Multipath Refinement Fusion Network for Efficient Image Deblurring

Zhichao Zhang [1,*], Jinsheng Deng [1,*], Hui Chen[1], Xiaoqing Yin [1,†],

Abstract—**Convolutional neural networks (CNNs) have achieved great success in image process and computer vision. Image deblurring problems addressed by CNNs can achieve better prediction accuracy than those that use traditional methods. CNNs, especially multiscale deep CNNs, nonetheless require substantial memory and computational resources to perform image deblurring, thus hindering their deployment in real-time applications. Computation-efficient networks lack the capability to deal with large-scale datasets and thus cannot generate accurate restoration results in many cases. To this end, we propose an efficient and accurate deep learning framework for image deblurring named MRFNet with three main features. 1) The framework is the first of its kind to utilize the multipath refinement fusion (MRF) network for image deblurring. 2) The MRFNet combines lightweight convolution and residual connection as a means of enhancing model performance. 3) The designed scale refinement loss function accelerates the training loss of the MRF. Comparative experiments on two popular datasets, namely, GOPRO and VisDrone, had been conducted to demonstrate the effectiveness of MRFNet. The experimental results indicate that MRFNet can achieve state-of-the-art prediction accuracy at a faster speed than the other deblurring models. The MRFNet code is available at https://github.com/zhangzhichao19020123/MRFNet.**

I. Introduction

Image processing and computer vision tasks have been widely adopted as deep learning methods. Image deblurring is a classical and important problem in industrial areas, such as aviation photo restoration, robotics recognition, and autonomous driving [1].

Blur kernels are adopted by traditional methods to simplify the problem of deblurring. However, the existing methods are limited because they merely classify blurred images as uniform, non-uniform, depth-aware, and so on. However, blurred images in real-world scenarios consist of mixed types of blurs, such as natural motion blur or camera shake blur, and the use of blur kernels is hardly considered.

Deep learning approaches have been proposed for handling complicated natural blurs. These methods use convolutional layers to extract features by scanning blurred and sharp images, followed by the use of deconvolution layers to fuse and record the learning results. Schuler et al. [11], Zhang et al. [12], Xu et al. [4] have adopted this two-stage traditional procedure by using a simple encoder and decoder neural network. However, these methods still use the traditional framework whose prediction performance remains to be low.

Inspired by the idea described above, Kupyn et al. [5] designed a new framework for deblurring that could calculate the differences of generative and original images.The network has since been called generative adversarial network (GAN). Since GAN's development, many other complicated GAN networks, such as DeblurGAN ver. 2 [6], have been used to deblur the images. However, GAN requires a large amount of computing and memory resources in the process of comparing the generated and real images of the discriminator. Moreover, loss exceeds the speed and memory usage of neural networks when loading the model and extracting the features.

With the advancements in the design of complicated network models, end-to-end deep learning approaches have also been proposed for deblurring. Such complicated networks can be classified into four classes: multiscale network, recurrent network, multipatch network, and scale-iterative network.

The frameworks of Nah et al. [7] and Lin et al. [8] entailed a multiscale style. The main idea of their framework is to implement the coarse-to-fine strategy to deblur the images in consecutive stages. The coarse stage obtains features by using scales, then the features are halved in a series of steps. The fine stage learns the larger-scale features with the aid of the coarse features until the original size is reached. The coarse-to-fine mechanism needs to be performed directly via the scale-cascaded structure. Thus, despite the fine results, the network's size and depth eventually become excessive, leading to high GPU memory consumption.

Multipatch networks had been proposed by Nekrasov [9] and Zhang et al. [10]. Both of them applied the recurrent method to reuse the last-turn results in the next round as a means of refining the final checkpoints. Images are separated into patches and extracted features, and the meaningful results are sent to the next iteration for further enhancement. This method can help to reduce the parameters by learning from patches in one round. However, the approach is hindered by high calculation cost and low efficiency.

Here, we combine the previous ideas into a single network and propose the MRFNet. In particular, we propose a network that can learn features by using the multipath. Each path fuses the results from both the lower-scale refinement patches and the last-step results to yield higher-latent prediction results. Each path is immediately calculated using the L2 weight loss function. We overcome the multiscale parameter problem and the recurrent architecture's low-efficiency issue by designing the core module of multipath refinement fusion (MRF) networks. The contributions of this research are as follows:

- We propose a novel MRF network architecture for image deblurring. Compared with the previous multiscale and recurrent architectures, our model is more efficient and performs well in terms of image quality and inference speed. Experiments have been conducted to prove the significant impact of the

* Authors have contributed equally.
† Corresponding authors.
1 Department of Computer Science, National University of Defense Technology, yinxiaoqing89@163.com

Manuscript 549 submitted to 2021 IEEE International Conference on Robotics and Automation (ICRA). Received October 29, 2020.

- lightweight process and the residual connection on the enhanced accuracy and decreased complexity of the proposed network.
- An MRF unit that can fuse the results to the features extracted from the multilevel path is designed. The lightweight process and the residual connection have been reconstructed accordingly in the network architecture. The modules not only can solve the multilevel requirement of concatenating different kinds of feature maps but can also help to train a deeper and fast network.
- State-of-the-art deblurring performance can be achieved according to the quantitative numerical analyses of PSNR and SSIM. A scale refinement loss function has been developed for VisDrone and GOPRO datasets, in which motion blur and Gaussian blur are both included.

