BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 70(3), 2022, Article number: e140466
DOI: 10.24425/bpasts.2022.140466

MFFNet: a multi-frequency feature extraction and fusion network for visual processing

Jinsheng DENG[1], Zhichao ZHANG[2]*, and Xiaoqing YIN[1]

[1] College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410000, China
[2] College of Computer, National University of Defense Technology, Changsha 410000, China

Abstract. Convolutional neural networks have achieved tremendous success in the areas of image processing and computer vision. However, they experience problems with low-frequency information such as semantic and category content and background color, and high-frequency information such as edge and structure. We propose an efficient and accurate deep learning framework called the multi-frequency feature extraction and fusion network (MFFNet) to perform image processing tasks such as deblurring. MFFNet is aided by edge and attention modules to restore high-frequency information and overcomes the multiscale parameter problem and the low-efficiency issue of recurrent architectures. It handles information from multiple paths and extracts features such as edges, colors, positions, and differences. Then, edge detectors and attention modules are aggregated into units to refine and learn knowledge, and efficient multi-learning features are fused into a final perception result. Experimental results indicate that the proposed framework achieves state-of-the-art deblurring performance on benchmark datasets.

Key words: deblurring; multi-feature fusion; deep learning; attention mechanism.

1. INTRODUCTION

Image attribute restoration is a traditional computer vision challenge that can be separated into many branches: deblurring, denoising, deraining, coloring, and inpainting. These challenges have similar attributes in causing image deficiencies; thus, we can address them in using deep learning methods, which are good at processing large datasets. We aim to find a method to tackle the challenges in image vision processing in an efficient and comprehensive manner.

Designing architectures and algorithms that enhance computer vision performance or imitate human perception is a huge challenge in artificial intelligence. Convolutional neural networks (CNNs) have exhibited promising performance in imitating brain neural cells to capture meaningful features and learn knowledge from specific visual systems. However, they employ a brute-force method to search images and locate latent meaningful information, and the transformation of the information obtained leads to key elements being omitted. We propose a framework with brain-like perception. The proposed network can handle information from multiple paths and extract features such as edges, colors, positions, and differences. The edge detectors and attention modules are aggregated into brains to refine and learn knowledge, and efficient multi-learning of features is carried out and the results are fused to obtain a final perceptive result. In this manner, the proposed network, called the multi-frequency feature fusion network (MFFNet), acts similarly to the human brain to extract the local texture effectively and process universal computer vision tasks such as inpainting, denoising, and super resolution. Image deblurring is a classical and important problem in industrial areas, such as aviation photo restoration, robotics recognition, and autonomous driving [1].

To simplify deblurring, traditional methods adopt fixed blur kernels. However, these methods are limited because they merely classify blurriness into uniform, nonuniform, and depth-aware categories. Moreover, while blurry images in real-world scenarios consist of mixed types of blur such as natural motion blur and camera shake blur, they can only identify a single type of blur at one time.

Deep learning approaches have been proposed for handling complicated natural blurs. These methods [2, 3] use convolutional layers to extract features by scanning blurred and sharp images, fusing features by deconvolution layers and recording the learning results. Schuler *et al.* [2], Zhang *et al.* [4], and Xu *et al.* [5] adopted this two-stage traditional procedure using an encoder–decoder neural network. However, these methods still adopt the traditional framework, resulting in low prediction performance.

Inspired by the problems described above, Kupyn *et al.* [6] designed a new framework for deblurring that can calculate the differences of generative and original images. Generative adversarial networks (GAN) have shown promising performance in image deblurring. Scholars have also made significant improvements to GANs, with improved variants such as DeblurGANv2 [7]. However, GANs are resource-intensive when comparing the generated and real images of the discriminator. With advancements in the design of sophisticated network models, more complicated end-to-end deep learning approaches have also been proposed for deblurring. Such networks can be classified into four network classes: multiscale, recurrent, multi-patch, and scale-iterative networks.

*e-mail: zhangzhichao11@nudt.edu.cn

Manuscript submitted 2021-07-15, revised 2021-10-25, initially accepted for publication 2021-11-25, published in June 2022.

Nah *et al.* [8] and Lin *et al.* [9] developed multiscale frameworks in which the underlying idea is to implement the coarse-to-fine strategy to deblur the images in consecutive stages. The coarse stage obtains features 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 achieving good results, the network size and depth eventually become excessive, leading to high GPU memory consumption.

Multi-patch networks have been proposed by Nekrasov [10] and Zhang *et al.* [11]. Both used a recurrent method by regarding the results of the last iteration as the input of the next round to refine final checkpoints. Images are separated into patches for extracting features, and the meaningful results are sent to the next iteration for further enhancement. This method is conducive to parameter reduction by learning from patches in one round. However, the approach is hindered by low image restoration efficiency.

