the SRN-Deblur network [10], when deblurring the motion blur in the GOPRO dataset [9]. The network remained compact and interpretable as it was created from an iterative MAP optimization algorithm, crafted for deblurring images.

REFERENCES

- [1] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. "Non-uniform Deblurring for Shaken Images," *International Journal of Computer Vision*, Springer Verlag, 2011, 98 (2), pp.168-186. hal-01053877
- [2] Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. "Single image deblurring using motion density functions," In *ECCV*, pages 171–184. Springer, 2010.
- [3] M. Hirsch, C. J. Schuler, S. Harmeling and B. Schölkopf, "Fast removal of non-uniform camera shake," *2011 International Conference on Computer Vision*, 2011, pp. 463-470, doi: 10.1109/ICCV.2011.6126276.
- [4] Y. Li, M. Tofighi, J. Geng, V. Monga and Y. C. Eldar, "Efficient and Interpretable Deep Blind Image Deblurring Via Algorithm Unrolling," in *IEEE Transactions on Computational Imaging*, vol. 6, pp. 666-681, 2020, doi: 10.1109/TCI.2020.2964202
- [5] D. Krishnan, T. Tay and R. Fergus, "Blind deconvolution using a normalized sparsity measure," *CVPR 2011*, 2011, pp. 233-240, doi: 10.1109/CVPR.2011.5995521.
- [6] J. Pan, D. Sun, H. Pfister and M. Yang, "Deblurring Images via Dark Channel Prior," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2315-2328, 1 Oct. 2018, doi:

- 10.1109/TPAMI.2017.2753804.
- [7] F. Wen, R. Ying, Y. Liu, P. Liu and T. -K. Truong, "A Simple Local Minimal Intensity Prior and an Improved Algorithm for Blind Image Deblurring," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 2923-2937, Aug. 2021, doi: 10.1109/TCSVT.2020.3034137.
 - [8] S. Zheng, L. Xu and J. Jia, "Forward Motion Deblurring," 2013 IEEE International Conference on Computer Vision, 2013, pp. 1465-1472, doi: 10.1109/ICCV.2013.185.
 - [9] J. Sun, Wenfei Cao, Zongben Xu and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 769-777, doi: 10.1109/CVPR.2015.7298677.
 - [10] L. Xu and J. Jia, "Depth-aware motion deblurring," 2012 IEEE International Conference on Computational Photography (ICCP), 2012, pp. 1-8, doi: 10.1109/ICCP.2012.6215220.
 - [11] Z. Hu, L. Xu and M. Yang, "Joint Depth Estimation and Camera Shake Removal from Single Blurry Image," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2893-2900, doi: 10.1109/CVPR.2014.370.
 - [12] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conference on Computer Vision Pattern Recognition*, 2017, pp. 257-265.
 - [13] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conference on Computer Vision Pattern Recognition*, 2018, pp. 8174-8182.
 - [14] Y. Yang, J. Sun, H. Li and Z. Xu, "ADMM-CSNet: A Deep Learning Approach for Image Compressive Sensing," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 521-538, 1 March 2020, doi: 10.1109/TPAMI.2018.2883941.
 - [15] T. M. Nimisha, A. K. Singh and A. N. Rajagopalan, "Blur-Invariant Deep Learning for Blind-Deblurring," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4762-4770, doi: 10.1109/ICCV.2017.509.
 - [16] Z. Chen and L. Chang, "Blind Motion Deblurring via Inceptionresnet by Using GAN Model," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, 2019, pp. 1463-1467. doi: 10.1109/ICASSP.2019.8683728
 - [17] A. Levin, Y. Weiss, F. Durand and W. T. Freeman, "Understanding Blind Deconvolution Algorithms," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2354-2367, Dec. 2011, doi: 10.1109/TPAMI.2011.148.
 - [18] T. F. Chan and Chiu-Kwong Wong, "Total variation blind deconvolution," in *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370-375, March 1998, doi: 10.1109/83.661187.
 - [19] Yu-Li You and M. Kaveh, "A regularization approach to joint blur identification and image restoration," in *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 416-428, March 1996, doi: 10.1109/83.491316.
 - [20] D. Perrone and P. Favaro, "A Clearer Picture of Total Variation Blind Deconvolution," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1041-1055, 1 June 2016, doi: 10.1109/TPAMI.2015.2477819.
 - [21] Y. Zhang, W. Ding and C. Liu, "Summary of Convolutional Neural Network Compression Technology," 2019 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 2019, pp. 480-483, doi: 10.1109/ICUS48101.2019.8995969.
 - [22] G. Richmond and A. Cole-Rhodes, "PCA Compression of Image Deblurring Neural Networks," 2021 SPIE Conference on Conference Artificial Intelligence and Machine Learning in Defense Applications III, Madrid, Spain
 - [23] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling. Efficient Filter Flow for Space-Variant Multi-frame Blind Deconvolution. In *Proc. Conf. Comput. Vision and Pattern Recognition*. IEEE 2010.
 - [24] Yeong Ho Ha and J. A. Pearce, "A new window and comparison to standard windows," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 2, pp. 298-301, Feb. 1989, doi: 10.1109/29.21693.
 - [25] R. Kohler, M. Hirsch, B. Mohler, B. Scholkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *Proc. Eur. Conf. Comput. Vision*. 2012, pp. 27-40, Springer.
 - [26] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
 - [27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. ICAIS*, Mar. 2010, pp. 249-256.
 - [28] W.T. Freeman and E. H. Adelson, The design and use of steerable filters" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891-906, Sep. 1991.
 - [29] Q. Barthelemy, A. Larue, A. Mayoue, D. Mercier, and J. I. Mars, "Shiftamp; 2D rotation invariant sparse coding for multivariate signals," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1597-1611, Apr. 2012.
 - [30] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction" *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248-272, Jan. 2008.
 - [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.
 - [32] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Machine Learning*, 2010, pp. 399-406. doi: 10.5555/3104322.3104374.