The rest of the paper is organized as follows. We introduce the related work on image deblurring in network architectures in Section II. Section III introduces the methodology and the implementation of our proposed network. We discuss our experimental results in Section IV. We conclude the paper in Section V.

II. Related work

Traditional methods [1],[11]–[15] rely on blur kernel estimation to reconstruct images by focusing on specific types of blurs. Recent studies have attempted to settle the restoration problem by adopting multiscale convolutional neural networks to deblur the images. In these end-to-end frameworks, blurred images are used as inputs for the neural network to immediately generate clear images [16]–[10]. Compared with the traditional methods, CNNs can greatly improve training speed, but its prediction accuracy is inefficient, and considerable GPU memory is occupied.

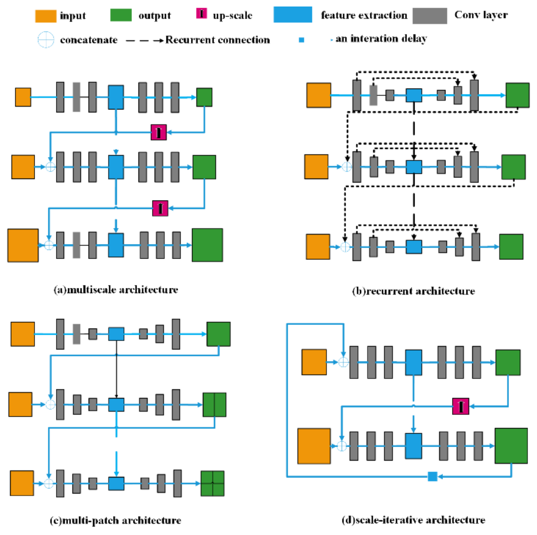

input
output
up-scale
feature extraction
Conv layer
concatenate
Recurrent connection
an interation delay
(a)multiscale architecture
(b)recurrent architecture
(c)multi-patch architecture
(d)scale-iterative architecture

Fig. 1: Various deblurring network architectures. (a) Nah et al. [7] proposed the multiscale architecture to extract features from different scales. (b) Tao et al. [20] proposed the recurrent architecture, in which the next round of training can be aided by the last-round results. (c) Zhang et al. [10] utilized the multipatch architecture to directly extract features from image pairs by cropping images in different scales. (d) Ye et al. [21] used the scale-iterative architecture to train the model with an upsampling path with aid of the last-iterative middle results. We combine the ideas of (a) and (b) and propose a new framework whose core module involves the MRF and call it MRFNet. The MRFNet can operate in both multiscale and recurrent manner.

As for the feature extraction, Image deblurring CNNs can be divided into GAN network, multiscale network, recurrent network, multipatch network, and scale-iterative architecture.

GAN [5],[22] is a conditional adversarial network that uses a multicomponent loss function. It is a method involving blind deblurring without the use of a blur kernel. During the training process, a discrimination network is introduced for the comparison between blur and original images. GAN retains most of the texture details in the images, and it can be applied to different sizes of images. However, this method takes a long time in generating images and computing the image differences.

By scaling an image into different sizes, multiscale networks [7],[8],[23] are able to extract various features from each scale, as shown in Fig. 1(a). The input images are converted into feature maps, then scales are used to halve the feature maps at the next level. In multiscale detection, the various scale features are fused by different methods. The various scale features contain a large quantity of information, suggesting high accuracy. However, the multiscale strategy strictly requires the features to be extracted from the small scale to the large scale, which means that large-scale concatenating needs to wait for the computing results from the small scales, resulting in slow training speed.

An input layer, a loop hiding layer, and an output layer constitute a recurrent network [20],[24],[25], as shown in Fig. 1(b). Recurrent networks can learn features and long-term dependencies in sequence. However, as the number of network layer increases, so does complexity. The increase in the number of layers or circular connections deepens the network, thus enabling the network to provide multilevel feature extraction. As the concatenating of recurrent networks relies heavily on last-round results, the process worsens if invalid features are extracted in these last-round results, then the deblurring inference becomes extremely unstable if some image restorations have poor quality.

DMPHN [10] is a CNN model that appears to be simple but operates as an effective multipatch network [10],[26],[27], as shown in Fig. 1(c). An input image is divided into different sizes each time, then the features are extracted by the multiscale architecture. Although DMPHN has attained remarkable progress in terms of computational effectiveness, its precision is low.

Ye et al. [21] proposed a scale-iterative network [21],[28] to restore sharp images in iteratively, as shown in Fig. 1(d). The super-resolution structure of the upsampling layer is adopted between two consecutive scales to restore the details. Image features are extracted from the small scale to the large scale, with the aim of reconstructing high-resolution images from low-resolution images. Then, the downsampling part starts to restore the image until its size equals that of the original image. Along with the configurable settings, the scale-iterative network has the advantages of being able to use different scales; moreover, its weight sharing can be preserved, and its training process is flexible. However, the method fails to achieve high deblurring precision and network efficiency, and hefty memory is needed for the iterative calculation.

Manuscript 549 submitted to 2021 IEEE International Conference on Robotics and Automation (ICRA). Received October 29, 2020.