Image deblurring using CNNs is more accurate than that using traditional methods. However, CNNs, especially multiscale deep CNNs, are memory- and computation-intensive when deblurring images, thereby hindering real-time application. Meanwhile, computation-efficient networks cannot handle large-scale datasets and often cannot generate satisfactory restoration results. Additionally, deep learning methods neglect edge and color reconstruction for specific regions and different restoration situations. Experiments have been conducted to prove the significant impact of the lightweight process and the residual connections on the enhanced accuracy and decreased complexity of the proposed network. Our contributions are as follows:

- We propose a novel multi-frequency feature extraction and fusion network for image deblurring. Compared with the previous multiscale and recurrent architectures, our model is more efficient and performs well in terms of image quality as Fig. 1 shows.
- A contextual attention fusion unit is designed that fuses the extracted edge and sharpness feature information from multilevel paths. The modules can both solve the multilevel requirement of concatenating different kinds of feature maps and help to train a deep and fast network.
- A multiscale refinement loss function was developed on the VisDrone and GOPRO datasets. State-of-the-art deblurring performance is achieved according to the results of quantitative analysis using PSNR and SSIM.

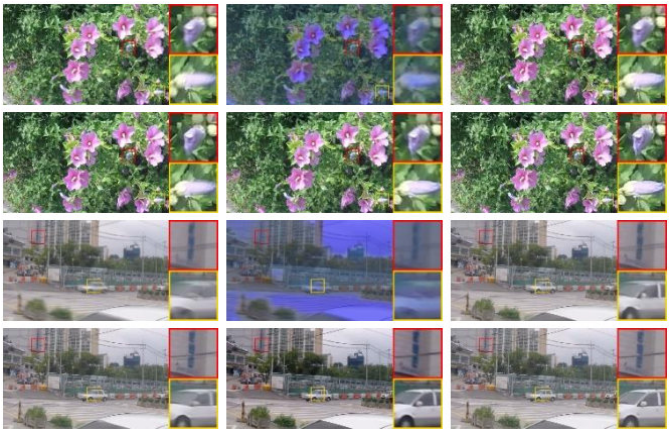

Fig. 1. Results of comparative experiments. Our restored images show vivid colors and sharp details. The above are blurry, DeblurGAN, DMPHN, SIUN, ours and ground truth images

The remainder of this paper is organized as follows. We discuss related work on image deblurring network architectures in Section 2. Section 3 illustrates the methodology and the implementation of our proposed network. We discuss our experimental results in Section 4 and conclude the paper in Section 5.

2. RELATED WORK

Traditional methods [4, 5, 11–15] rely on blur kernel estimation to reconstruct images by focusing on specific types of blurs. Recent studies [16, 17] have attempted to solve the restoration problem by adopting multiscale CNNs to deblur images. In these end-to-end frameworks [18, 19], blurry images are used as inputs for the neural network to immediately generate clear images. Compared with traditional methods, CNNs have greatly improved computing speed, but their prediction accuracy is low, and considerable GPU memory is needed.

As for feature extraction, image deblurring CNNs can be categorized into GAN, multiscale network, recurrent network, multi-patch network, and scale-iterative network architectures.

By scaling an image into different sizes, multiscale networks [20, 21] are able to extract various features from each scale, as shown in Fig. 2a. The input images are converted into feature maps, and then scales are used to halve the feature maps at the next level. In multiscale detection, various scale features are fused by different methods. These various scale features contain a large quantity of information, suggesting high accuracy. However, the multiscale strategy strictly requires that the features be extracted from small to large scale, which means that large-scale concatenating needs to wait for the computing results from the small scales, resulting in a slow training speed.

A recurrent network comprises an input layer, a loop-hiding layer, and an output layer [12, 22, 23], as shown in Fig. 2b. Recurrent networks can learn features and long-term dependencies in a sequence. However, the complexity increases with the number of network layers. 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 [11] is a CNN model that appears to be simple but operates as an effective multi-patch network as shown in Fig. 2c. In DMPHN, the input image is divided into different sizes each time; then, 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.* [13] proposed a scale-iterative network [24] that restores sharp images iteratively, as shown in Fig. 2d. The super-resolution structure of the upsampling layer is adopted between

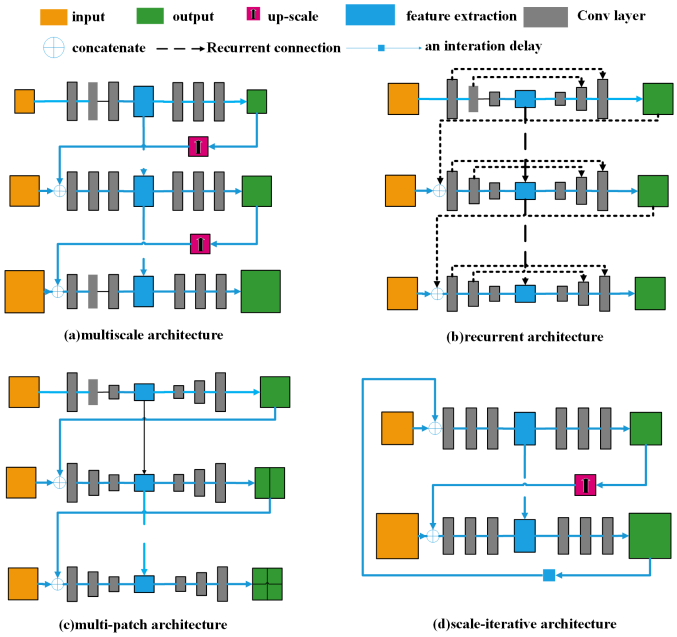

Fig. 2. Various deblurring network architectures. (a) Multiscale architecture, which extracts features from different scales. (b) Recurrent architecture, in which the next round of training is aided by the last-round results. (c) Multi-patch architecture, which directly extracts features from image pairs by cropping images at different scales. (d) Scale-iterative architecture, which is used to train the model with an upsampling path with the aid of the last-iterative middle results. We combine architectures (a) and (b) and propose a new framework, whose core module involves the MFF, called MFFNet. MFFNet operates in both a multiscale and a recurrent manner

two consecutive scales to restore the details. Image features are extracted from small to large scale, with the aim of reconstructing high-resolution images from low-resolution images. Then, the downsampling part restores the image until its size equals that of the original image. 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 a significant amount of memory is needed for the iterative calculations.

Our proposed method combines the edge feature learning strategy and the contextual attention modules for image restoration, which facilitates locating the object aided by structure information and the adoption of appropriate deblurring priors to reconstruct the sharp images.

3. MODEL DESIGN AND IMPLEMENTATION

The proposed MFFNet is designed to ensure a balance between accuracy and speed. We first exploited the recurrent and multiscale strategies to learn the multi-frequency information. Then, we designed a structure with a branch depth and fusion unit on the basis of the lightweight process and remote residual connections [25]. Finally, a multiscale refinement loss function was used to train the network in a coarse-to-fine manner.

3.1. Multiscale and recurrent learning

Recurrent and multiscale learning strategies were applied in this study. The basic idea of the multiscale learning strategy in Fig. 3a is to extract features from the large, coarse scale maps and upsampled results. Meanwhile, in the recurrent learning strategy in Fig. 3a, 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 at different scales instead of directly predicting the whole deblurred image. In our method, the network only needs to focus on learning the highly nonlinear residual features, which is effective in restoring deblurred images in a coarse-to-fine manner. The architecture of the proposed MFFNet is shown in Fig. 3.

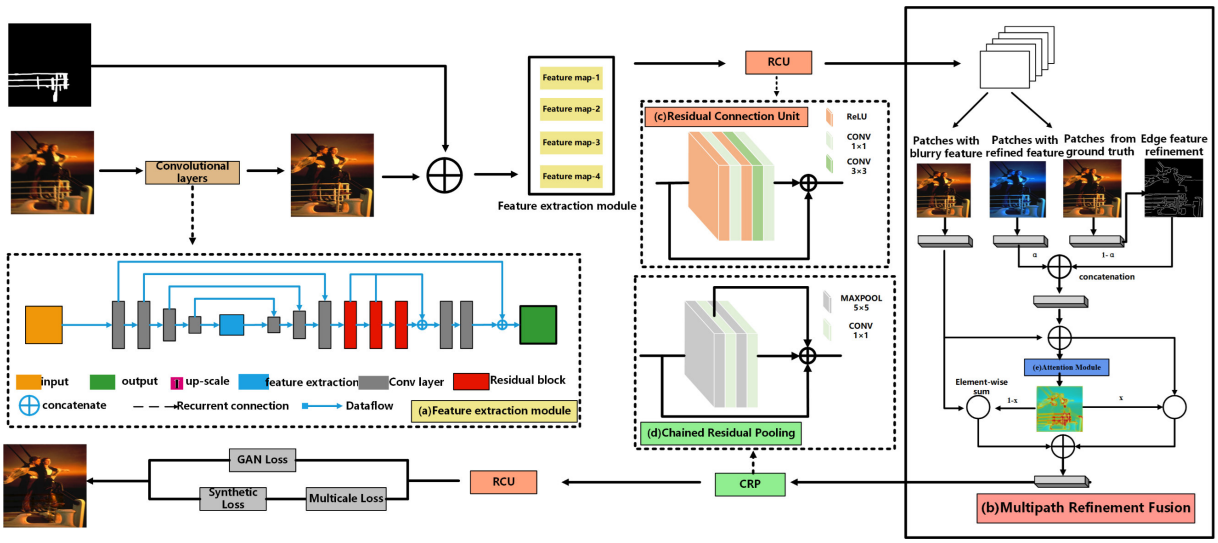

Fig. 3. Multi-frequency feature extraction and fusion (MFF) framework. The image is separated into different scales from top to bottom. Brown blocks denote the extraction path of features from scales. The multipath refinement fusion (MRF) blocks fuse the recurrent last-round results and the upsampling feature maps into a single refinement process. The four refinement paths finally compute the loss in the scale refinement loss function, and then the best deblur results are obtained. RCU, residual connection unit; CRP, chained residual pooling; LW, lightweight strategy