III. Model Design and Implementation

An MRFNet has been extensively constructed to ensure the balance between accuracy and speed, as these aspects are currently lacking in the existing studies. We first exploit the recurrent and multiscale strategies to train the network. Then, a structure with a branch depth and fusion unit is built on the basis of the lightweight process, the MRF unit, and the remote residual connection. Finally, a scale refinement loss function is used to train the network in a coarse-to-fine manner.

A. *Multiscale and Recurrent Learning*

Instead of stacking the encoder and decoder processes to directly perform deblurring refinement, the recurrent and multiscale learning strategies were applied in this study. The basic idea of the multiscale learning strategy is for the top network to extract features from the large-coarse scale maps and the bottom network upsampling results. Meanwhile, in the recurrent learning strategy, the bottom layer acquires fusion information from the small refinement maps and the top feedback. In our work, the two strategies were combined by designing four refinement paths to extract features in different scales instead of directly predicting the whole deblurred image. In our method, the top network only needs to focus on learning highly nonlinear residual features, which is effective in restoring deblurred images in a coarse-to-fine manner. The architecture of the proposed MRFNet is shown in Fig. 2.

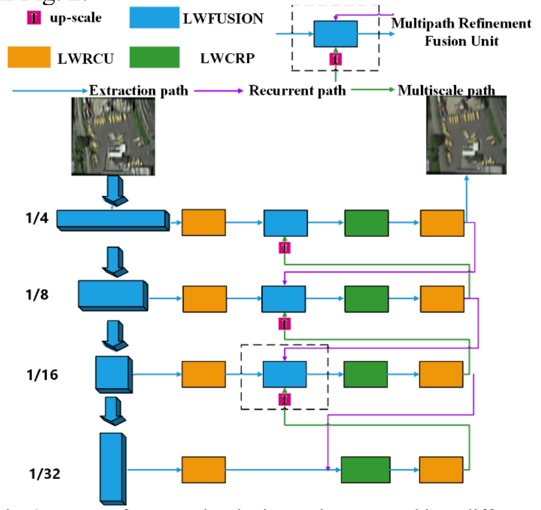


Fig. 2: MRF framework. The image is separated into different scales from top to bottom. Blue line refers the extraction path of extracting features from scales. Then, the blocks of MRF fuse the recurrent last-round results (purple line) and the upsampling feature maps (green line) as a single refinement process. The total four refinement paths finally compute the loss in the scale refinement loss function, then the best deblur results are obtained.

In the multipath input stream illustrated in Fig. 3(d), the upper MRFNet layer takes the left and right images as the input and processes the deblur datasets into a total of four scales, i.e., k is from 2 to 4.

The four-scale blur feature maps are denoted as b_k, while the refinement results are denoted as l_k. First, the k level of the multipath input stream concatenates the same scale feature maps b_k and the upsampling feature maps l_{k+1} as middle feature maps, which is denoted as c_k.

$$c_k = b_k \oplus l_{k+1} \quad (2 \leq k \leq 4) \qquad (1)$$

Then, the fusion unit adds both c_k and the last-round results l_{k-1} as the final result, which is denoted as l_k. This process briefly describes how the refinement fusion path works. The whole process can be calculated as

$$l_k = c_k + l_{k-1} \quad (2 \leq k \leq 4) \qquad (2)$$

B. *Lightweight Process and MRF*

A large number of parameters and floating-point operations of the original MRF network originates from the commonly used 3 × 3 convolution. Therefore, we focus on replacing these elements with simpler counterparts without compromising performance.

The original design of an MRF network is to use an encoder–decoder structure equipped with four feature extraction and downsampling layers.

Each path has a fusion unit. The basic block uses 3 × 3 convolution, which we call the fusion unit. Here, the 1 × 1 fusion unit in Fig. 3(a) is replaced with 3 × 3 convolution.

A chained residual pool (CRP) is also considered [35] to naturally illustrate why the lightweight process works and how the former three units are reshaped. The lightweight process is applied to the CRP unit by substituting the 5 × 5 and 3 × 3 convolution with the 5 × 5 and 1 × 1 convolution in Fig. 3(b).

The refinement path adopts a convolutional layer with a stride of 1 followed by a convolution layer with a stride of 2 such that they consistently shrink the feature map size by half [29]. The two convolution layers act as a residual connection unit (RCU) [35]. Two RCUs are installed in the encoder, while three RCUs are installed in the decoder. All of the blocks use 1 × 1, 3 × 3, and 1 × 1 convolutions compared with those in the RCU that use 3 × 3 and 3 × 3 convolution. We call the two convolution layers as the lightweight residual connection unit (LWRCU), as illustrated in Fig. 3(c).

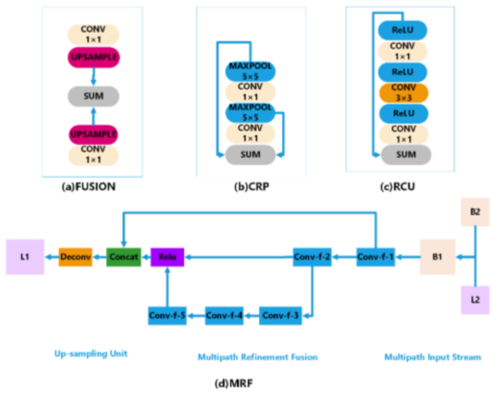


Fig. 3: Different parts of the network. (a) Fusion unit, (b) improved CRP module, (c) lightweight network structure of RCU, and (d) MRF unit.

TABLE I: Specific parameters of the MRFNet.

Network	Kernel	Stride	Padding	Network	Kernel	Stride	Padding
Conv1	5×5×32	1	2	conv_r2_m2	1×1×128	1	1
Conv2	1×1×64	1	1	conv_r2_m3	3×3×128	1	1
Conv3	5×5×128	2	2	conv_r2_m4	1×1×128	1	1
Conv4	1×1×128	1	1	deconv2	4×4×64	1	2
Conv5	3×3×256	1	2	conv_r3_1	3×3×64	1	1
Conv6	1×1×256	1	1	conv_r3_m1	3×3×64	1	1
Conv7	3×3×256	1	2	conv_r3_m2	1×1×64	1	1
Conv8	1×1×256	1	1	conv_r3_m3	3×3×64	1	1
conv_r1_1	3×3×256	1	1	conv_r3_m4	1×1×64	1	1
conv_r1_m1	3×3×256	1	1	deconv3	4×4×32	1	2
conv_r1_m2	3×1×256	1	1	conv_r4_1	3×3×32	1	1
conv_r1_m3	3×3×256	1	1	conv_r4_m1	3×3×32	1	1
conv_r1_m4	3×1×256	1	1	conv_r4_m2	3×3×32	1	1
deconv1	4×4×128	1	2	conv_r4_m3	1×1×32	1	1
conv_r2_1	3×3×128	1	1	conv_r4_m4	3×3×32	1	1

Manuscript 549 submitted to 2021 IEEE International Conference on Robotics and Automation (ICRA). Received October 29, 2020.

Intuitively, a convolution with a relatively large core size is designed to increase the size of the receiving field (as well as the global context coverage). The 1 × 1 convolution can only locally transform the features of each pixel from one space to another. Here, we empirically prove that the replacement with 1 × 1 convolution would not weaken network performance. Particularly, we replace the 3 × 3 convolution in the CRP and the fusion block with a 1 × 1 counterpart, and we modify the RCU to LWRCU with a bottleneck design, as shown in Fig. 3(c). In using this method, we can reduce the number of parameters by more than 50% and the number of triggers by more than 75% (Table I). The convolutions have been shown to save considerable computation without sacrificing performance.

We also enhance the MRF unit illustrated in Fig. 3(d). Deep residual networks obtain rich feature information from multi-size inputs [30]. The residual block originally derived in for the image classification tasks is widely used to learn robust features and train deeper networks. Residual blocks can well address vanishing gradient problems. Thus, we replace the connection layer with the MRF unit.

Here, the MRF is specifically designed as a combination of multiple convolution layers (conv-f-1 to conv-f-5), and each convolution layer is followed by a ReLU activation function. Conv-f-2 uses feature maps generated by Conv-f-1 to generate more complex feature maps. Similarly, conv-f-4 and conv-f-5 continue to use the feature map generated by conv-f-3 for further processing. Finally, the feature maps obtained from multiple paths are fused together. The specific calculation expression is as follows:

$$y = f_2\big(f_1(x)\big) + f_4\Big(f_3\Big(f_2\big(f_1(x)\big)\Big)\Big) \qquad (3)$$

where *f*, *x* and *y* represent the convolution operation, the characteristic graph of the input, and the characteristic graph of the output, respectively.

We construct a residual connection in each path of the MRFNet. In the process of forward transmission, the remote residual connections transmit low-level features, which are used to refine the visual details of the coarse high-level feature maps. The residual connections allow the gradients to propagate directly to the early convolution layers, thus contributing to effective end-to-end training.

We set the number of paths from 1 to 6 for the multipath process. The operation takes up the least parameters when the paths are equal to 3, whereas the best performance is achieved when the paths are equal to 4. When the number of paths is less than 3, the extracted features are not accurate, and the training loss remains at a high level all of the time. When the number of paths exceeds 4, the deblurring encounters severe performance degradation. To this end, we choose the four-path refinement setting as the final backbone.

C. Loss Function Design

Given a pair of sharp and blurred images, MRFNet takes these images as the input and produces four groups of feature maps in different scales. Assume that the input image size is H × W. The four scales of the feature maps are H/4 × W/4, H/8 × W/8, H/16 × W/16, and H/32 × W/32. In training the MRFNet in an end-to-end manner, we adopt the L2 loss between the predicted deblurring result map and the ground truth as follows:

$$L(\theta) = \frac{1}{2N}\sum_{i=1}^{N} ||x_s^i - F(x_l^i)||^2 \qquad (4)$$

where θ is the parameter set, x_s is the ground truth patch, and *F* is the mapping function that generates the restored image from the *N*-interpolated LR training patches x_l. Here, the patch size is defined at different levels.

The scale refinement loss function is useful in learning the features in a coarse-to-fine manner. Each refinement path has a loss function that can be used to evaluate the training process. When others adopt a single final L2 loss function, our scale refinement loss function computes the results in different scales, which leads to a much faster convergence speed and an even higher inference precision.

$$L_{final} = \frac{1}{2K}\sum_{k=1}^{K} \frac{1}{c_k w_k h_k} ||L_k - S_k||^2 \qquad (5)$$

where L_k represents the model output of the scale level *k*, and S_k is the k-scale sharp maps.