In the multipath input stream illustrated in Fig. 3a, the upper MFFNet layer takes blurry 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, whereas 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} into a middle feature map, which is denoted as c_k.

$$c_k = b_k \oplus l_{k+1} \quad (2 \le k \le 4). \tag{1}$$

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 demonstrates how the refinement fusion path works. The whole process can be calculated as

$$l_k = c_k + l_{k-1} \quad (2 \le k \le 4). \tag{2}$$

3.2. Edge reconstruction and attention process

Real-world image capture cannot avoid blur. For instance, Fig. 4a shows a fast-moving car on the street, which causes motion blur and, owing to the distance, the street is far from the lens, which causes Gaussian blur. The procedure employed by MFFNet to restore images comprises three steps: edge reconstruction (Fig. 4c), blur species locating (Fig. 4b), and patches deblurring (Fig. 4d).

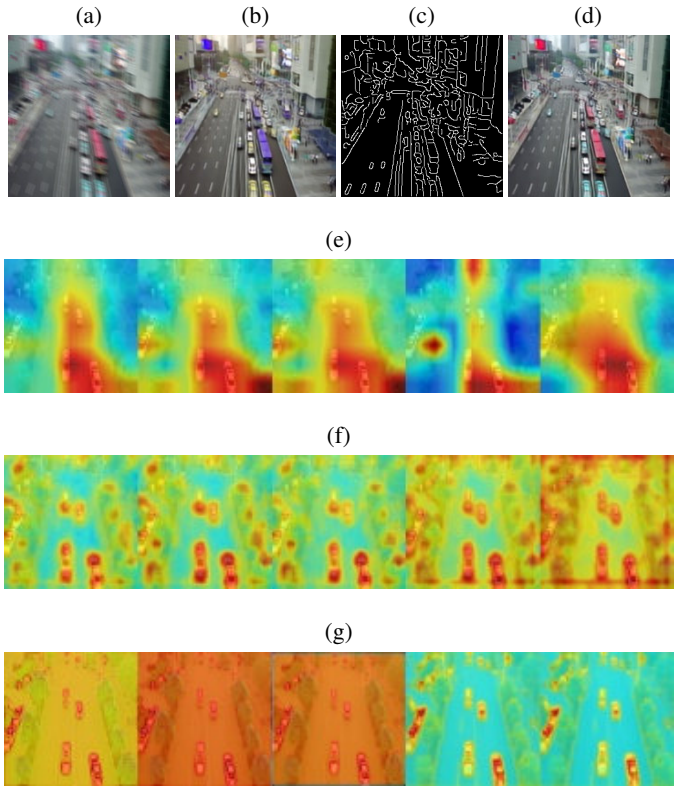

Fig. 4. Multifeature extraction for edge and sharpness

Edge reconstruction

Edge information (high-frequency features) is very important for reconstructing images because structure reconstruction is beneficial for the refinement of different blur kernels [26]. Given the blur and ground truth pairs as inputs, the edge generative network predicts the structure of the whole picture. Then, a pretrained classification network preprocesses the edge feature information to determine the location and the class associated with the deblur kernels.

The ground truth images are preprocessed into grayscale images for further edge feature extraction, and these images are sent to a discriminator for benchmark comparison. L_{edge} is the loss function of the visual evaluation; it is designed in a generative and adversarial manner. The generator G_e produces various generative edge maps for the discriminator D_e to judge the realness of the generation.

$$\begin{aligned} L_{\text{edge}} &= \min_{G_e} \max_{D_e} L_{G_e} \\ &= \min_{G_e} \left(a_{\text{adv},1} \max_{D_e} (L_{\text{adv},1}) + a_{FM} L_{AM} \right). \end{aligned} \tag{3}$$

Locate deblur category

The attention mechanism [28] functions like neural cells to focus on the interesting aspects: broad-view, classification, and location. First, we search the background to make a broad view for latent meaningful objects by convolutional layers and extract semantic information through the multipath refinement fusion unit.