The loss at each scale is normalized by the number of channels C_s, width W_s, and height H_s.

IV. Performance Evaluation

In this section, we compare MRFNet to the recently adopted methods of DeblurGAN, DMPHN, and SIUN in terms of accuracy and time efficiency.

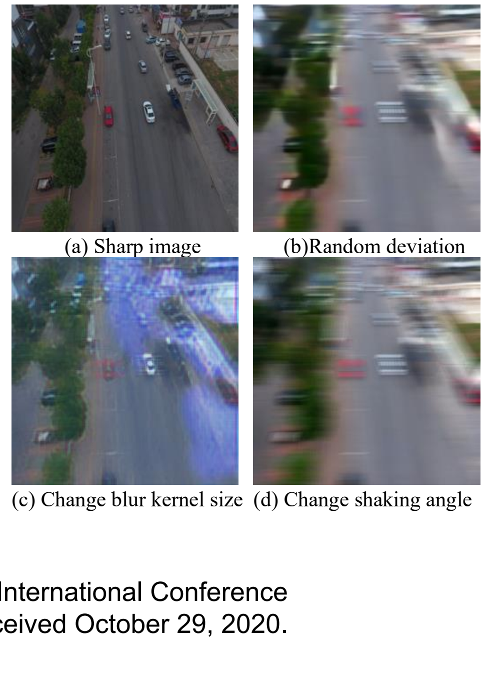
A. Experimental Setup

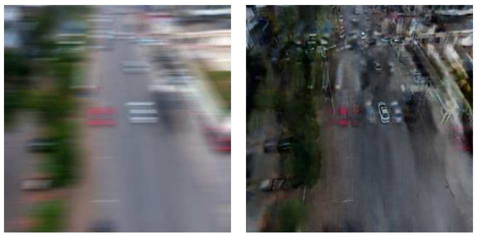
We implement our MRFNet by using Caffe. The model is trained with Adam (β_1 = 0.9, β_2 = 0.999). In the training process, the images are randomly cropped to 256 × 256. The batch size of 16 is used for the training in four NVIDIA RTX2080Ti GPUs. At the beginning of each epoch, the learning rate is initialized as 10^{-4} and subsequently decayed by half every 10 epochs. We train 70 epochs for VisDrone and 50 epochs for GOPRO.

For the time efficiency aspect, we evaluate the inference time of the existing state-of-the-art CNNs on RTX2080Ti GPUs with 11 GB of memory.

B. Dataset

We use two public datasets to train and evaluate the performance of MRFNet. The first dataset is VisDrone, which provides synthetic blur techniques. The second dataset is GOPRO, which is captured in real-world scenarios.


(a) Sharp image (b)Random deviation
(c) Change blur kernel size (d) Change shaking angle



(e) Change shaking length (f)Motion blur

Fig. 4: Overview of dataset augmentation by changing the parameters. The first column shows a sharp image, an image blurred by changing the blur kernel size, and an image blurred by altering the blur shaking length. The second column shows the blurred image added with random standard deviation, in which the image is blurred by changing the shaking angle. The last image is generated by real motion blur.

The benchmark dataset of VisDrone consists of 288 video clips formed by 261,908 frames and 10,209 static images. The dataset, which was captured by various drone-mounted cameras, covers a wide range of elements, including location (taken from 14 different cities separated by thousands of kilometers in China), environment (urban and country), objects (pedestrian, vehicles, or bicycles.), and density (sparse and crowded scenes). The dataset was collected using various drone platforms (i.e., drones of different models) in different scenarios under various weather and lighting conditions.

We produce the VisDrone blur dataset by using different methods, including Gaussian blur. Several data enhancement techniques can be utilized to prevent our network from overfitting. In terms of geometric transformation, the images are flipped horizontally, vertically, and rotated at random angles. For the colors, the RGB channels are replaced randomly. For the image color degradation, the saturation in the HSV color space is multiplied by a random number in [0,5]. In addition, Gaussian blur is added to the blurred image. Finally, to ensure that our network is robust to noise of different intensities, the standard deviation of noise is also randomly sampled from the Gaussian distribution N (0-1).

The GOPRO dataset contains images with changeable motion blurs. GOPRO cameras can record sets of sharp information that are integrated over time to generate blurred images rather than using a preset kernel to blur the generated images. When the camera sensor of GOPRO receives light during exposure, it accumulates a clear image stimulus each time, resulting in a blurred image [31].

C. *Comparative Experiments*

We conduct comparative experiments with DeblurGAN, DMPHN, and SIUN to verify the performance of our model. Our proposed MRFNet can achieve state-of-the-art performance compared with SIUN on VisDrone. The values of PSNR and SSIM are much higher than DeblurGAN and DMPHN, suggesting the advantage of our method in handling Gaussian blurs. Moreover, our method performs better than SIUN and DMPHN and even much better than DeblurGAN in dealing with the motion blurs of GOPRO. The trends in Table II prove the superiority of the MRFNet framework based on the PSNR and SSIM values.

TABLE II. Testing results of the blurred image datasets and their PSNR and SSIM values.

Method	**GOPRO**		**VisDrone**	
	PSNR	***SSIM***	***PSNR***	***SSIM***
deblur GAN	28.22642	0.747912	28.29447	0.609642
DMPHN	34.21846	0.898285	28.54136	0.526301
SIUN	34.46135	0.900913	28.28039	0.543417
Our model	**34.63429**	**0.907881**	**29.40845**	**0.862474**

D. *Ablation Experiments*

The original MRF network used as the benchmark is denoted as RefineNet. From this original RefineNet, we add the lightweight process to the benchmark and denote it as LWRefineNet. Then, we add the remote residual connection to the refinement path and residual connection to fusion unit and denote it as RCRefineNet. Finally, we combine lightweight and residual strategy to the benchmark network and define it as MRFNet.

TABLE III. Memory consumption of graphics cards by different methods

Method	**GOPRO**	**VisDrone**
	Network(MB)+Batch(8)	***Network(MB)+Batch(16)***
deblur GAN	**4538**	6012
DMPHN	6541	7329
SIUN	8399	8561
Our model	5452	**5898**

As shown in Table III, DeblurGAN requires the least GPU memory usage at 4538 MB, while our method requires slightly higher GPU memory usage than DeblurGAN in GOPRO. For the VisDrone dataset, our network consumes the least GPU memory for the batch size of 16. The lightweight process can reduce the parameters of the model, thus contributing to the low-memory requirement.

TABLE IV: Average time of inferring images.

Method	**GOPRO**	**VisDrone**
	InferTime(s)	***InferTime(s)***
Deblur GAN	2.346	2.144
DMPHN	1.886	0.764
SIUN	0.684	0.357
LWRefineNet	**0.494**	**0.319**

As shown in Table IV, LWRefineNet is the fastest method in terms of the time of loading the network model and the inferences. The inference is run on GTX1650 4G GPU. The image size from GOPRO is 1280 × 768, while that from VisDrone is 256 × 256.

TABLE V: Quantitative numerical results on PSNR and SSIM.

Method	**GOPRO**		**VisDrone**	
	PSNR	***SSIM***	***PSNR***	***SSIM***
RefineNet	34.17826	0.894369	28.73991	0.854758
LWRefineNet	34.21445	0.906998	29.24461	0.860164
RCRefineNet	34.39430	0.903012	29.03971	0.858601
MRFNet	**34.63429**	**0.907881**	**29.40845**	**0.862474**

As shown in Table V, the LWRefineNet and RCRefineNet perform slightly better than RefineNet. MRFNet has the most significant numerical results.

As shown in Fig. 5, our scale refinement loss function takes each sub-task as an independent component within a single unified task, allowing the training process to converge more rapidly and perform better than those of the other methods. The training losses of the other approaches decrease remarkably at the first round and consistently stay at 6% with a smooth trend in the following training courses. Our method, aided by the loss weight scheduling technique, exhibits a dramatic downward trend and remains at approximately 4%. The model accuracy improvements (approximately 10% to 21%) resulting from the multiple rounds of training for the four loss weight groups verify the good convergence and advantages of our method's training strategy.

The experimental results indicate that MRFNet can achieve considerable precision. Furthermore, MRFNet runs much faster than the other deblurring models, such as SIUN and DMPHN. Compared with DeblurGAN, the proposed MRFNet model performs well both in terms of speed and deblurring quality of images. Owing to the added lightweight process, the GPU memory's occupation remains at a low level.

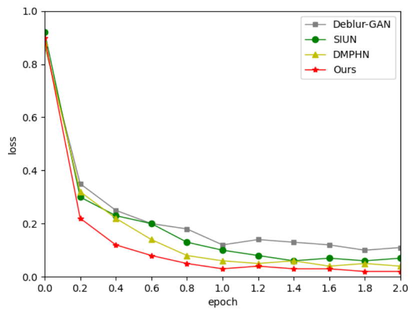
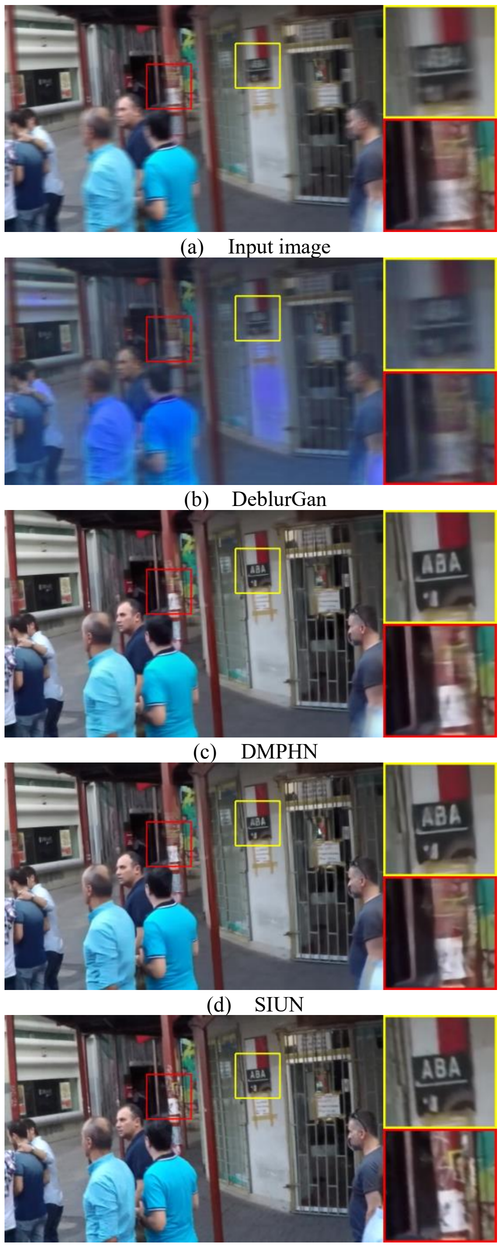