The next step is classification. For a given image, $g_l(a,b)$ is the spatial information in the first layer, and G_l represents the sum of $g_l(a,b)$. Thus, for a specific object class, the input $\sum A_l G_l$ is the input of the softmax function. A is the weight corresponding to the class; it predicts the essential level of G_l. Finally, Q is the output of the softmax function; it is denoted as $\frac{\exp(S)}{\sum_e \exp(S)}$. The score S is defines as follows:

$$S = \frac{\sum A \sum g_l(a,b)}{\sum (a,b) \sum A_l \sum g_l(a,b)}. \tag{4}$$

The global average pooling score is used to predict the importance of the location of (a,b) leading to the classification of a blurry object in the image.

From Figs. 4e, 4f, and 4g, we conclude that changing the receptive field can result in the generation of different contextual attention results. When the receptive field is large, objects are perceived as a whole. When the receptive field is small, we find that each part of the object is attended to, and the texture is located in detail.

Third, we locate the deblur category. Based on the edge maps, we can search and place the blurry objects into six categories, as depicted in Fig. 5. In terms of each category, MFFNet has a different deblur kernel to refine the blur features for specific objects.

The attention module can find and locate the general objects and apply different deblur approaches through the deep learning training process. It deblurs the specific objects into sharp objects with the aid of the edge generation modules and the contextual attention mapping.

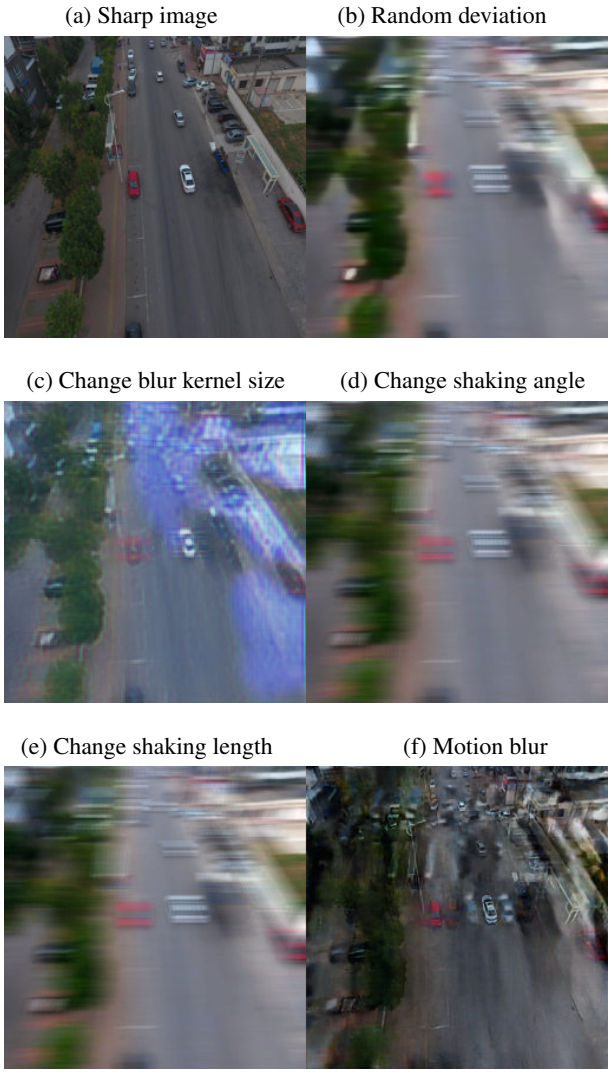

Fig. 5. Overview of blur categories for latent deblur algorithms. The first column shows a sharp image, an image blurred by changing the blur kernel size, and an image blurred by altering the blur shaking length. The second column shows the blurred image with a random standard deviation added, an image blurred by changing the shaking angle, and an image generated by real motion blur

Patches deblurring

Following edge feature extraction and location of the contextual attention, we can determine the structure information, predicted object, and blurry potential class. Subsequently, we use the deblurring feature prior network to deblur the images and obtain sharper images. In this way, we restore the image for applying different blur strategies in different regions. Hence, the target is more specific, the performance is improved, and the reconstruction of the object structure is meaningful and vivid.

4. PERFORMANCE EVALUATION

In this section, we compare MFFNet to the recently adopted methods, DeepDeblur [27], DeblurGAN [6], DeblurGANv2 [7], DMPHN [10], and SIUN [13], in terms of accuracy and time efficiency.

4.1. Experimental setup

We implemented MFFNet using Caffe. The model was trained with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$). In the training process, the input images were randomly cropped to 256×256. A batch size of 16 was used for the training on four NVIDIA RTX2080Ti GPUs. At the beginning of each epoch, the learning rate was initialized as 10^{-4} and subsequently decayed by half every 10 epochs. We trained 150 epochs for VisDrone and 150 epochs for GOPRO.

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

4.2. Dataset

We trained and evaluated the performance of MFFNet on two popular benchmark datasets: VisDrone and GoPro. VisDrone [29] provides synthetic blur techniques and real blurry aerial scenarios. GOPRO [7] provides real-world motion-blur scenarios. The size of the images in GOPRO is 1280×768, whereas that in VisDrone is 256×256.