Fig. 5: Training loss of the four methods. Only the first two epochs are shown.



(e) Our method

Fig. 6: Visual effects of different methods. From top to bottom: blurred image, results of DeblurGAN, DMPHN, SIUN and ours. The left images are global deblur results, while local restoration details are shown on the right. Our results show clear object boundaries without artifacts.

Our method can also recover more details and achieve relatively high SSIM and PSNR values. Figs. 6 and 7 show that results of the other models whose images remain unstable and sometimes contain artifacts and color distortions, whereas the MRFNet performs image deblurring in a stable and sharp manner. For instance, the handwriting and flags in Fig.6 and the details in the streets in Fig. 7 are processed to perfect state by MRFNet.

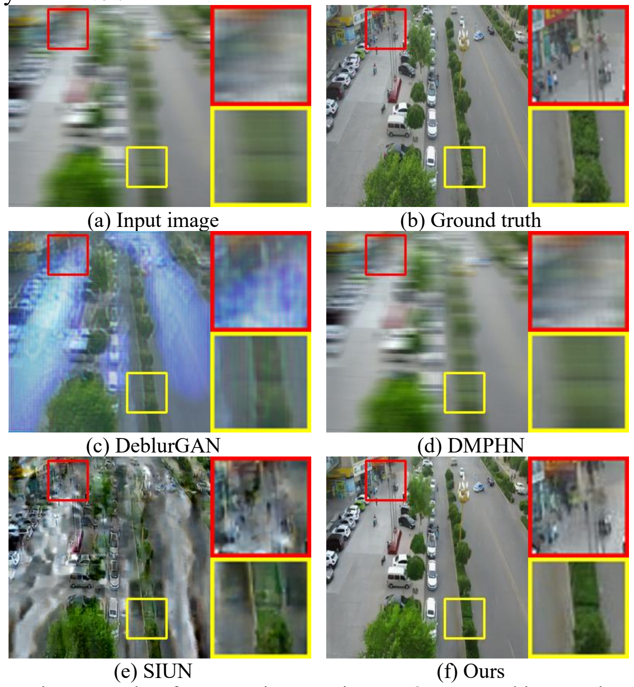


Fig. 7: Results of comparative experiments. Our restored images show vivid colors and sharp details.

V. Conclusion and Future Work

In this study, we propose an efficient yet accurate framework called the MRFNet. The proposed network exploits the MRF, lightweight process, remote residual connection, and scale refinement loss function to enable the model to handle motion and Gaussian blur scenarios while preserving its fast inference speed. We have compared MRFNet with existing state-of-the-art models on two popular datasets.

In the future, we will develop fast deblurring inference of MRFNet on edge devices. As computational capability will likely be much lower than that of GPUs used in our experiments, the techniques of model compression, including pruning, quantization, and so on, will also be explored. We will also adapt this model to video deblurring or the deblurring of inpainting results at the post-processing stage.

Acknowledgements

We thank the anonymous reviewers for their constructive comments and suggestions. We would also like to thank the National University of Defense Technology (NUDT) Technology Center for providing the GPU clusters. This research is supported by a technical center think-tank of NUDT. This work is also supported by the National Natural Science Foundation of China (Grant No. 62001493).

References

[1] S. Cho and S. Lee, ‘‘Fast motion deblurring,’’ ACM Trans. Graph., vol. 28,no. 5, p. 145, 2009.

[2] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf,‘‘A machine learning approach for non-blind image deconvolution,’’in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013,pp. 1067–1074..

[3] J. Zhang, J. Pan, W.-S. Lai, R. W. Lau, and M.-H. Yang, ‘‘Learning fully convolutional networks for iterative non-blind deconvolution,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 3817–3825..

[4] L. Xu, J. S. Ren, C. Liu, and J. Jia, ‘‘Deep convolutional neural network for image deconvolution,’’ in Proc. Adv. Neural Inf. Process. Syst., 2014,pp. 1790–1798..

[5] Kupyn O, Budzan V, Mykhailych M, et al. DeblurGAN: Blind motion deblurring using conditional adversarial networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Reconitinn. 2018: 8183-8192

[6] Kupyn O, Tetiana Martyniuk. DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better

[7] S. Nah, T. Hyun Kim, and K. Mu Lee, ‘‘Deep multi-scale convolutional neural network for dynamic scene deblurring,’’ in Proc. IEEE Conf. Com-put. Vis. Pattern Recognit., Jul. 2017, pp. 3883–3891..