4.3. Comparative experiments

We conducted comparative experiments with DeepDeblur [27], DeblurGAN [6], DeblurGANv2 [7], DMPHN [11], and SIUN [13] to verify the performance of our model. MFFNet achieved state-of-the-art performance compared with SIUN. The values of PSNR and SSIM are much higher than those of DeblurGAN, DeepDeblur, and DMPHN, suggesting that the proposed method is advantageous in handling mix blurs.

Moreover, our method performed better than SIUN and DMPHN and even much better than DeblurGANv2 in dealing with the motion blur of GOPRO. The trends in Table 1 prove the superiority of the MFFNet framework based on the PSNR and SSIM values. Because the VisDrone dataset has extreme blur and distorted texture augmentation, other methods obtained very low SSIM values. This is because they lack the ability to restore severe cases of missing structure information from extremely blurry images.

As shown in Table 2, DeblurGAN requires the least GPU memory at 4538 MB, while our method requires slightly more GPU memory than DeblurGAN in GOPRO. This is because DeblurGAN has the least parameters and the worst deblur restoration performance, respectively. For the VisDrone dataset, our network consumes the least GPU memory for the batch size of 16. The lightweight process reduces the number of parameters of the model, which contributes to it meeting the low-memory requirement.

As shown in Table 3, MFFNet is the fastest method in terms of the loading time of the network model and the inferences. The inference was executed on an RTX2080Ti 12G GPU. The image size from GOPRO was 1280×768 and that from VisDrone was 256×256. To prevent our network from overfitting, several data enhancement techniques were employed. Of the 24,000 pairs of images, 22,000 pairs were used for training and the remainder for testing. We augmented the data in VisDrone using techniques such as extreme blur, distorted texture, cropping patches, and image rotation. In terms of geometric trans-

Table 1
Testing results of the blurred image datasets and their PSNR and SSIM values

Method		DeepDeblur [27]	DeblurGAN [6]	DeblurGANv2 [7]	DMPHN [11]	SIUN [13]	Our model
VisDrone	PSNR	27.1494	28.29447	28.43967	28.54136	28.28039	**29.40845**
	SSIM	0.53937	0.60964	0.61488	0.52630	0.54342	**0.86247**
GOPRO	PSNR	29.4237	28.22642	32.19638	34.21846	34.46135	**34.63429**
	SSIM	0.76137	0.74791	0.8711	0.89829	0.90091	**0.90788**

Table 2
GPU memory consumption by different methods (Model=MFFNet)

Method		DeepDeblur [27]	DeblurGAN [6]	DeblurGANv2 [7]	DMPHN [11]	SIUN [13]	Our model
VisDrone	Network (MB) + Batch (8)	7930	6012	8107	7329	8561	**5898**
GOPRO	Network (MB) + Batch (8)	6311	**4538**	6861	6541	8399	5452

Table 3
Average time inferring images

Method		DeepDeblur [27]	DeblurGAN [5]	DeblurGANv2 [6]	DMPHN [11]	SIUN [13]	Our model
VisDrone	InferTime (s)	2.362	2.144	2.663	0.764	0.357	**0.319**
GOPRO	InferTime (s)	2.427	2.346	2.528	1.886	0.684	**0.494**

formations, the patch was flipped horizontally or vertically and rotated at a random angle. For color, the RGB channel was randomly replaced. To consider image degradation, saturation in the HSV color space was multiplied by a random number in the range [0,5]. In addition, Gaussian random noise was added to the blurred image. To make our network robust to noise at different levels, the standard deviation of noise was also randomly sampled from a Gaussian distribution N (0–1). In the form of a preset blur kernel, blur was artificially added to the clear image to ensure that pairs of training data could be obtained.

4.4. Loss design and training strategy

Given a pair of sharp and blurred images as input, MFFNet produces four groups of feature maps at different scales. Assuming that the input image size is $H \times W$, the four scales of the feature maps are $H/4 \times W/4$, $H/8 \times W/8$, $H/16 \times W/16$, and $H/32 \times W/32$.

Loss design

In the training process, we adopted the $L2$ loss between the predicted deblurring result map and the ground truth as follows:

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \left\| x_s^i - F\left(x_l^i\right) \right\|^2, \tag{5}$$

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

The multiscale 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. Moreover, our scale refinement loss function computes the results at different scales, which leads to a much faster convergence speed and an even higher inference precision. The final loss is calculated as follows:

$$L_{\text{final}} = \frac{1}{2K} \sum_{k=1}^{K} \frac{1}{c_k w_k h_k} \left\| L_k - S_k \right\|^2 + L_{\text{edge}}, \tag{6}$$

where L_k represents the model output of the scale level K, and S_k is the k-th scale sharp map. The loss at each scale is normalized by the number of channels C_k, width W_k, and height H_k. L_{final} is the final loss function of the training process.