[8] Lin, G., Milan, A., Shen, C., & Reid, I. (n.d.). High-Resolution Semantic Segmentation.2017Ma, K., Feng, H., Luo, J., & Bo, Q. (2019). Re fi neNet4Dehaze : Single Image Dehazing Network Based on RefineNet. 498–507..

[9] Nekrasov, V., & Reid, I. (2018). Light-Weight RefineNet for Real-Time Semantic Segmentation. 1–15.

[10] H. Zhang, Y . Dai, H. Li, and P . Koniusz, ‘‘Deep stacked hierarchical multi-patch network for image deblurring,’’ in Proc. IEEE Conf. Comput. Vis.Pattern Recognit., Jun. 2019, pp. 5978–5986..

[11] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, ‘‘Non-uniform Deblurring for Shaken Images,’’ Int. J. Comput. Vis., vol. 98, no. 2, pp. 168–186,Jun. 2012.

[12] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, “Deblurring text images via L0-regularized intensity and gradient prior,” in Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2901-2908, June,2014.

[13] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, ‘‘Fast removal of non-uniform camera shake,’’ in Proc. Int. Conf. Comput. Vis., Nov. 2011,pp. 463–470

[14] O. Whyte, J. Sivic, and A. Zisserman, ‘‘Deblurring shaken and partially saturated images,’’ Int. J. Comput. Vis., vol. 110, no. 2, pp. 185–201,Nov. 2014.

[15] L. Xu, S. Zheng, and J. Jia, ‘‘Unnatural L0sparse representation for natural image deblurring,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,Jun. 2013, pp. 1107–1114

[16] D. Krishnan, T. Tay, and R. Fergus, ‘‘Blind deconvolution using a normalized sparsity measure,’’ in Proc. CVPR, Jun. 2011, pp. 233–240.

[17] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, ‘‘Deblurring text images via L0-regularized intensity and gradient prior,’’ in Proc. IEEE Conf. Comput. Vis.Pattern Recognit., Jun. 2014, pp. 2901–2908.

[18] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, ‘‘Deblurring face images with exemplars,’’ inProc. Eur . Conf. Comput. Vis.Cham, Switzerland: Springer,2014, pp. 47–62

[19] S. Sreehari, S. V . V enkatakrishnan, B. Wohlberg, G. T. Buzzard,L. F. Drummy, J. P . Simmons, and C. A. Bouman, ‘‘Plug-and-play priors for bright field electron tomography and sparse interpolation, ’’IEEE Trans.Comput. Imaging, vol. 2, no. 4, pp. 408–423, 2016..

[20] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, ‘‘Scale-recurrent network for deep image deblurring,’’ in Proc. IEEE Conf. Comput. Vis. PatternRecognit., Jun. 2018, pp. 8174–8182.

[21] Ye, M., Lyu, D., & Chen, G. (2020). Scale-Iterative Upscaling Network for Image Deblurring. IEEE Access, 8, 18316–18325.

[22] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, ‘‘Learning to Deblur,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 7,pp. 1439–1451, Jul. 2016.

[23] Dai, J., & Wang, Y. (2019). Multiscale Residual Convolution Neural Network and Sector Descriptor-Based Road Detection Method. IEEE Access, 7, 173377–173392

[24] K. Schelten, S. Nowozin, J. Jancsary, C. Rother, and S. Roth, ‘‘Interleave dregression tree field cascades for blind image deconvolution,’’ in Proc.IEEE Winter Conf. Appl. Comput. Vis., Jan. 2015, pp. 494–501

[25] K. Zhang, W. Zuo, S. Gu, and L. Zhang, ‘‘Learning deep cnn denoiser prior for image restoration,’’ inProc. IEEE Conf. Comput. Vis. Pattern Recognit.,Jul. 2017, pp. 3929–3938..

[26] Nekrasov, V., & Reid, I. (2018). Light-Weight RefineNet for Real-Time Semantic Segmentation. 1–15

[27] Ma, K., Feng, H., Luo, J., & Bo, Q. (2019). RefineNet4Dehaze : Single Image Dehazing Network Based on RefineNet. 498–507..

[28] Wang, Q., Shi, S., Zheng, S., Zhao, K., & Chu, X. (2020). FADNet : A Fast and Accurate Network for Disparity Estimation.

[29] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. CoRR, abs/1412.6856, 2014.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016

[31] Y.-W. Tai, X. Chen, S. Kim, S. J. Kim, F. Li, J. Yang, J. Yu,Y. Matsushita, and M. S. Brown. Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. PAMI, 35(10):2498–2512, 2013. 3

[32] J. Sun, W. Cao, Z. Xu, and J. Ponce, ‘‘Learning a convolutional neural network for non-uniform motion blur removal,’’ in Proc. IEEE Conf.Comput. Vis. Pattern Recognit., Jun. 2015, pp. 769–777

[33] Sun j, Cao W, Xu Z, et al. Learning a convolutional neural network for non-uniform motion blur removal[C]//Proceedings of the IEEE Conference on Computer Version and Pattern Recognition.2015:769-777

[34] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.

[35] Shi, J. (2019). MR Image Super-Resolution via Wide Residual Networks With Fixed Skip Connection. 23(3), 1129–1140

[36] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf.Fast removal of non-uniform camera shake. In ICCV, 2011.1, 3.