Progressive weighted training process

The progressive weighted training process ensures the training converge fast and smooth. At the multipath refinement extraction and fusion stage, the task is to fuse the deblurring feature and the edge feature from the outputs, to generate the final restored frame.

During the training process, the patches with blurry features, refined features, and ground truth are used as inputs.

First, the edge feature is extracted from the ground truth in the patches. Here, α is a hyperparameter that is set to zero initially to control the proportion of the refined resource. Second, the refined patches and the mixed edge feature patches are fused in the contextual attention module. Then, the contextual attention module uses the softmax function to predict the foreground and generate the preliminary activated heatmaps. Third, α is set to one, and the deblur refined feature patches are sent to the attention module during the training process and predicted by the attention module once again. The results are compared with the synthesis loss function between the predicted deblurring results and the patches with sharp feature. Therefore, at the beginning of the training, the deblurring feature refines the input blurry images and benefits the edge feature extraction; in the middle of

Table 4
Quantitative numerical results for PSNR and SSIM values

Method		RefineNet [9]	LR-RefineNet	EA-RefineNet	MFFNet
VisDrone	PSNR	28.73991	29.24461	29.03971	**29.40845**
	SSIM	0.85476	0.86016	0.85860	**0.86247**
GOPRO	PSNR	34.17826	34.21445	34.3943	**34.63429**
	SSIM	0.89437	0.90700	0.90301	**0.90788**

the training process, the deblurring and edge features are fused by controlling the parameter α; each path containing a different scale of double feature patches is refined and matched by the multipath context attention module with the activated heatmaps to infer the final predictions.

4.5. Ablation experiments

The original MFFNet uses RefineNet [9] as the benchmark. We added a lightweight and residual connection to the benchmark and denoted it as LR-RefineNet. Next, we added edge reconstruction and attention modules to LR-RefineNet and denoted it as EA-RefineNet. Finally, we combined the lightweight residual strategy and attention modules into MFFNet.

As shown in Table 4, LR-RefineNet and EA-RefineNet performed slightly better than RefineNet. MFFNet had the best numerical results.

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

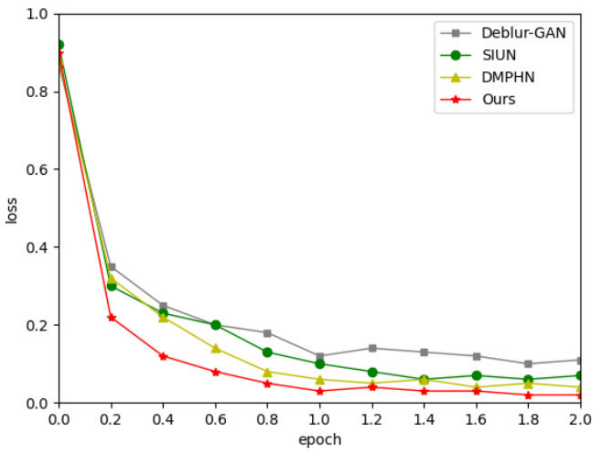

Fig. 6. Training loss of the four methods. Only the first two epochs are shown

In summary, the experimental results indicate that MFFNet can achieve considerable precision. Furthermore, MFFNet runs 4–5 times faster than other deblurring models such as SIUN and DMPHN. Compared with DeblurGAN and DeblurGANv2, the proposed MFFNet model performs well both in terms of speed and the deblurring quality of images. Owing to the added lightweight process, the GPU memory occupation remains at a low level. Our method can also recover more details and achieve relatively high SSIM and PSNR values. Figure 7 shows the results of other models whose images remain unstable and sometimes contain artifacts and color distortions, whereas the MFFNet performs image deblurring in a stable and sharp manner.

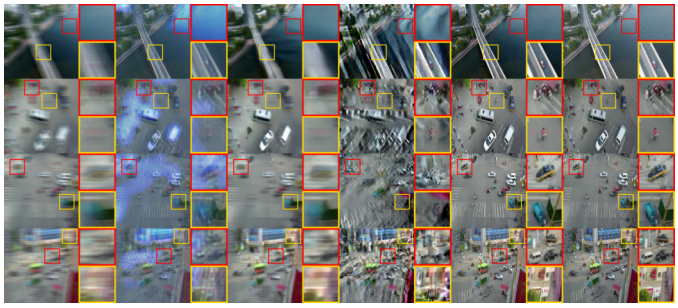

Fig. 7. Results of comparative experiments: (left to right) blurry, DeblurGAN, DMPHN, SIUN, ours, and ground truth images

5. CONCLUSION AND FUTURE WORK

In this study, we propose an efficient and accurate framework called MFFNet. The proposed network utilizes a lightweight process, remote residual connection, edge attention mechanisms, and scale refinement loss function to enable the model to handle real blur scenarios, preserving fast inference speed and high precision. It can extract various features by scheduling the weight of the joint training losses and carries out fusion guided by attention modules, leading to accurate and efficient image restoration. We compared MFFNet with existing models on two popular deblurring datasets and showed that it achieves state-of-the-art performance.

In future work, we will develop a faster deblurring inference engine for MFFNet. The computational capability will likely be much lower than that of the GPUs used in our experiments. Model compression techniques, such as pruning and quantization, will also be explored. We will also apply this model to video deblurring or the deblurring of inpainting results at the post-processing stage.

REFERENCES

[1] A. Cichocki, T. Poggio, S. Osowski, and V. Lempitsky, "Deep learning: Theory and practice," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 66, no. 6, pp. 757–759, 2018.

[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] C. Dhanamjayulu *et al.*, "Identification of malnutrition and prediction of BMI from facial images using real-time image processing and machine learning," *IET Image Process.*, 2021, doi: 10.1049/ipr2.12222.

[5] 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.

[6] O. Kupyn *et al.*, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 81838192.

[7] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. 2019 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Korea (South), 2019, pp. 8877–8886, doi: 10.1109/ICCV.2019.00897.

[8] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3883–3891.

[9] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 5168–5177, doi: 10.1109/CVPR.2017.549.

[10] V. Nekrasov, C. Shen, and I. Reid, "Light-weight RefineNet for real-time semantic segmentation," in *Proc. BMVC*, 2018.

[11] 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.

[12] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8174–8182.

[13] M. Ye, D. Lyu, and G. Chen, "Scale-iterative upscaling network for image deblurring," *IEEE Access*, vol, 8, pp. 18316–18325, 2020.

[14] T.R., Gadekallu, D.S. Rajput, M.P.K. Reddy, K. Lakshmanna, S. Bhattacharya, S. Singh, A. Jolfaei, and M. Alazab, "A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU," *J. Real Time Image Process.*, vol. 18, pp. 1383–1396, 2021, doi: 10.1007/s11554-020-00987-8.

[15] 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.

[16] Z. Zhang, H. Chen, X. Yin, and J. Deng, "Joint generative image deblurring aided by edge attention prior and dynamic kernel selection," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021.

[17] Z. Zhang, H. Chen, X. Yin, and J. Deng, "EAWNet: An edge attention-wise objector for real-time visual internet of things," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021.

[18] Z. Zhang, H. Chen, J. Deng, and X. Yin, "A double feature fusion network with progressive learning for sharper inpainting," in *Proc. 2021 Int. Joint Conf. Neural Netw. (IJCNN)*, 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9534018.

[19] Z. Zhang, J. Deng, H. Chen, and X. Yin, "Rotated YOLOv4 with attention-wise object detectors in aerial images," in *Proc. 2021 4th Int. Conf. Robot Systems and Applications*, April 2021, pp. 1–6.

[20] A. Naeem, A.R. Javed, M. Rizwan, S. Abbas, J.C.W. Lin, and T.R. Gadekallu, "DARE-SEP: A hybrid approach of distance aware residual energy-efficient SEP for WSN," *IEEE Trans. Green Comm. Netw.*, vol. 5, no. 2, pp. 611–621, 2021.

[21] J. Dai, and Y. Wang, "Multiscale residual convolution neural network and sector descriptor-based road detection method," *IEEE Access*, vol. 7, pp. 173377–173392, 2019.

[22] K. Schelten, S. Nowozin, J. Jancsary, C. Rother, and S. Roth, "Interleaved regression tree field cascades for blind image deconvolution," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 494–501.

[23] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3929–3938.

[24] Q. Wang, S. Shi, S. Zheng, K. Zhao, and X. Chu, "FADNet : A fast and accurate network for disparity estimation," in *Proc. ICRA*, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. Comput. Vis. and Pattern Recognit, (CVPR)*, 2016.

[26] S. Zheng, Z. Zhu, J. Cheng, Y. Guo, and Y. Zhao, "Edge heuristic GAN for non-uniform blind deblurring," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1546–1550, 2019.

[27] J. Mei, Z. Wu, X. Chen, *et al.* "DeepDeblur: Text image recovery from blur to sharp," *Multimed. Tools Appl.*, vol. 78, pp. 18869–18885, 2019, doi: 10.1007/s11042-019-7251-y.

[28] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. 2016 IEEE Conf. Comput. Vis. and Pattern Recognit, (CVPR)*, 2016.

[29] P. Zhu *et al.*, "VisDrone-VID2019: The vision meets drone object detection in video challenge results," in *Proc. 2019 IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Seoul, Korea (South), 2019, pp. 227–235, doi: 10.1109/ICCVW.2019.00031.